

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТАВРІЙСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. І. ВЕРНАДСЬКОГО

Н. В. Апатова, О. М. Гончарова, Ю. Ю. Дюлічева

ІНФОРМАТИКА ДЛЯ ЕКОНОМІСТІВ

ПІДРУЧНИК

*Затверджено
Міністерством освіти і науки України
для студентів вищих навчальних закладів*

Київ
«Центр учбової літератури»
2011

УДК 330.47(075.8)
ББК 32.973я73
А 76

*Гриф надано
Міністерством освіти і науки України
(Лист № 1/11-4563 від 01.06.2010)*

Рецензенти:

Чепурко В. В. – доктор економічних наук, професор;
Сейдаметова З. С. – доктор педагогічних наук, доцент;
Матвеев В. В. – кандидат фізико-математичних наук, доцент.

Апатова Н. В., Гончарова О. М., Дюлічева Ю. Ю.
А 76 Інформатика для економістів. Підручник. – К.: Центр учбової літератури, 2011. – 456 с.

ISBN 978-611-01-0159-2

Підручник написаний згідно з останніми освітньо-кваліфікаційними вимогами та програмами підготовки бакалаврів економічних спеціальностей. Він є початковим курсом з економічної інформатики і призначений для формування базових знань про комп'ютерно-інформаційні технології та можливості їх застосування при розв'язанні прикладних економічних задач і прийнятті економічних рішень. У книзі викладено основні принципи та особливості економічної інформації, апаратного та програмного забезпечення для її обробки, загальні тенденції розвитку і застосування операційних систем, технології Microsoft Office, комп'ютерних мереж, а також сучасних прогресивних інформаційних та цифрових технологій.

Для студентів економічних спеціальностей вищих навчальних закладів, а також для широкого кола фахівців, що працюють з економічною інформацією та інформаційними системами в економіці.

УДК 330.47(075.8)
ББК 32.973я73

ISBN 978-611-01-0159-2

© Апатова Н. В., Гончарова О. М.,
Дюлічева Ю. Ю., 2011.
© Центр учбової літератури, 2011.

Розділ I. ОСНОВНІ ПОНЯТТЯ ІНФОРМАТИКИ

Впровадження комп'ютерів в усі сфери людської діяльності вимагає від фахівців нового покоління у галузі фінансів і кредиту, економічної кібернетики, бухгалтерського обліку й аудита та інших сфер діяльності наявності знань, умінь і практичних навичок використання новітніх інформаційних систем і технологій. Сьогодні неможливо уявити ефективну роботу представника економічної спеціальності без знання основ роботи комп'ютера й комп'ютерних мереж, зокрема, Інтернету. Так, в 2000 р. у швейцарському місті Давосі відбувся економічний форум, на якому було визнане існування нової категорії економічних взаємин за назвою «Інтернет-Економіка». Це лише один з прикладів необхідності вивчення для економістів комп'ютерної мережі Інтернет.

У цьому розділі розглядаються основні визначення та поняття інформатики та економічної інформатики.

1.1. Об'єкт, предмет, метод і завдання вивчення інформатики

Передісторія появи інформатики настільки ж стародавня, як і історія розвитку людського суспільства, що пов'язана з перетворенням суспільних відносин під впливом змін способів опрацювання інформації.

Перший важливий етап розвитку людського суспільства пов'язаний з появою усного мовлення, другий – з винаходом писемності, третій (середина XVI ст.) – друкарства, четвертий (кінець XIX ст.) – електрики (з'явилася можливість передавати інформацію за допомогою телеграфу, телефону, радіо тощо), п'ятий (середина XX ст.) – з винаходом комп'ютерної техніки. П'ятий етап став початком для формування інформатики як наукової дисципліни.

У процесі розвитку людського суспільства спостерігалось зростання обсягів оброблюваної інформації, що привело до перерозподілу трудових ресурсів зі сфери матеріального виробництва в нову інформаційну сферу і появи нового виду ресурсів – інформаційного ресурсу суспільства. Розумне керування будь-якою сферою людського суспільства й, у першу чергу, сферою економіки можливо тільки на підставі вчасного накопичення й опрацювання інформаційного ресурсу.

➤ **Інформаційний ресурс** – це сукупність документів в архівах, фондах, бібліотеках, банках даних та інших інформаційних системах.

Впровадження електронних обчислювальних машин (ЕОМ) у різні сфери людської діяльності спричинило формування процесу інформатизації суспільства.

➤ **Інформатизація суспільства** – це організований науково-технічний і соціально-економічний процес створення оптимальних умов для задоволення інформаційних потреб суспільства на підставі формування й використання інформаційних ресурсів.

Результатом процесу інформатизації є створення інформаційного суспільства. Для інформаційного суспільства характерна розвинена інфосфера (інформаційна сфера). Згідно з ученням В.І. Вернадського про ноосферу інфосфера розглядається як один з етапів перетворення біосфери в ноосферу – сферу розуму.



Володимир Іванович Вернадський (1863 — 1954) – видатний російський вчений ХХ ст., натураліст, мислитель і громадський діяч, автор учення про ноосферу. До кола його інтересів входили геологія, кристалографія, мінералогія, геохімія, біологія, філософія.

Дослідження останніх років чітко показують, що *інформатика як фундаментальна наукова дисципліна буде визначати напрямки формування і розвитку глобального інформаційного суспільства, заснованого на знаннях.*

Французький термін *informatique* (інформатика), утворений шляхом злиття термінів інформація (*information*) і автоматика (*automatique*), дослівно перекладається як «інформаційна автоматика або автоматизована переробка інформації», тому під **інформатикою** традиційно розуміють комплексну наукову й інженерну дисципліну, що вивчає процеси збору, зберігання й переробки інформації за допомогою обчислювальної техніки.

Однак єдиного загально визначеного пояснення, що слід розуміти під терміном «інформатика», не існує. Тому наведемо декілька визначень інформатики. Існуюча різноманітність визначень терміна «інформатика» пояснюється багатогранністю функцій, засобів і методів цієї наукової дисципліни. У зарубіжних країнах терміну «інформатика» відповідає, щонайменше, чотири галузі знань – Information science, Computer science, Computer science and Information science, Computational science. Наприклад, Information science вивчає різні інформаційні процеси, Computer science спрямована на вивчення комп'ютерної бази для проведення наукових досліджень і прикладних розробок.

З погляду В.М. Глушкова **поняття інформатики** охоплює галузі, пов'язані з розробкою, створенням, використанням і матеріально-технічним

обслуговуванням системи опрацювання інформації, включаючи машини, обладнання, математичне забезпечення, організаційні аспекти, а також комплекс промислового, комерційного, адміністративного, соціального та політичного впливу».

З погляду академіка Н.Н. Моїсеєва «**інформатика** – це якась синтетична дисципліна, що містить у собі й розробку нової технології наукових досліджень і проектування, які засновані на використанні електронної обчислювальної техніки, і кілька великих наукових дисциплін, пов'язаних із проблемою спілкування з машиною, і, нарешті, зі створенням машини».

З погляду академіка А.Н. Дородніцина «**інформатика** – це наука про перетворення інформації, яка базується на обчислювальній техніці».

З погляду А.П. Єршова, Ю.І. Шемакіна, Ю.А. Шрейдера та А.Д. Урсула «**інформатика** – фундаментальна наука, що вивчає загальні інформаційні закономірності, притаманні як живим, так і штучним системам».

➤ **Об'єктом інформатики** є інформаційні процеси, що спостерігаються у системах різної природи.

➤ **Предметом інформатики** виступають основні властивості й закономірності, притаманні інформаційним процесам, які спостерігаються в різних інформаційних середовищах.

➤ **Найбільш популярними методами інформатики** є метод інформаційного моделювання, метод інформаційного підходу, віртуальна реальність.

Умовно інформатику можна розділити на дві великі частини: теоретичну інформатику і практичну інформатику. *Теоретична інформатика* вивчає принципи проектування, створення і використання автоматизованих інформаційних систем. *Практична інформатика* вивчає конкретні різновиди інформаційних технологій.

У зв'язку з виникненням інформатизації в різних сферах людської діяльності стали з'являтися різні галузеві напрямки інформатики: економічна інформатика, військова інформатика, медична інформатика, соціальна інформатика, біологічна інформатика тощо. Виділення галузевих інформатик пов'язане з різноманіттям сучасних уяв про інформацію й обумовлене тим, що феномен інформації демонструє себе по-різному в різних інформаційних середовищах. Розглянемо об'єкт, предмет, метод і завдання економічної інформатики.

➤ **Економічна інформатика** – це наукова дисципліна, яка вивчає інформаційні системи (ІС), що використовуються для підготовки й прийняття рішень у сфері економіки, управління та бізнесу.

➤ **Об'єктом економічної інформатики** є інформаційні системи, що забезпечують розв'язок підприємницьких і організаційних завдань, які виникають в економічних системах. При вивченні інформаційних систем особлива увага приділяється питанням керування ІС і економіці інформаційних систем, тобто питанням одержання оцінок витрат і прибутків від впровадження, експлуатації і розвитку інформаційної системи у структурі керування.

➤ **Предметом економічної інформатики** є способи автоматизації інформаційних процесів із застосуванням економічних даних.

➤ **Методом економічної інформатики** є моделювання бізнес-процесів.

➤ **Завданнями економічної інформатики** є вивчення теоретичних основ інформатики; набуття знань, умінь і навичок використання прикладних систем обробки економічних даних і систем програмування для персональних комп'ютерів, локальних і глобальних комп'ютерних мереж.

Оскільки кожна з галузевих інформатик пов'язана з вивченням способів зберігання, передачі й опрацювання інформації, то в наступному пункті розглянемо одне з найважливіших понять інформатики – поняття інформації, її властивості й способи подання.

1.2. Інформація та її властивості

Термін «інформація» походить від латинського слова «informatio», що означає роз'яснення, виклад. Багато процесів у природі пов'язані з передачею, перетворенням і зберіганням інформації. Під інформацією ми розуміємо ті нові відомості про навколишній світ, які ми одержуємо в результаті взаємодії з ним, пристосовуючись до нього та змінюючи його в процесі пристосування.

➤ **Інформація** – це відомості про те, що є об'єктом зберігання, передачі й перетворення. Інформація – сукупність деяких відомостей, кількісних даних.

До початку нинішнього сторіччя для передачі й зберігання інформації використовувалися книги, а її переробкою займалася сама людина. Однак науково-технічна революція, що відбувається, зробила людину неспроможною у потужному потоці друкованої, кіно-, теле- та іншої продукції. Щорічно у світі публікується близько 100 тисяч найменувань журналів на 60 мовах, 5 млн. наукових статей, книг, брошур, 250 тисяч дисертацій і звітів. Сучасному фахівцеві варто було б прочитувати півтори тисячі сторінок щодня, щоб не відставати від рівня сьогодення. Керування виробництвом, використання складних технологій вимагає від керівників швидкого аналізу виникаючих ситуацій, ухвалення правильного

рішення. Складне наукове завдання вже не може бути вирішене вручну, навіть якщо талановитий учений витратить на нього все своє життя. Переробляти всі ці дані людині допомагає комп'ютер. На стику науки й техніки з'явився новий напрямок — інформаційні технології. За визначенням одного із засновників вітчизняної інформатики В. М. Глушкова **«Інформаційні технології — процеси, пов'язані з переробкою інформації»**.

Інформація не існує сама по собі, вона характеризує якийсь об'єкт, явище або процес і сприймається споживачем. Перелічимо властивості, які повинна мати інформація.

1. *Вірогідність (адекватність)* – це властивість інформації однозначно відповідати відображуваному об'єктові або реальному стану об'єктивної дійсності при відсутності прихованих помилок у такій інформації.

2. *Повнота* – це властивість інформації, що дозволяє характеризувати об'єкт вичерпним для споживача засобом, що й надає можливість ухвалювати на основі такої інформації управлінські рішення.

3. *Релевантність (цінність, корисність)* – це властивість інформації відповідати потребам споживача, що характеризує, наскільки інформація сприяє досягненню поставлених перед споживачем цілей і завдань.

4. *Доступність* – це властивість інформації, що характеризує можливість її одержання будь-яким споживачем.

5. *Актуальність (вчасність)* – це властивість інформації, що характеризує відповідність інформації теперішньому моменту часу.

6. *Коректність* – це властивість інформації, що полягає в такому її зображенні, щоб інформація однозначно сприймалася всіма її споживачами.

7. *Захищеність* – це властивість інформації, що характеризує неможливість несанкціонованого доступу й цілеспрямованого спотворення інформації.

8. *Ергономічність* – це властивість інформації, що характеризує достатність обсягу і форми інформації для даного споживача.

Окремий інтерес представляє особливим чином організована інформація – дані й знання.

➤ **Дані** – це інформація, представлена в зручному для опрацювання вигляді.

➤ **Знання** – це інформація, на основі якої можна одержати нову інформацію (зробити певні висновки) за допомогою логічних міркувань.

Для зручності передачі інформації були розроблені методи кодування інформації. Кодування – це процес подання інформації у вигляді послідовності символів деякого алфавіту, яка має назву кодової комбінації або коду. Питання кодування інформації тісно пов'язані з розробкою різних інформаційних технологій. Відповіді на ці питання дозволяють вирішувати

багато практично важливих завдань програмування, таких як зображення чисел, тексту або графіки в пам'яті комп'ютера, стиснення інформації в базах даних, захист інформації від несанкціонованого доступу тощо.

Кодування інформації. Розглянемо схему передачі інформації від джерела повідомлення до адресата (рисунк 1.2.1). Така схема є досить загальною й дозволяє описати процес передачі інформації в комп'ютерній мережі (джерело повідомлення й адресат – комп'ютер) або передачі інформації за допомогою мобільного зв'язку (джерело повідомлення й адресат – людина). Слід зауважити, що від пропускну здатності каналу зв'язку й захищеності (завадостійкості) переданого повідомлення залежить ефективність і вчасність прийняття управлінських рішень у різних соціально-економічних системах.

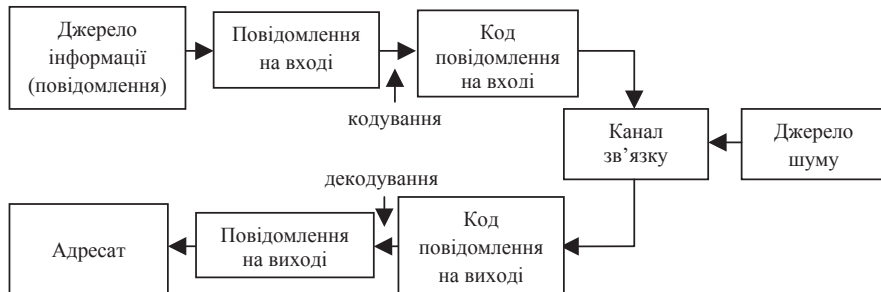


Рис.1.2.1. Схема передачі інформації по каналу зв'язку

Зі схеми видно, що кодування являє собою перехід від повідомлення на вході каналу зв'язку до коду повідомлення на вході, а декодування – зворотний процес переходу від коду повідомлення на виході каналу зв'язку до повідомлення на виході, при цьому повідомлення на вході й виході з каналу зв'язку повинні збігатися.

Більш суворо задачу кодування можна сформулювати в такий спосіб. Нехай є два алфавіти: алфавіт A , що складається з n символів і алфавіт B , що складається з t символів. Під алфавітним кодуванням будемо розуміти таке відображення F , яке кожному слову з непустиї підмножини слів над алфавітом A ставить у відповідність деяке слово над алфавітом B . Слова над алфавітом B називаються кодовими послідовностями або кодами. Традиційно, відображення F задається у вигляді деякого алгоритму й повинне задовольняти наступним вимогам:

1) взаємна однозначність, тобто можливість для кожного повідомлення однозначно побудувати його код і, обернено, по кожному коду однозначно відновити вихідне повідомлення;

2) завадостійкість, тобто можливість виявляти і виправляти помилки, які можуть виникнути при передачі коду повідомлення по каналу зв'язку під впливом джерела шуму;

3) економічність, тобто можливість побудови кодів мінімальної довжини для повідомлень, що зустрічаються найчастіше (реалізація ефективного стиснення).

Побудова ефективного алгоритму кодування – це пошук компромісу між другою і третьою вимогами, які суперечать одна одній, оскільки завадостійкість досягається за рахунок збільшення довжини кодових послідовностей.

Найпоширенішим для подання інформації в комп'ютері є двійкове кодування – подання тексту, графіки тощо у вигляді двійкових послідовностей символів, тобто послідовностей символів, що складаються тільки з нулей та одиниць. У різних технічних пристроях одиниця інтерпретується як наявність імпульсу, нуль – як відсутність імпульсу; у логіці одиниця інтерпретується як правдиве висловлення (правда), а нуль – як хибне (неправдиве) висловлення. Символи 0 і 1 називаються *бітами*. Нехай алфавіт A складається з n символів, алфавіт B складається із двох символів – 0 і 1. Виникає питання: скільки двійкових послідовностей довжини n можна побудувати? Очевидно, число таких двійкових послідовностей буде дорівнювати $\underbrace{2 \cdot 2 \cdot \dots \cdot 2}_n = 2^n$.

Розглянемо кодування текстової інформації за допомогою двійкових послідовностей. Нехай є деяка абетка. Кожній літері абетки можна зіставити порядковий номер, тобто ціле число, а кожному цілому числу – зіставити двійкову послідовність. Наприклад, для англійської мови Інститутом стандартизації США була введена система кодування ASCII (American Standard Code for Information Interchange), що дозволяє закодувати $2^8=256$ символів у вигляді двійкових послідовностей довжини 8. Система кодування ASCII складається з базової (значення кодів від 0 до 31) і розширеної таблиці (значення кодів від 32 до 255). Розроблювачі ASCII перші 32 кода (від 0 до 31) базової таблиці використали для подання керуючих символів; коди значень від 32 до 127 були використані для подання символів англійського алфавіту, розділових знаків, знаків арифметичних дій і інших допоміжних символів.

У світі існують і інші системи кодування. Наприклад, російськомовним і україномовним користувачам продуктів компанії Microsoft добре відома система кодування Windows-1251.

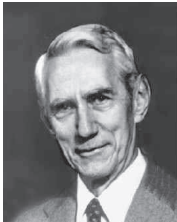
Двійкові послідовності з 8 бітів від 0000 0000 до 1111 1111 називаються *байтом*.

Розглянемо способи кількісного виміру інформації.

Кількість інформації. Найбільш відомими способами виміру кількості інформації є об'ємний, ентропійний і алгоритмічний способи.

Об'ємний спосіб виміру кількості інформації. Під обсягом інформації в повідомленні мають на увазі кількість символів цього повідомлення. Обсяг інформації – це досить груба кількісна характеристика інформації, оскільки суттєво залежить від форми подання інформації.

Ентропійний спосіб виміру кількості інформації. У теорії інформації під інформацією за Шенноном розуміють відомості, що зменшують існуючу до їхньої появи невизначеність.



Клод Елвуд Шеннон (1916 — 2001) – видатний американський математик та інженер, один із засновників математичної теорії інформації та зв'язку, а також один із засновників теорії дискретних автоматів.

Нехай адресат (одержувач повідомлення) має деякі уявлення про результати певного експерименту. Ці результати реалізуються з деякими ймовірностями. Якщо результати експерименту рівноймовірні, то невизначеність адресата, що пов'язана з результатом експерименту, максимальна. Таким чином, кількість інформації в повідомленні визначається тим, наскільки зменшиться невизначеність після одержання повідомлення. З цього погляду, кількість інформації, що міститься в отриманому повідомленні тим більше, чим більше була невизначеність до передачі повідомлення.

Припустимо, що експеримент має k рівноймовірних результатів, тобто ймовірність появи будь-якого результату даного експерименту дорівнює $1/k$. Міра загальної невизначеності експерименту дорівнює $\log k$. У теорії інформації за основу логарифма беруть два, тобто за одиницю виміру невизначеності приймають невизначеність, яка міститься в експерименті з двома рівноймовірними результатами. Така одиниця виміру невизначеності називається *бітом*.

У загальному випадку, коли результати w_1, w_2, \dots, w_n експерименту α реалізуються з різними ймовірностями p_1, p_2, \dots, p_n *міра невизначеності, яка пов'язана з експериментом α або ентропія* обчислюється за формулою

$$H(\alpha) = -\sum_{i=1}^n p_i \log_2 p_i$$

Розглянемо приклади. Нехай в урні міститься 99 чорних куль і 1 біла куля. Кулі в урні ретельно перемішані. Навмання витягується одна куля. Оскільки ми майже впевнені у результаті експерименту, тобто у появі кулі чорного кольору, то невизначеності, пов'язаної з результатом експерименту, практично немає, і міра невизначеності або ентропія в цьому випадку буде мінімальна й близька до нуля. Дійсно, експеримент має два результати: w_1 – поява кулі білого кольору з імовірністю $p_1 = 0,01$ і w_2 – поява кулі чорного кольору з імовірністю $p_2 = 0,99$. Тоді ентропія обчислюється за формулою $H(\alpha) = -0,01 \cdot \log_2 0,01 - 0,99 \cdot \log_2 0,99 \approx 0,081$.

Нехай в урні міститься 50 чорних куль і 50 білих куль. Кулі в урні ретельно перемішані. Навмання витягується одна куля. Оскільки ми не впевнені у результаті експерименту, тобто неможливо заздалегідь угадати якого кольору буде витягнута куля, то невизначеність, яка пов'язана з результатом даного експерименту, максимальна. Дійсно, експеримент має два результати: w_1 – поява кулі білого кольору з імовірністю $p_1 = 0,5$ і w_2 – поява кулі чорного кольору з імовірністю $p_2 = 0,5$. Тоді ентропія обчислюється за формулою $H(\alpha) = -0,5 \cdot \log_2 0,5 - 0,5 \cdot \log_2 0,5 \approx 0,1$.

Алгоритмічний спосіб виміру інформації. Алгоритмічний підхід до визначення поняття «кількість інформації» був запропонований Андрієм Миколайовичем Колмогоровим у 1965 році.



Андрій Миколайович Колмогоров (1903 — 1987) – видатний представник сучасної математики, автор більш ніж 500 наукових праць з теорії функцій, теорії множин, топології, теорії ймовірностей, теорії інформації, теорії алгоритмів і теорії турбулентності. Його ім'я стоїть поруч з іменами Пуанкаре і Гільберта.

Згідно з алгоритмічним підходом за кількість інформації приймається значення деякої функції від складності кожного з об'єктів і довжини програми (алгоритму) перетворення одного об'єкта в інший. Інтуїтивно зрозуміло, що комп'ютерна програма, яка друкує повідомлення, що містить тільки нулі, у край проста. Усе, що потрібно робити цій програмі, – це друкувати один і той самий символ – нуль. Якщо ж повідомлення являє собою деяку послідовність, яка не підпорядковується ніяким закономірностям, то таке повідомлення не може бути реалізоване жодною «простою» програмою. У цьому випадку, довжина програми буде близька до довжини самої послідовності. Таким чином, кількість інформації в повідомленні

визначається складністю програми, яка здатна відтворити це повідомлення – послідовність символів.

1.3. Економічна інформація, її класифікація і властивості

Залежно від сфери людської діяльності, у якій виникає й використовується інформація, розрізняють наукову інформацію, економічну (соціально-економічну, управлінську, виробничу) інформацію, технічну інформацію, медичну інформацію тощо. Кожна з цих видів інформації має свої особливі технології опрацювання, значення цінність, форми подання, вимоги до точності, вірогідності, оперативності висвітлення фактів.

Економічна інформація нерозривно пов'язана з економікою й народним господарством. Як правило, цей вид інформації виникає в процесі виробничо-господарської діяльності й використовується з метою керування цією діяльністю.

Інформація, яка обслуговує процеси виробництва, розподілу, обміну й споживання матеріальних благ і забезпечує рішення організаційно-економічного керування економікою та її ланками, називається *управлінською*. Вона являє собою різні відомості економічного, технологічного, соціального, юридичного, демографічного й іншого змісту. В управлінській діяльності інформація виступає як один з найважливіших ресурсів поряд з матеріальними, енергетичними, трудовими, фінансовими ресурсами.

У технології обробки початкові відомості про виробничі й господарські операції, людей, випуск продукції, факти придбання й продажу товарів виконують роль предметів праці, а одержувана результатна інформація – продукту праці; вона використовується для аналізу й прийняття рішень.

Економічна інформація висвітлює соціально-економічні процеси, як у сфері виробництва, так і у невиробничій сфері, у всіх органах і на всіх рівнях галузевого регіонального керування. Економічна інформація висвітлює процеси виробничо-господарської діяльності за допомогою системи натуральних і вартісних показників. У всіх випадках при цьому використовуються кількісні величини, числові значення.

Відмітними рисами економічної інформації є її об'ємність і циклічність. Для більшості виробничих і господарських процесів характерна повторюваність їх стадій і, відповідно типу інформації, що висвітлює ці

процеси. Ця властивість дозволяє багаторазово використовувати розроблені моделі процесів і комп'ютерні програми з їхньої реалізації.

Економічна інформація міститься в документах, передається каналами зв'язку, у тому числі через комп'ютерні мережі, записується на магнітних і оптичних носіях.

У вузькому значенні під економічною інформацією розуміють будь-які економічні показники. Економічні показники описують різні прояви сутності: об'єкт, явище, процес. Кожна сутність має певні властивості, що виражаються у вигляді атрибутів або елементів даних. Атрибути, також називані реквізитами, є неподільними даними, вони мають одиниці виміру й висвітлюють якісні та кількісні властивості сутності. Якісні властивості можуть бути представлені у вигляді тексту.

Для того, щоб класифікувати інформацію, розглянемо поняття системи. У найбільш широкому значенні, під **системою** S розуміють сукупність взаємозалежних елементів деякої природи, які розглядаються як єдине ціле й спрямовані на досягнення загальної мети. Система не існує ізольовано, а, як правило, обмінюється інформацією із зовнішнім середовищем за допомогою каналів зв'язку – входів і виходів і в найпростішому вигляді може бути представлена схемою 1.3.1.

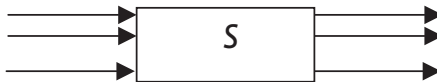


Рис.1.3.1. Найпростіша графічна схема системи S

Економіку можна розглядати як систему, у якій протікають процеси виробництва, розподілу, обміну й споживання матеріальних і інших благ.

Розрізняють зовнішню й внутрішню економічну інформацію. Зовнішня інформація надходить в економічну систему із зовнішнього середовища, внутрішня інформація – це інформація про внутрішній стан системи та її функціонування. Зовнішньою інформацією є, наприклад, тенденції зміни ВВП, рівень цін і безробіття, різні законодавчі й нормативні акти, що регулюють діяльність економічної системи, інформація про попит і пропозицію тощо. Внутрішня інформація – це виробнича, фінансово-економічна, управлінська інформація, що характеризує стан різних підсистем економічної системи, а також інформація про наявність і використання ресурсів, що забезпечують ефективне функціонування системи.

Розрізняють первинну й вторинну економічну інформацію. Первинна економічна інформація – це інформація, отримана із зовнішнього середовища

й ще не оброблена усередині економічної системи. Вторинна економічна інформація – це інформація, оброблена економічною системою.

Питання і завдання для самостійної роботи

1. Що таке інформатика?
2. Що вивчає економічна інформатика?
3. Що таке інформаційний ресурс?
4. Чим характеризується об'єкт або явище?
5. Наведіть приклади різних якісних характеристик об'єкта, досліджуваного різними науками (об'єкти: дерево, людина, повітря та ін.).
6. Які значення можуть мати властивості?
7. Що ми розуміємо під словом «інформація»?
8. Які властивості притаманні інформації?
9. Який зв'язок між даними й реальним світом?
10. Що таке дані і знання?
11. Які способи кількісного виміру інформації Ви знаєте?
12. Обгрунтуйте необхідність використання комп'ютера при розв'язанні сучасних завдань опрацювання інформації
13. Що таке інформаційна технологія?
14. Як називаються дані, якими представлена економічна інформація?
15. З якою швидкістю повинен читати текст сучасний фахівець, щоб не втратити кваліфікацію (підрахуйте приблизно кількість знаків на одній сторінці підручника й сформулюйте відповідь в одиницях знак/година). Тривалість робочого дня — 8 годин.
16. Як можна класифікувати економічну інформацію?
17. Як розуміти процес інформатизації суспільства?
18. Що таке система?
19. Який вигляд має схема передачі інформації по каналу зв'язку?
20. Які системи кодування Вам відомі?

1.4. Інформаційна діяльність

Діяльність людини, що пов'язана з процесами одержання, перетворення й накопичення інформації, називається **інформаційною діяльністю**.

Перші спроби створення інструментів для опрацювання інформації пов'язані з прагненням спростити виконання дій над числами. У Стародавньому Китаї близько 4 тисяч років тому були винайдені рахівниці. Греки й римляне більш двох тисячоріч назад почали використовувати «абак» — рахункову дошку, на якій числа зображувалися певною кількістю

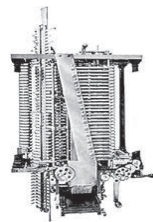
камінчиків, а дії над числами — пересуванням камінчиків. В 1642 році відомий французький фізик і математик Блез Паскаль винайшов арифмометр — пристрій для виконання додавання й вирахування, а через двадцять років німецький математик Г. Лейбніц удосконалив цю машину для чотирьох арифметичних дій. Арифмометри кілька сторіч вірно служили людям у якості незамінних помічників у бухгалтерському обліку, проведенні наукових розрахунків і інших галузях. Однак можливості арифмометра були обмежені, швидкість обчислень була невеликою, а «пам'ять» дозволяла зберігати лише результат чергової операції.

Наприкінці минулого століття в США здійснювався черговий перепис населення. Для статистичної обробки результатів був застосований табулятор Германа Холлерита — пристрій, що обробляє перфокарти. Перфокарти були розміром з долар, мали 12 рядків, у кожному з яких можна було пробити до 20 отворів. Комбінації отворів відповідали таким даним, як вік, стать, місце народження, кількість дітей та іншим. Г. Холлерит в 1924 році заснував фірму IBM (International Business Mashines Corporation).

Однак ідеї, реалізовані в сучасних комп'ютерах, були сформульовані у 30-х роках XIX століття англійцем Чарльзом Беббіджем. Розроблена ним Аналітична машина повинна була мати реєстри, у яких зберігалися б проміжні результати обчислень. Архітектура Аналітичної машини практично відповідала сучасним ЕОМ. В Аналітичній машині були присутні керуючий барабан (аналог керуючого пристрою), сховище (аналог запам'ятовувального пристрою) і млин (аналог арифметичного пристрою). Передбачалося, що Аналітична машина буде програмувальним пристроєм. Ч. Беббідж прагнув побудувати свою машину з механічних елементів (включаючи паровий двигун), але технічний рівень того часу так і не дозволив йому це зробити. Тільки через більш ніж сто років ці ідеї знайшли своє реальне втілення.



Чарльз Беббідж (1791 — 1871) — англійський математик і винахідник Аналітичної машини — першого пристрою, що має архітектуру сучасної ЕОМ.



Опису Аналітичної машини, що залишив Ч. Беббідж, виявилось достатньо для створення в 1943 році універсальної обчислювальної машини Марко-1, зібраної на електромеханічних реле. Ця машина досягала в довжину

17 метрів і у висоту 2,5 м, містила близько 750 тисяч деталей, з'єднаних проводами довжиною більш ніж 800 км. Вона проробила в Гарвардському університеті більш ніж 17 років. За день машина виконувала обчислення, на які раніше йшло півроку. Першою електронно-обчислювальною машиною, зібраною на радіолампах, стала ЕНІАК (Electronic Numerical Integrator and Computer). Так уперше в назві з'явилось слово «комп'ютер» — обчислювач.

У нашій країні перша ЕОМ (електронно-обчислювальна машина) була створена в 1951 році в м.Києві групою вчених під керівництвом академіка В. А. Лебедева.

Перші комп'ютери розуміли лише мову двійкових кодів — послідовностей, складених із цифр 0 і 1. Коли говорять про покоління обчислювальної техніки, то виділяють дві головні ознаки: елементну базу комп'ютера й рівень спілкування між комп'ютером і людиною. Чим складніше будова комп'ютера, тим більш складні програми, потрібні для його керування. Для швидкого й правильного написання програм використовуються мови, близькі до природньої. На сучасних комп'ютерах такими мовами є вже не тільки мови програмування, фрази яких складаються зі слів, але й таблиці, рисунки. Тим самим розширюються галузі застосування комп'ютерів і розв'язувані ними завдання.

В 1975 році два Стіва — Возняк і Джобс сконструювали перший персональний комп'ютер Епл, а в 1981 фірма ІВМ випустила свою знамениту модель ІВМ РС. З її появою почалася нова епоха, яка й характеризується багатопланою інформаційною діяльністю людини. У тому ж 1981 році фірма Sony анонсувала першу 3,5-дюймову дискету й дисковід. В 1985 році був випущений перший компакт-диск і накопичувач CD-ROM. В 1989 році фірма Intel випускає процесор 486, що містить мільйон транзисторів.

В 1970 році вперше у світі був здійснений зв'язок між двома комп'ютерами університету Каліфорнії й університету штату Юта, а в 1990 році в дослідному центрі CERN у Женеві була розроблена мова розмітки гіпертексту HTML (Hypertext Markup Language) і вперше з'явилася World Wide Web (www) — світова віртуальна файлова система, призначена для організації on-line доступу до інформаційних ресурсів.

Зараз простіше сказати, де не використовуються персональні комп'ютери, ніж перелічити всі галузі їх застосування. Комп'ютери у виробництві працюють на всіх етапах: від конструювання окремих деталей виробу, його дизайну до збирання й продажу. Система автоматизованого виробництва (САВ) дозволяє створювати креслення, відразу одержуючи загальний вид об'єкта, управляти верстатами по виготовленню деталей. Гнучка виробнича система (ГВС) дозволяє швидко реагувати на зміну ринкової ситуації, практично миттєво розширювати або звертати

виробництво виробу або замінити його іншим. Легкість переведення конвеєра на випуск нової продукції дає можливість робити безліч різних моделей виробу.

Комп'ютери дозволяють швидко опрацьовувати інформацію від різних датчиків, у тому числі від автоматизованої охорони, від датчиків температури для регулювання витрат енергії на опалення, від банкоматів, що реєструють витрату грошей клієнтами, від складної системи томографа, що дозволяє «побачити» внутрішню будову органів людини й правильно поставити діагноз.

Комп'ютер перебуває на робочому столі фахівця будь-якої професії. Він дозволяє зв'язатися, по спеціальній комп'ютерній пошті, з будь-якою точкою земної кулі, приєднатися до фондів великих бібліотек, не виходячи з будинку, використовувати потужні інформаційні системи — енциклопедії, вивчати нові науки й здобувати різні навички за допомогою навчальних програм і тренажерів. Модельєрові він допомагає розробляти викрійки, видавцеві — компонувати текст та ілюстрації, художникові — створювати нові картини, а композиторів — музику. Дорогий експеримент може бути повністю обчислений та імітований на комп'ютері.

Розробка способів і методів подання інформації, технології розв'язання завдань із використанням комп'ютерів стала найважливішим прикладним аспектом інформатики.

Покоління електронно-обчислювальних машин. Традиційно в історії розвитку електронно-обчислювальних машин виділяють п'ять етапів або п'ять поколінь ЕОМ.

I покоління ЕОМ відносять до початку 50-х років ХХ століття та пов'язують із серійним виробництвом ЕОМ, у яких були реалізовані основні принципи архітектури Джона фон Неймана. У процесорах ЕОМ I покоління використовувалися електронні лампи – діоди й тріоди. До цього покоління можна віднести МЕЛМ (мала електронно-лічильна машина), ВЕЛМ (велика електронно-лічильна машина), М-1, Мінськ-1, Урал-1 тощо. Швидкість таких ЕОМ становила 2-3 тисячі операцій на секунду, а пам'ять – близько 2 тисяч машинних слів.

Слід зауважити, що в СРСР перша ЕОМ – МЕЛМ – була створена під керівництвом Сергія Олександровича Лебедева у 1951 році. Серед творців перших ЕОМ слід пригадати також І.С. Брука, М.А. Карцева, Б.І. Рамєєва, В.М. Глушкова, Ю.А. Базилевського.

II покоління ЕОМ відносять до кінця 50-х і початку 60-х років ХХ століття та пов'язують із появою транзисторів у процесорах і ферритових сердечників в оперативних запам'ятовувальних пристроях. Швидкість ЕОМ зросла до сотень тисяч операцій на секунду, а пам'ять – до десятків тисяч

машинних слів. До ЕОМ II покоління відносять М-40, М-50, Мінськ-2, МІР-1 і т.д. Велика увага в машинах цього покоління приділялася створенню системного програмного забезпечення, компіляторів і засобів вводу-виводу.

III покоління ЕОМ відносять до кінця 60-х початку й 70-х років ХХ століття та пов'язують із появою інтегральних напівпровідникових схем у процесорах. Швидкість ЕОМ зросла до мільйона операцій на секунду, а пам'ять – до сотень тисяч машинних слів. До ЕОМ III покоління відносять ІВМ-360, Дніпро-2, МІР-2 і т.і.

IV поколінням ЕОМ відносять до другої половини 70-х років і початку 80-х років ХХ століття та пов'язують із появою великих інтегральних схем. Швидкість ЕОМ зросла приблизно до 10⁹ операцій на секунду, а пам'ять – до 10⁷ машинних слів. До ЕОМ IV покоління відносять ЕС-1015, Електроніка МС 0501, Іскра-226 і т.і.

Проект ЕОМ V покоління. На початку 80-х років Японія оголосила про початок робіт над обчислювальними системами п'ятого покоління (ОСПП). Проект був розрахований на десять років. Аналогічні проекти розроблялися у Великобританії (проект Елві), США (проект Корпорації), у Західній Європі (проект ЕСПРІ). По баченню розвитку обчислювальної техніки в ті роки, усю її історію можна було поділити на три етапи: обчислювальний, загальноінформаційний і «умовивідний». Останній етап і передбачалося реалізувати в комп'ютерах п'ятого покоління, причому одним з пунктів були дослідження й розробки, пов'язані з комп'ютерними мережами і прикладними системами.

Японський проект припускав створення й широке впровадження комп'ютерів із вбудованим штучним інтелектом, що дозволяють здійснювати спілкування між людиною й комп'ютером природньою мовою, обмін графічною й образотворчою інформацією. Концептуально ОСПП склалися із трьох сфер: 1) сфера людини й прикладних завдань; 2) апаратні засоби й комунікації; 3) програмне забезпечення.

Центральним завданням, яке необхідно було розв'язати – це розробка технологій обробки знань – «гносеоінформації» (knowledge information). Системи керування базами знань (гносеобрази) були засновані на реляційній моделі подання даних. У якості підзавдань було створення баз знань і роботів, що наділені органами почуттів і штучним інтелектом.

Найбільш амбіційною заявою розроблювачів проекту було те, що вони збиралися внести в базу знань усі знання, які нагромадило людство за останні чотириста років.

Проект ОСПП містив 26 тем науково-дослідних і дослідно-конструкторських робіт, об'єднаних в 7 груп:

1) базові прикладні системи (системи машинного перекладу, системи, що відповідають на запитання, прикладні системи розуміння мови, картин і зображень);

2) базові системи програмного забезпечення (системи керування базами знань, вирішення проблем і логічного висновку, інтелектуального інтерфейсу);

3) нові високорозвинені архітектури (машини логічного програмування, функціональні машини, машини реляційної алгебри, машини підтримки даних абстрактного типу, машини потоків даних, інноваційні фон Нейманівські машини);

4) архітектура розподілених функцій (мережна архітектура, машини баз даних, швидкодійні машини чисельних обчислень, система людино-машинного спілкування високого рівня);

5) технологія надвеликих інтегральних схем (СБІС – кремнієвих чипів), що включає розробку архітектури СБІС і інтелектуальну систему САПР (систему автоматичного проектування) СБІС;

6) технологія розробки систем (інтелектуальні системи програмування, система проектування баз знань, системна технологія розробки комп'ютерних архітектур, системи баз даних і розподілених баз даних);

7) технологія забезпечення розробок.

У чистому вигляді проект обчислювальних систем п'ятого покоління так і не був реалізований, і саме через неможливість у той час розв'язати складні завдання штучного інтелекту. Багато які з них, наприклад, створення численних і багатопредметних експертних систем, не вирішені дотепер, розвиток комп'ютерної техніки пішов по шляху мережних структур. Головна причина не настільки успішного завершення проекту – у складності самої людської психіки, нерозкритих таємниць мислення й організації пам'яті людини. Проте, багато підзавдань були вирішені частково й вирішуються в цей час: розпізнавання й конструювання зображень, комп'ютерний переклад, мовні аналізатори, паралельні обчислення й інші.

Заслуговує на особливу увагу, що проект ОСПП розглядався як новий глобальний економічний фактор. Перелічимо його основні наслідки.

1. Одержання прямого ресурсу робочої сили у вигляді «інтелектуальних» роботів, наділених органами почуттів і здатних до «розуміння» своїх завдань, а також до самодіагностики й саморемонту. Таким чином, планування ресурсів робочої сили не залежить від демографічних умов.

2. Поширення досвіду фахівців високого рівня усередині кожної професійної групи через комп'ютерні експертні системи.

3. Підвищення національного науково-дослідного потенціалу шляхом: створення інтелектуалізованих робочих місць із персональними комп'ютерами; заміни реального експерименту машинним; забезпечення оперативного доступу всіх науково-дослідних організацій до всієї необхідної для них інформації.

4. Автоматизація сфери розподілу товарів і послуг.

5. Автоматизація організаційно-розпорядчого керування.

6. Мінімізація «ентропії» усієї економічної системи шляхом усунення непродуктивної витрати матеріалів, енергії, транспорту й інших ресурсів.

Питання і завдання для самостійної роботи

1. Хто винайшов перший арифмометр? Назвіть інші досягнення цього вченого.
2. Скільки символів містить алфавіт машинної мови?
3. Які обчислювальні машини робила (і продовжує робити) фірма ІВМ?
4. Докладно розповісти про відомі Вам застосування комп'ютерів.
5. Перелічіть відмітні риси ЕОМ різних поколінь.
6. У чому суть проекту ЕОМ V покоління?
7. Що таке інформаційна діяльність?
8. Розповісти про принципи роботи Аналітичної машини Ч. Беббіджа.
9. Перелічіть основні етапи розвитку обчислювальної техніки.
10. Яке призначення має система автоматизованого виробництва?
11. Яке призначення має гнучка виробнича система?
12. Яке центральне завдання необхідно розв'язати для створення обчислювальних систем п'ятого покоління?

Список літератури

1. Бауэр Ф.Л., Гооз Г. Информатика. Вводный курс: в 2-х ч. М.: Мир, 1990. – 336 с.
2. Евдокимов В. В. Экономическая информатика: Учебник для вузов. — СПб., 1997.
3. Зацеркляний М.М., Мельников О.Ф., Струков В.М. Основы компьютерных технологий для экономистів: Навчальний посібник. – Київ, 2007. – 672 с.
4. Информатика. Базовый курс / Под ред. С.В. Симоновича. – СПб.: Питер, 2005. – 640 с.

5. Информатика: Учебник / Под ред. Н.В. Макаровой. – 3-е изд., перераб. – М.: Финансы и статистика, 2001.
6. Каймин В.А. Информатика: Учебник. – 2-е изд., перераб. и доп. – М.: ИНФРА-М, 2001. – 272 с.
7. Колин К.К. Информатизация образования и фундаментальные проблемы информатики//
8. Острейковский В.А. Информатика: Учебник для вузов. – М.:Высш.шк., 2000. – 511 с.
9. Практикум по экономической информатике: Учеб. пособ. – Ч.1 / Под ред. Е.Л. Шуремова, Н.А. Тимаковой, Е.А. Мамонтовой. – М.: Перспектива, 2000.
10. Соболев Б.В., Галин А.Б., Панов Ю.В., Рашидова Е.В., Садовой Н.Н. Информатика: учебник. – Изд. 3-е, дополн. И перераб. – Ростов н/Д: Феникс, 2007. – 446 с.
11. Ткаченко В.А. Экономическая информатика: онлайн учебник
12. Цымбал В.П. Теория информации и кодирования. – «Вища школа», 1973. – 232 с.
13. Шаратов О.Д., Дербенцев В.Д., Семьонов Д.С. Економічна кібернетика: Навчальний посібник. – К.: КНЕУ, 2005. – 231 с.
14. Экономическая информатика: Учебник / Под ред. В.П. Косарева и Л.В. Еремина. – М. Финансы и статистика, 2002. – 592 с.
15. Яблонский С.В. Введение в дискретную математику: Учеб. Пособие для вузов. / Под ред. В.А. Садовниченко. – 3-е изд., стер. – М.:Высш. шк.; 2001. – 384 с.

Розділ II. ТЕОРЕТИЧНІ ОСНОВИ ІНФОРМАТИКИ

У цьому розділі наведені основні відомості з теорії алгоритмів, математичної логіки, арифметичних і логічних основ комп'ютерів, систем числення, а також правила переведення з однієї системи числення в іншу; подання цілих і дійсних чисел у комп'ютерах; основні поняття алгебри логіки й встановлюється зв'язок між двійковим кодуванням і алгеброю логіки.

Основи теорії алгоритмів

Поняття алгоритму – одне з основних понять математики та інформатики. Нестроге, інтуїтивне визначення алгоритму настільки ясне, що не виникає сумнівів, коли йдеться про знайдений алгоритм розв'язання деякої конкретної задачі. Однак, якщо задача не вирішена, то при пошуку її розв'язання необхідно або довести відсутність алгоритму, або довести його існування. Для того, щоб довести відсутність алгоритму розв'язання поставленої задачі, необхідно точно уявляти, що таке алгоритм, тобто мати строге визначення алгоритму. Для строгого визначення алгоритму необхідно уточнити поняття елементарного кроку або елементарної дії, порядок виконання елементарних кроків тощо. З цією метою були розроблені різні алгоритмічні моделі, такі як машина Тьюрінга, нормальний алгоритм Маркова, машина Поста, машина з довільним доступом, клас частково рекурсивних функцій і т.і. У теорії алгоритмів доведено, що всі ці алгоритмічні моделі еквівалентні в тому розумінні, що якщо алгоритм може бути реалізований в одній з алгоритмічних моделей, то він може бути реалізований і в будь-якій іншій алгоритмічній моделі.



Рис. 2.1. Алгоритмічні моделі, що уточнюють поняття «алгоритм»

Ми зупинимося на описі двох алгоритмічних моделей – машина Тьюрінга й нормальний алгоритм Маркова. Популярність цих моделей можна пояснити їхньою наочністю. Так, машина Тьюрінга являє собою абстрактний пристрій, роботу якого можна спостерігати крок за кроком, а нормальний алгоритм Маркова реалізує просту ідею – заміну однієї частини слова іншим словом.

2.1. Поняття алгоритму та його властивості. Форми запису алгоритму

Поняття алгоритму належить до фундаментальних математичних понять, що характеризується набором властивостей. Для спрощення розуміння сутності алгоритму академіком А.П. Єршовим було дано наступне визначення.

➤ **Алгоритм** – зрозуміла й точна інструкція виконавцеві реалізувати послідовність дій, спрямованих на досягнення зазначеної мети або на розв'язання поставленої задачі.

Важливим у даному визначенні є суб'єкт, на який орієнтовано виконання алгоритму – деякий «виконавець». Ним може бути як людина, так і неживий пристрій – комп'ютер. Одним суб'єктом алгоритм може бути складений, придуманий, а іншим – виконаний, причому формально, крок за кроком. Виконавець може не знати про суть розв'язуваної задачі, її постановку й предметну галузь, для якої вона розв'язується. Виконавцеві досить уміти виконувати зазначені в алгоритмі послідовності елементарних дій – команди.

Послідовність дій можна назвати алгоритмом, якщо вона має наступні властивості.

1. *Визначеність* – кожна команда повинна бути здійснена виконавцем.
 2. *Ефективність* – кожна дія повинна мати результат /ділення на 0/.
 3. *Скінченність* – алгоритм одержує результат за кінцеве число кроків.
 4. *Дискретність* – алгоритм складається з окремих дій (кроків, команд).
 5. *Масовість* – алгоритм складається для розв'язання класу задач.
 6. *Введення* – алгоритм має вхідні дані, що підлягають обробці (або вихідну ситуацію).
 7. *Вивід* – алгоритм одержує вихідні дані (або кінцеву ситуацію).
- Алгоритм можна подати в різних формах: мовній (як на природній, так і штучній мові) або словесній та графічній.

При *словесній формі подання алгоритму* природньою мовою алгоритм записується у вигляді пронумерованої послідовності кроків. Прикладом словесної форми подання алгоритму природньою мовою є алгоритм Евкліда обчислення найбільшого спільного дільника *nod* двох натуральних чисел x і y :


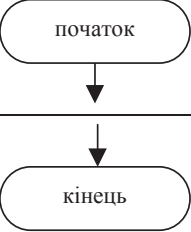

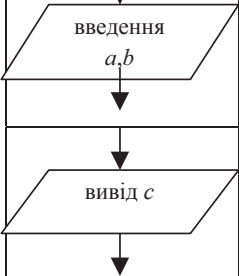

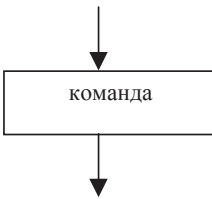
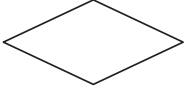
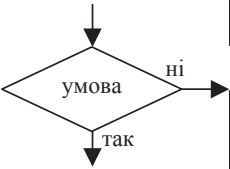
1. Якщо числа рівні, вибрати будь-яке з них у якості відповіді й далі не виконувати алгоритм.
2. Визначити більше із чисел.
3. Замінити більше число різницею більшого й меншого.
4. Почати алгоритм спочатку.

Словесна форма запису має на увазі й можливість запису алгоритму на алгоритмічній (штучній) мові. Наприклад, запис алгоритму Евкліда умовною мовою програмування має такий вигляд.

```
1 крок. ВВЕДЕННЯ двох цілих чисел  $x$  і  $y$ 
2 крок. ЯКЩО  $x=y$ , ТО
     $nod:=x$  // прийняти в якості відповіді будь-яке
    // з чисел  $x, y$ 
    ВИВІД  $nod$ 
    STOP
3 крок. ЯКЩО  $x>y$ , ТО  $x:=x-y$  // визначити більше з
    ІНАКШЕ  $y:=y-x$  // чисел  $x$  і  $y$  і замінити
    // більше із чисел різницею
    // більшого й меншого.
4 крок. ПЕРЕЙТИ до кроку 2.
```

Будь-який алгоритм можна представити *графічно* – у вигляді *блок-схеми*. Для зображення блок-схем використовуються спеціальні геометричні фігури (блоки), з'єднані стрілками, що визначають порядок просування від однієї фігури до іншої. Кожний елемент блок-схеми – це крок алгоритму. Розглянемо основні елементи блок-схеми, перелічені в таблиці № 2.1.1.

Таблиця № 2.1.1. Основні елементи блок-схеми

Вид блоку і його назва	Призначення блоку	Приклад запису
 Блок початку або кінця алгоритму	овали й кола служать для позначення початку й кінця алгоритму, а також для точок «з'єднання» стрілок	
 Блок введення або вивіду	паралелограм використовується для позначення введення й вивіду даних, тобто введення (ініціалізація) або вивід значень змінних	
 Блок дії (виконання команди)	прямокутник використовується для запису команди, що дозволяє змінити значення змінної, обробити інформацію, тому дана фігура іноді називається «обробкою», а символом того, що змінна одержує нове значення, служить знак «:=», названий «присвоюванням»	
 Блок логічної умови	ромб («розпізнаватель») служить для запису деякого відношення або умови; у випадку виконання умови далі треба просуватися по стрілці зі знаком «+» (або словом «так»); якщо умова не виконується – перейти по стрілці зі знаком «-» (або словом «ні»)	

Важливим правилом є обмеження на кількість вихідних з кожної фігури стрілок: крім ромба, у якому їх дві, для всіх інших фігур вихідною може бути тільки одна стрілка.

Алгоритм Евкліда у вигляді блок-схеми представлено на рисунку 2.1.2.

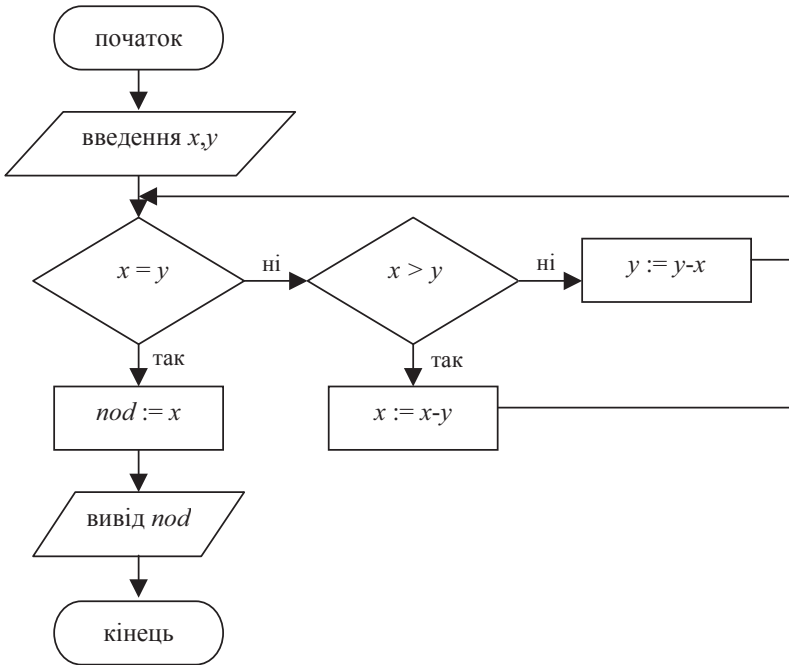


Рис.2.1.2 Блок-схема алгоритму Евкліда обчислення найбільшого спільного дільника

2.2. Алгоритмічна система А. Тьюрінга

Машина Тьюрінга – це абстрактний пристрій, що містить нескінченну, розбиту на секції (комірці) стрічки, пристрій керування й зчитувальну (друкуючу) голівку. У кожній комірці в кожний момент часу може бути записаний рівно один символ із *зовнішнього* алфавіту $A = \{a_0, a_1, a_2, \dots, a_n\}$.

Зовнішній алфавіт містить спеціальний символ a_0 – порожній символ. При розв'язанні задачі звичайно в якості порожнього символу a_0 використовується або 0 або λ . Голівка може розрізняти (розпізнавати) на стрічці символи й при необхідності стирати й уписувати символи в комірці.



Алан Тьюрінг (1912 — 1954) – видатний англійський математик, один з засновників інформатики й теорії штучного інтелекту, його вважають першим теоретиком сучасного програмування й, нарешті, першим у світі хакером (дешифрувальником, зламщиком кодів). Він внес фундаментальний вклад в теорію алгоритмів, коли вигадав знамениту машину Тьюрінга.

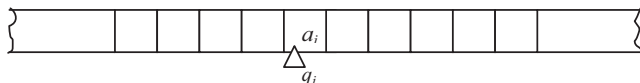
Голівка машини Тьюрінга, зчитуючи символи, може переходити з одного стану в інший. Виділяють початковий q_1 й заключний q_0 стани машини Тьюрінга. Набір символів для позначення внутрішніх станів машини, називається **внутрішнім** алфавітом. Він складається з букв q_0, q_1, \dots, q_r .

Машина Тьюрінга виконує наступні найпростіші операції:

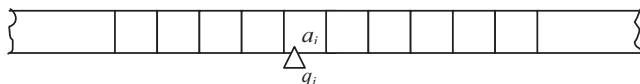
1. може розпізнавати символ, поміщений у комірку;
2. може вписувати символ у порожню комірку;
3. може стирати символ і замінити його іншим;
4. може стирати символ, не вписуючи нового;
5. може здійснювати крок на одну комірку вправо (R), уліво (L) або залишати голівку на місці (S).
6. може змінювати стан голівки з q_i на q_k .

Команду машини Тьюрінга – елементарний крок записують у вигляді п'ятірки значень $q_i a_i q_j a_j d_j$ або $q_i a_i \rightarrow q_j a_j d_j$, де d_j – це символ руху R , L або S . Пояснимо, як слід розуміти команду $q_i a_i q_j a_j d_j$:

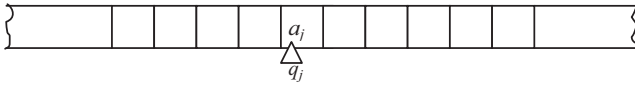
- 1) голівка перебуває в стані q_i й зчитує (розпізнає) символ a_i ;



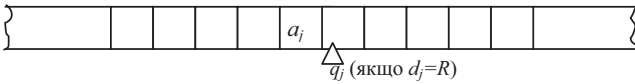
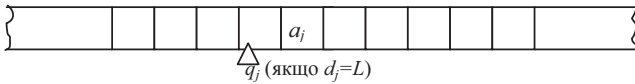
- 2) стирає символ a_i і на його місце записує символ a_j ;



3) голівка змінює стан q_i на стан q_j ;



4) голівка здійснює рух на один символ вліво, якщо $d_i = L$, або на один символ вправо, якщо $d_i = R$, або залишається на місці, якщо $d_i = S$.



Залежно від стану й символу, що спостерігається голівкою, пристрій керування в кожний момент часу в процесі роботи дає «накази» голівці, керуючись командами. Сукупність усіх команд, що описують роботу деякої машини Тьюрінга, називається її *програмою*.

Програма може бути представлена у вигляді таблиці 2.2.1. Команда $q_i a_j d_j$ записується в таблиці в такий спосіб: у комірці, яка перебуває на перетинанні стовпця, відповідного до символу q_i внутрішнього алфавіту й рядка, відповідного до символу a_j зовнішнього алфавіту, записується перетворення $q_j a_j d_j$, де d_j – це символ руху R, L або S .

Таблиця 2.2.1. Подання програми машини Тьюрінга у вигляді таблиці

		символи внутрішнього алфавіту					
		q_1	q_2	...	q_i	...	q_r
символи зовнішнього алфавіту	a_0						
	a_1						
	\vdots						
	a_i				$q_j a_j d_j$		
	\vdots						
	a_n						

Наприклад, програму машини Тьюрінга

$$\begin{cases} q_1 0 q_1 0 R \\ q_1 1 q_2 0 R \\ q_2 1 q_1 0 R \\ q_2 0 q_0 1 S \end{cases}$$

можна записати у вигляді таблиці:

	q_1	q_2
0	$q_1 0 R$	$q_0 1 S$
1	$q_2 0 R$	$q_1 0 R$

Домовимося, що на стрічці машини Тьюрінга споконвічно записане деяке слово, голівка перебуває в початковому стані q_1 й зчитує перший ліворуч символ слова. Говорять, що машина Тьюрінга застосовна до слова, записаного на стрічці, якщо будучи застосованою, до цього слова, машина Тьюрінга, що задається програмою, закінчує обробку слова за кінцеве число кроків. А якщо ні, то, говорять, що машина Тьюрінга не застосовна до слова.

Машина Тьюрінга, що задається програмою, завершує свою роботу (зупиняється) у наступних випадках:

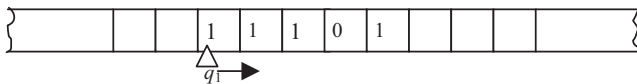
- 1) машина Тьюрінга переходить у заключний стан q_0 ;
- 2) жодна з команд програми, що визначає машину Тьюрінга, не застосовна до слова, записаного на стрічці.

Приклад 2.2.1. З'ясувати, чи застосовна машина Тьюрінга, обумовлена програмою

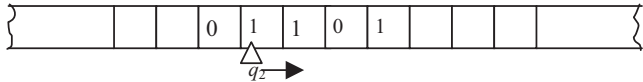
$$\begin{cases} q_1 0 q_1 0 R & (1) \\ q_1 1 q_2 0 R & (2) \\ q_2 1 q_1 0 R & (3) \\ q_2 0 q_0 1 S & (4) \end{cases}$$

до слова 11101.

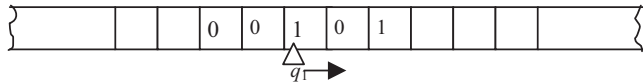
Розв'язання. На стрічці машини Тьюрінга записане слово 11101, голівка машини Тьюрінга зчитує першу ліворуч одиницю в слові 11101 і перебуває в стані q_1 , тобто $q_1 11101$.



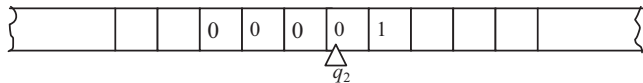
Єдина команда програми, яка застосовна до слова, коли голівка перебуває в стані q_1 й зчитує одиницю – це друга команда q_11q_20R програми. Команда пропонує одиницю, яка зчитується голівкою, замінити нулем, стан q_1 замінити станом q_2 і здійснити рух (зрушити голівку) на одну комірку вправо. Результат застосування команди запишемо у вигляді: $0q_21101$.



Після застосування команди голівка перебуває в стані q_2 й зчитує одиницю, отже, застосовна тільки третя команда q_21q_10R програми. Команда пропонує одиницю, яка зчитується, замінити нулем, стан q_2 замінити станом q_1 і здійснити рух на одну комірку вправо. Результат застосування команди – $00q_1101$.



Після застосування команди голівка перебуває в стані q_1 й зчитує одиницю, отже, застосовна тільки друга команда програми. Команда пропонує одиницю, яка зчитується голівкою, замінити нулем, стан q_1 замінити станом q_2 і здійснити рух на одну комірку вправо. Результат застосування команди – $000q_201$.



Після застосування команди голівка перебуває в стані q_2 й зчитує нуль, отже, застосовна тільки четверта команда q_20q_01S програми. Команда пропонує нуль, що зчитується, замінити одиницею, стан q_2 замінити заключним станом q_0 і залишити голівку на місці. Результат застосування команди – $000q_011$. Машина Тьюрінга, обумовлена програмою, завершує роботу, перетворивши за кінцеве число кроків слово 11101 в слово 00011 .

Приклад 2.2.2. Написати програму машини Тьюрінга, яка переносить усі палички, що розміщені праворуч від символу * вліво, стирає символ * і зупиняється.

Розв'язок. Заданий зовнішній алфавіт $A = \{\lambda, |, *\}$. Опишемо покрокові дії, які будуть реалізовані в програмі. У початковий момент голівка машини Тьюрінга перебуває в стані q_1 і зчитує першу ліворуч паличку слова.

1. Зітремо рівно одну паличку спочатку слова. Для цього замінимо паличку | порожнім словом λ , змінимо стан q_1 на новий стан q_2 і здійснимо рух на одну позицію вправо, тобто виконаємо команду $q_1 | q_2 \lambda R$.

2. У стані q_2 будемо рухатися вправо, не змінюючи зчитувані символи ($|, *$), поки не досягнемося кінця слова, тобто символу λ . Іншими словами, буде виконено команду $q_2 | q_2 | R$ і $q_2 * q_2 * R$.

3. Рівно один символ λ наприкінці слова заміняємо паличкою. Для цього необхідно замінити стан q_2 новим станом q_3 і здійснити рух на одну позицію вліво, тобто виконати команду $q_2 \lambda q_3 | L$.

4. У стані q_3 рухаємося вліво, залишаючи незмінними символи паличок і * доти, поки в стані q_3 голівка не буде оглядувати перший ліворуч символ λ , тобто виконуються команди $q_3 | q_3 | L$ і $q_3 * q_3 * L$.

5. Голівка в стані q_3 оглядає перший ліворуч символ λ , залишає цей символ без зміни, змінює стан на q_1 і здійснює рух на одну позицію вправо, тобто виконується команда $q_3 \lambda q_1 \lambda R$.

6. Якщо в стані q_1 голівка оглядає паличку, то слід повторити всі дії, починаючи з першого кроку, а якщо голівка в стані q_1 оглядає зірочку, то всі палички ліворуч від зірочки були перенесені вправо, а, виходить, потрібно стерти зірочку й зупинитися, тобто виконати команду $q_1 * q_0 \lambda S$.

Представимо програму у вигляді наступної таблиці.

символи внутрішнього алфавіту				
символи зовнішнього алфавіту		q_1	q_2	q_3
	λ		$q_3 L$	$q_1 \lambda R$
		$q_2 \lambda R$	$q_2 R$	$q_3 L$
	*	$q_0 \lambda S$	$q_2 * R$	$q_3 * L$

Продемонструємо роботу програми для слова $\|\ast\|$, записаного на стрічці. Результатом роботи програми повинне бути слово $\|\|\|$

$$\lambda q_1 \|\ast\| \lambda \quad (1) \qquad \lambda \lambda \lambda q_2 \ast \|\|\lambda \quad (11)$$

$$\lambda \lambda q_2 \|\ast\| \lambda \quad (2) \qquad \lambda \lambda \lambda \ast q_2 \|\|\lambda \quad (12)$$

$$\lambda \lambda \|\ast q_2\| \lambda \quad (3) \qquad \lambda \lambda \lambda \ast \|\ast q_2\| \lambda \quad (13)$$

$$\lambda \lambda \|\ast q_2\| \lambda \quad (4) \qquad \lambda \lambda \lambda \ast \|\ast q_2\| \lambda \quad (14)$$

$$\lambda \lambda \|\ast\| q_2 \lambda \quad (5) \qquad \lambda \lambda \lambda \ast \|\ast q_3\| \lambda \quad (15)$$

$$\lambda \lambda \|\ast q_3\| \lambda \quad (6) \qquad \lambda \lambda \lambda \ast q_3 \|\|\lambda \quad (16)$$

$$\lambda \lambda \|\ast q_3\| \lambda \quad (7) \qquad \lambda \lambda \lambda q_3 \ast \|\|\lambda \quad (17)$$

$$\lambda \lambda q_3 \|\ast\| \lambda \quad (8) \qquad \lambda \lambda q_3 \lambda \ast \|\|\lambda \quad (18)$$

$$\lambda q_3 \lambda \|\ast\| \lambda \quad (9) \qquad \lambda \lambda \lambda q_1 \ast \|\|\lambda \quad (19)$$

$$\lambda \lambda q_1 \|\ast\| \lambda \quad (10) \qquad \lambda \lambda \lambda q_0 \lambda \|\|\lambda \quad (20)$$

Питання і завдання для самостійної роботи

1. Написати програму машини Тьюрінга, що перетворить слово з n одиниць у слово, що містить n одиниць і нуля, за якими розміщені n одиниць, при $n \geq 1$.
2. Написати програму машини Тьюрінга, що перетворить слово з n нулів, за якими розміщені n одиниць у слово, що містить n слів виду 01, при $n \geq 1$.
3. Написати програму машини Тьюрінга, що перетворить слово, що містить n одиниць, за якими розміщений нуль і m одиниць у слово, що містить m одиниць і нуль, за якими розміщені n одиниць, при $n \geq 1$ й $m \geq 1$.
4. Написати програму машини Тьюрінга, що перетворює слово з n ($n \geq 1$) паличок у слово з $2n$ паличок.
5. Написати програму машини Тьюрінга, що реалізує алгоритм Евкліда обчислення найбільшого загального дільника двох натуральних чисел.

6. Написати програму машини Тьюрінга, що реалізує додавання двох натуральних чисел в алфавіті $\{\lambda, |, *\}$ (кожне із чисел задається набором паличок, символ $*$ відокремлює представлення одного натурального числа від іншого).

7. З'ясувати, чи застосовна машина Тьюрінга, обумовлена програмою P , до слова F .

$$7.1. P = \begin{cases} q_1 0 q_2 1 R \\ q_1 1 q_2 1 L \\ q_2 0 q_3 1 R \\ q_2 1 q_3 0 R \\ q_3 1 q_1 1 R \end{cases}$$

a) $F = 111011$

b) $F = 11111$

$$7.2. P = \begin{cases} q_1 0 q_1 1 R \\ q_1 1 q_2 0 R \\ q_2 0 q_1 1 R \\ q_2 1 q_3 1 L \\ q_3 0 q_1 1 L \end{cases}$$

a) $F = 10101$

b) $F = 10011$

$$7.3. P = \begin{cases} q_1 0 q_1 1 R \\ q_1 1 q_2 1 R \\ q_2 0 q_3 0 R \\ q_2 1 q_1 0 L \\ q_3 0 q_2 1 S \end{cases}$$

a) $F = 111001$

b) $F = 100101$

8. Нехай на стрічці машини Тьюрінга записане число n у десятковій системі числення, тобто n – слово над алфавітом $\{\lambda, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$. Написати програму машини Тьюрінга, що збільшує число n на 2.

9. Опишіть принцип роботи машини Тьюрінга.

10. Перелічити властивості алгоритму.

2.3. Алгоритмічна система А.А. Маркова



Андрій Андрійович Марков [молодший] (1903 — 1979) – видатний радянський математик, член-кореспондент АН СРСР, який є автором нормального алгоритму (алгоритму), засновником школи конструктивної математики і логіки в СРСР.

Розглянемо поняття нормального алгоритму, запропонованого Андрієм Андрійовичем Марковим. Уся вихідна й проміжна інформація про результати перетворення задається у вигляді слів деякого кінцевого алфавіту $A = \{a_1, a_2, \dots, a_n\}$. З букв утворюються слова, є порожнє слово λ .

Перетворення по Маркову полягає в тому, що частина слова замінюється іншим словом у даному алфавіті. Нехай R, P, Q – слова, складені із символів алфавіту A . Побудуємо слово $S=RPQ$, отримане в результаті злиття слів R, P і Q відповідно. Слово P називають початковим відрізком слова S або префіксом, а слово Q називають кінцевим відрізком слова S або постфіксом. Нехай T – слово, складене із символів алфавіту A , тоді говорять, що слово $\tilde{S}=RTQ$ отримане зі слова $S=RPQ$ шляхом заміни слова P на слово T або в результаті виконання підстановки слова T замість P . Застосовану операцію підстановки записують у вигляді $P \rightarrow T$.

Для виконання алгоритму, записаного в системі А.А.Маркова, необхідно визначити *входження*: чи має місце наявність даного слова (підстроки) в іншому слові (рядку) і скільки раз підстрока входить у рядок. *Розпізнавання входження* – перша елементарна операція. *Підстановка* – друга елементарна операція: $P \rightarrow T$ (замінити P на T). Підстановка $\rightarrow T$ полягає в тому, що розпізнається *порожнє* слово λ й замість нього здійснюється підстановка слова T . Перше входження порожнього слова – це порожнє слово λ перед даним словом. Визначимо підстановку спеціального виду, названу *заключною підстановкою* $P \rightarrow \circ T$. Така підстановка виконується тільки один раз. Символи « \rightarrow » і « \circ » – це додаткові символи, що не належать алфавіту A .

Алгоритми, складені з елементарних операцій «розпізнавання входження» і «підстановка» називаються **нормальними алгоритмами**.

Сконструювати нормальний алгоритм – значить вписати в певному порядку одну підстановку за іншою, тобто вписати схему підстановок виду:

$$\begin{aligned} P_1 &\rightarrow T_1 \\ P_2 &\rightarrow T_2 \\ &\dots \\ P_m &\rightarrow \circ T_m \end{aligned}$$

де $P_1, P_2, \dots, P_m, T_1, T_2, \dots, T_m$ – слова над алфавітом A .

При складанні нормального алгоритму Маркова деякі задачі не вдається розв'язати без введення додаткових символів – розширення алфавіту A . При введенні деякого додаткового символу α за допомогою підстановки $\lambda \rightarrow \alpha$ намагаються цю підстановку записати самою останньою в системі підстановок, щоб забезпечити виконання цієї підстановки рівно один раз. Таким чином, при написанні нормального алгоритму Маркова часто формування системи підстановок відбувається «знизу нагору». Звичайно на самому початку системи підстановок розташовані підстановки, що

дозволяють завершити виконання алгоритму, далі всі інші підстановки, що складають тіло програми, а аж унизу розташовані підстановки, відповідальні за ініціалізацію – устанавлення початкових значень.

Застосувати вже складений алгоритм до даного слова – це значить:

1. переглянути слово ліворуч праворуч і виконати першу застосовну підстановку із числа підстановок алгоритму;
2. процес застосування підстановки здійснювати доти, поки має місце хоча б одна зі звичайних підстановок або заключна.

Іноді алгоритм може виявитися *непридатним* до даного слова. Це буває, якщо процес застосування підстановки до заданого слова триває нескінченно.

Нормальний алгоритм Маркова, будучи застосовним до деякого слова, завершує роботу в наступних випадках:

1. жодна з підстановок не застосовна жодного разу;
2. була застосована заключна підстановка.

Приклад 2.3.1. Нехай дане слово $acbabba$ в алфавіті $A = \{a, b, c\}$ й заданий нормальний алгоритм Маркова, представлений у вигляді впорядкованої послідовності підстановок:

$$ab \rightarrow ba \quad (1)$$

$$ac \rightarrow ab \quad (2)$$

$$aa \rightarrow bc \quad (3)$$

Потрібно з'ясувати, яке слово буде результатом застосування заданого нормального алгоритму до даного слова.

Розв'язок. Переглядаємо слово $acbabba$ ліворуч праворуч. Розпізнаємо входження в задане слово слова ab – це означає, що застосовна перша підстановка (ab – слово в лівій частині першої підстановки). У результаті застосування першої підстановки одержимо слово $acbbaba$. Переглядаємо отримане слово $acbbaba$ ліворуч праворуч і знову розпізнаємо входження в це слово слова, розташованого в лівій частині першої підстановки. Після застосування першої підстановки одержимо слово $acbbbaa$. Переглядаємо слово $acbbbaa$ ліворуч праворуч. Перша підстановка не застосовна, тому що слово ab у слові $acbbbaa$ не міститься. Розпізнаємо входження слова ac у слово $acbbbaa$, а це означає, що застосовна друга підстановка нормального алгоритму. Її результатом буде слово $abbbbaa$. Далі, до слова $abbbbaa$ знову буде застосовна перша підстановка і т.і. Нормальний алгоритм завершить свою роботу, коли ні одну з підстановок застосувати буде не можна. Увесь процес перетворення вихідного слова $acbabba$ із вказівкою номерів застосовних підстановок представлений нижче.

$acbabba \xrightarrow{(1)} acbbaba \xrightarrow{(1)} acbbbaa \xrightarrow{(2)} abbbbaa \xrightarrow{(1)} babbbbaa$
 $\xrightarrow{(1)} bbabbaa \xrightarrow{(1)} bbbabaa \xrightarrow{(1)} bbbbaaa \xrightarrow{(3)} bbbbbca$

Заданий нормальний алгоритм перетворить слово $acbabba$ у слово $bbbbca$

Приклад 2.3.2. Нехай дане слово $bcacabc$ в алфавіті $A = \{a, b, c\}$ й заданий нормальний алгоритм Маркова, представлений у вигляді впорядкованої послідовності підстановок:

$$cb \rightarrow cc \quad (1)$$

$$cca \rightarrow ab \quad (2)$$

$$ab \rightarrow bca \quad (3)$$

Потрібно з'ясувати, чи можна застосувати заданий нормальний алгоритм до слова $bcacabc$.

Розв'язок. Переглядаємо ліві частини підстановок і «розпізнаємо» входження слова ab у вихідне слово $bcacabc$. Це означає, що до вихідного слова застосовна єдина підстановка з номером (3). Результатом застосування цієї підстановки буде слово $bcabcac$. Запишемо весь процес перетворення вихідного слова із вказівкою номерів застосованих підстановок.

$bcacabc \xrightarrow{(3)} bcabcac \xrightarrow{(1)} bcaccac \xrightarrow{(2)} bcacabc$ - одержали вихідне слово, до якого циклічно будуть застосовуватися підстановки (3), (1), (2), при цьому процес ніколи не зупиниться. Це означає, що нормальний алгоритм Маркова не можна застосувати до слова $bcacabc$

Приклад 2.3.3. Подано деяке число у десятковій системі числення, тобто слово в алфавіті $A = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 0\}$. Потрібно сконструювати нормальний алгоритм, застосування якого до довільного слова над алфавітом A , дозволяє отримати число, яке більше на 1.

Розв'язок. Запишемо систему підстановок, що збільшує, наприклад, число 18 на одиницю. Для цього застосуємо підстановку $\rightarrow *$, яка означає замінити перше ліворуч порожнє слово символом *. Оскільки потрібно, щоб ця підстановка була здійснена рівно один раз до виконання всіх інших підстановок у нормальній схемі, розмістимо цю підстановку в самому кінці системи підстановок. Результатом застосування підстановки до слова 18 буде слово *18. Будемо зрушувати символ * уліво доти поки не досягнемся кінця слова. Це можна зробити підстановкою $*8 \rightarrow 8*$. У загальному вигляді, таку підстановку можна записати так $*\varepsilon \rightarrow \varepsilon *$, де $\varepsilon = 1, 2, 3, 4, 5, 6, 7, 8, 9, 0$. Наступною застосовною підстановкою $* \rightarrow +$ відзначимо молодший розряд, який повинен бути збільшений на одиницю. Якщо молодший розряд дорівнює значенню від 0 до 8, то він буде збільшений на одиницю, і нормальний алгоритм завершить роботу. Реалізуємо цю дію у вигляді однієї з

наступних заключних підстановок: $0+ \rightarrow \circ$ 1,
 $1+ \rightarrow \circ$ 2, $2+ \rightarrow \circ$ 3, $3+ \rightarrow \circ$ 4, $4+ \rightarrow \circ$ 5, $5+ \rightarrow \circ$ 6, $6+ \rightarrow \circ$ 7, $7+ \rightarrow \circ$ 8,
 $8+ \rightarrow \circ$ 9. Якщо ж молодший розряд рівний 9, то додавання одиниці
приводить до перенесення одиниці в старший розряд. Реалізуємо цю дію у
вигляді підстановки $9+ \rightarrow +0$. Для випадку, коли у всіх розрядах дев'ятки
введемо заключну підстановку $+ \rightarrow \circ$ 1.

Повністю нормальний алгоритм Маркова має вигляд:

$$* \varepsilon \rightarrow \varepsilon * \quad (1)$$

$$* \rightarrow + \quad (2)$$

$$0+ \rightarrow \circ 1 \quad (3)$$

$$1+ \rightarrow \circ 2 \quad (4)$$

$$2+ \rightarrow \circ 3 \quad (5)$$

$$3+ \rightarrow \circ 4 \quad (6)$$

$$4+ \rightarrow \circ 5 \quad (7)$$

$$5+ \rightarrow \circ 6 \quad (8)$$

$$6+ \rightarrow \circ 7 \quad (9)$$

$$7+ \rightarrow \circ 8 \quad (10)$$

$$8+ \rightarrow \circ 9 \quad (11)$$

$$9+ \rightarrow +0 \quad (12)$$

$$+ \rightarrow \circ 1 \quad (13)$$

$$\rightarrow * \quad (14)$$

Приклади роботи алгоритму:

$$1) 18 \xrightarrow{(14)} *18 \xrightarrow{(1)} 1*8 \xrightarrow{(1)} 18* \xrightarrow{(2)} 18+ \xrightarrow{(11)} 19$$

$$2) 299 \xrightarrow{(14)} *299 \xrightarrow{(1)} 2*99 \xrightarrow{(1)} 29*9 \xrightarrow{(1)} 299* \xrightarrow{(2)} 299+ \xrightarrow{(12)} 29+0 \xrightarrow{(12)} 2+00 \xrightarrow{(5)} 300$$

Питання і завдання для самостійної роботи

1. Написати нормальний алгоритм Маркова для сортування по зростанню слова над алфавітом $\{0,1,2,3\}$.

2. Написати нормальний алгоритм Маркова, що витирає слово в даному алфавіті $A = \{a_1, a_2, \dots, a_n\}$.
3. В алфавіті $A = \{a, b, c\}$ задане слово кінцевої довжини. Потрібно сконструювати нормальний алгоритм Маркова, застосування якого до даного слова дозволить отримати слово, що містить усі букви a у його правій частині.
4. В алфавіті $A = \{ |, +, * \}$ задане слово, що містить набір з кінцевого числа паличок $|$. Сконструювати нормальний алгоритм Маркова, застосування якого до даного слова, яке містить парне число паличок, дозволить отримати слово $+$, а застосування цього ж алгоритму до слова, яке містить непарне число паличок – до слова $*$.
5. Нехай заданий нормальний алгоритм Маркова, представлений у вигляді впорядкованої послідовності підстановок:

$$ab \rightarrow c$$

$$cb \rightarrow a$$

$$a \rightarrow \circ\lambda$$

і слово в алфавіті $A = \{a, b, c\}$

a) *abbacb*; b) *cbbabcb*; c) *abcbababb*

Потрібно з'ясувати, яке слово буде результатом застосування цього нормального алгоритму до заданого слова.

6. Нехай задано два натуральні числа, реалізовані у вигляді слів з відповідним набором паличок. Слова розділені знаком $+$. Сконструювати нормальний алгоритм Маркова, що реалізує додавання двох натуральних чисел.
7. Сконструювати нормальний алгоритм Маркова, що дозволяє перевірити, чи є задане слово паліндромом чи ні.

Комп'ютерна арифметика і логіка

Розглянемо арифметичні й логічні основи комп'ютерів, систем числення, а також правила переведення з однієї системи числення в іншу; представлення цілих і дійсних чисел у пам'яті комп'ютера; основні поняття алгебри логіки, а також встановимо зв'язок між двійковим кодуванням і алгеброю логіки.

2.4. Система числення

➤ **Системою числення** називається спосіб найменування й запису чисел за допомогою знаків.

Будь-яке число може бути представлено у вигляді послідовності знаків (цифр) у даній системі числення. Такі знаки (цифри) називаються базисними числами (знаками, цифрами).

Існують позиційні й непозиційні системи числення. У позиційних системах числення значення числового знака залежить від його розташування (позиції) у записі числа, а в непозиційних системах – не залежить.

Число одиниць будь-якого розряду, що об'єднуються в одиницю більш старшого розряду, називається **основою** позиційної системи числення й позначається q . Основа позиційної системи – це не що інше, як кількість різних базисних цифр, які можуть бути використані для запису числа в даній системі числення. У дванадцятиричній системі, наприклад, існували наступні назви розрядів: одиниця, дюжина (12), гросс (144).

Якщо основа $q \leq 10$, то в якості базисних чисел використовуються традиційні арабські цифри від 0 до 9. Наприклад, у двійковій системі числення базисними числами будуть 0 і 1, у десятковій – 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Якщо основа $q > 10$, то використовують спеціальні знаки, найчастіше – латинські букви від A до Z . Наприклад, базисними числами в шістнадцятиричній системі числення є 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A (10), B (11), C (12), D (13), E (14), F (15).

Перетворення із системи числення з основою q у десяткову систему числення

У десятковій системі будь-яке число можна уявити у вигляді багаточлена по ступенях основи:

$$N_{10} = a_{n-1} \cdot 10^{n-1} + \dots + a_1 \cdot 10^1 + a_0 \cdot 10^0 + a_{-1} \cdot 10^{-1} + \dots + a_{-m} \cdot 10^{-m},$$

де $a_{n-1}, \dots, a_0, a_{-1}, \dots, a_{-m}$ – базисні числа десяткової системи числення.

Аналогічно, для будь-якої системи числення з основою q справедливе представлення:

$$N_q = \underbrace{a_{n-1} \cdot q^{n-1} + \dots + a_1 \cdot q + a_0}_{\text{ціла частина}} + \underbrace{a_{-1} \cdot q^{-1} + a_{-2} \cdot q^{-2} + \dots + a_{-m} \cdot q^{-m}}_{\text{дробова частина}},$$

де $a_{n-1}, \dots, a_0, a_{-1}, \dots, a_{-m}$ – базисні числа системи числення з основою q .

Це представлення, у дійсності, являє собою правило перетворення із системи числення з основою q у систему числення з основою 10 ($q \rightarrow 10$).

Правило перетворення із системи счислення з основою q у десяткову систему числення. Для перетворення числа із системи числення з основою q у десяткову систему, необхідно записати число по ступеням основи і виконати арифметичні дії у десятковій системі числення. При цьому слід зазделегідь домовитися з якою точністю буде виконено перетворення дробової частини числа, для цього необхідно зазначити кількість цифр дробової частини.

Приклад 2.1. 1) Перетворити число 101001 із двійкової системи числення в десяткову ($q=2 \rightarrow 10$);

2) Перетворити число 201 із трійкової системи числення в десяткову ($q=3 \rightarrow 10$);

3) Перетворити число 3F із шістнадцятиричної системи числення в десяткову ($q=16 \rightarrow 10$);

4) Перетворити число 1202,11 із трійкової системи числення в десяткову ($q=3 \rightarrow 10$) з точністю до 0,01;

5) Перетворити число 44,04 із п'ятиричної системи числення в десяткову ($q=5 \rightarrow 10$) з точністю до 0,1.

Розв'язок. 1) $N_2 = 101001_2 = 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 32 + 8 + 1 = 41_{10}$

2) $N_3 = 201_3 = 2 \cdot 3^2 + 0 \cdot 3^1 + 1 \cdot 3^0 = 2 \cdot 9 + 1 = 19_{10}$

3) $N_{16} = 3F_{16} = 3 \cdot 16^1 + 15 \cdot 16^0 = 48 + 15 = 63_{10}$

$$4) N_3 = 1202,1 \underset{3}{3}_3 = 1 \cdot 3^3 + 2 \cdot 3^2 + 0 \cdot 3^1 + 2 \cdot 3^0 + 1 \cdot 3^{-1} + 3 \cdot 3^{-2} = 27 + 2 \cdot 9 + 2 \cdot 1 + 1 \cdot 1/3 + 3 \cdot 1/9 \approx 47 + 0,33 + 0,33 = 47,66_{10}$$

$$5) N_5 = 44,0 \underset{5}{4}_5 = 4 \cdot 5^1 + 4 \cdot 5^0 + 0 \cdot 5^{-1} + 4 \cdot 5^{-2} = 20 + 4 + 4 \cdot 1/25 \approx 24,2_{10}$$

Перетворення з десятичної системи числення в систему числення з основою q

Для перетворення цілих чисел з *десятичної системи числення в систему з основою q* ($10 \rightarrow q$) використовується алгоритм Евкліда послідовного ділення. **Правило перетворення цілого числа з десятичної системи числення в систему числення з основою q** представлено у вигляді блок-схеми на рисунку 2.4.1.

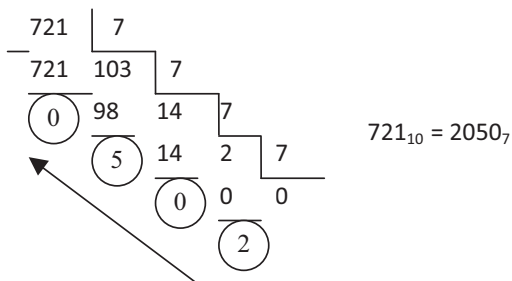
Перетворення десятичного дробу в дріб з основою q із заданою точністю здійснюється по алгоритму послідовного множення. Необхідно помножити заданий дріб на нову основу. Ціла частина отриманого добутку – це перша цифра нового дробу (після коми). Для одержання наступної цифри шуканого дробу множимо дробову частину отриманого добутку на основу q . Ціла частина результату і є наступна цифра. Процес триває доти, поки не буде досягнута заздалегідь застережена точність. **Правило перетворення десятичного дробу в дріб з основою q** представлено у вигляді блок-схеми на рисунку 2.4.2.

Приклад 2.4.1.

1) Перетворити число 721 з десятичної системи числення в сімичинну ($10 \rightarrow 7$);

2) Перетворити число 0,72 з десятичної системи в трійкову систему числення.

Розв'язання. Ділимо число на нову основу, одержуючи цілу частку й остачу (у цьому випадку частка рівна 103, остача рівна 0, але її треба записати). Якщо отримана частка більше 7 (у нас $103 > 7$), ділимо цю частку знову на величину основи й записуємо нові частку й остачу (частка рівна 14, остача 0). Оскільки частка знову більше основи, ділимо її на основу. Одержуємо частку 2 і остачу 0. Процес завершений. Випишемо цифри в наступному порядку: першою – остання частка, а за нею – усі остачі від останнього до першого. Отриманий результат можна перевірити, використовуючи перше правило перетворення із системи з основою q в десятикову.



2)

$0,72 \cdot 3 = 2,16$	(2)	↓	$0,72_{10} \approx 0,201_3$
$0,16 \cdot 3 = 0,48$	(0)		
$0,48 \cdot 3 = 1,44$	(1)		

Питання і завдання для самостійної роботи

1. Перетворіть число а) 11100110, б) 0111000111, с) 10101100110110, д) 10011, е) 001100100110, ф) 101100011000 із двійкової системи числення в десяткову, шістнадцятиричну й восьмиричну системи числення.
2. Перетворіть число а) D5A83F, б) 1A7F, с) E2B2F, д) 7AC4, е) 79A5B із шістнадцятиричної системи числення у восьмиричну, десяткову й п'ятиричну системи числення.
3. Перетворіть число а) 128, б) 1113, с) 2879, д) 333 з десятикової системи числення у двійкову, сімричну, дев'ятиричну системи числення.
4. Перетворіть число а) 301, б) 311, с) 344, д) 57 з восьмиричної системи числення в шістнадцятиричну, десяткову й сімричну системи числення.
5. Перетворіть число 32,98 з десятикової системи числення в п'ятиричну систему числення.
6. Перетворіть число 37, 25 з десятикової системи числення у двійкову, восьмиричну й шістнадцятиричну систему числення.

7. Порівняйте два числа: а) 1001112 і 338; б) 33114 і 5027; с) 2020023 і 22416;
 д) $A24F_{16}$ і 10011001012; е) 44235 і 15417; ф) $2D_{16}$ і 509.
8. Розташувати наступні числа в порядку збування (незростання):
 а) $15B_{16}$, 1117, 1001000002, 111203, 10334, 305, 7548, 12006;
 б) 1228, 236, 1000111012, $F78AB_{16}$, 203617, 30014, 2213;
 с) 5627, $A71C_{16}$, 00110101112, 13104, 305106, 2023;
 д) 132014, 01010111012, 10768, 40325, 60007;
 е) 101000011110110102, 706589, 5768, 3012034, $A3C_{16}$;
 ф) 7082139, $9FA616$, 70128, 1002213, 413216, $E84A_{16}$.

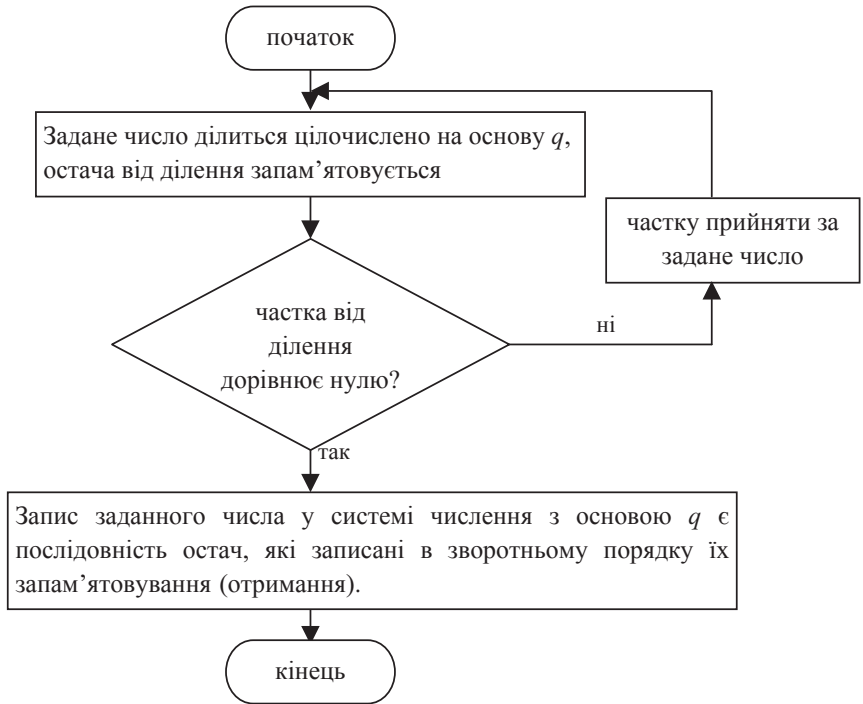


Рис.2.4.1. Правило перетворення цілого числа з десяткової системи числення в систему числення з основою q

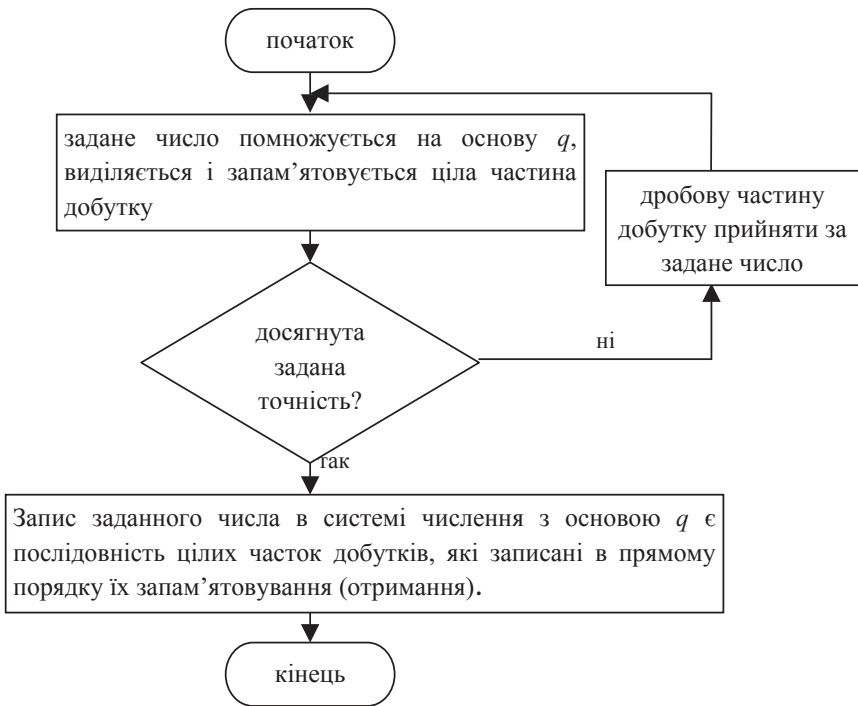


Рис.2.4.2. Правило перетворення дробового числа з десяткової системи числення в систему числення з основою q із заданою точністю

2.5. Обробка чисел в комп'ютері

Комп'ютер виконує операції над числами, записаними у двійковій системі числення, і використовує правила двійкового додавання й двійкового множення. представлені в вигляді наступних таблиць додавання і множення.

Таблиця додавання

+	0	1
0	0	1
1	1	10

Таблиця множення

*	0	1
0	0	0
1	0	1

Комп'ютер виконує перетворення чисел з десяткової системи у двійкову й обернено. Для того щоб «розуміти» машинне слово були розроблені восьмирична й шістнадцятирична системи числення. Ці системи числення більш «ощадливі» у тому розумінні, що восьмирична вимагає в 3 рази менше розрядів для представлення числа в порівнянні із двійковою, а шістнадцятирична – в 4 рази менше розрядів для запису числа. У той же час перетворення з восьмиричної й шістнадцятиричної систем числення у двійкову виконується дуже просто – слід замінити кожен цифру числа у восьмиричній (шістнадцятиричній) системі числення еквівалентним представленням довжини 3 (4) цієї цифри у двійковій системі числення. Також легко виконати й зворотне перетворення із двійкової системи числення у восьмиричну або шістнадцятиричну систему. Для перетворення числа із двійкової системи у восьмиричну (шістнадцятиричну) слід розбити двійкове представлення числа праворуч ліворуч на двійкові слова довжини 3 (4) і замінити їхніми еквівалентними представленнями у восьмиричній (шістнадцятиричній) системах числення.

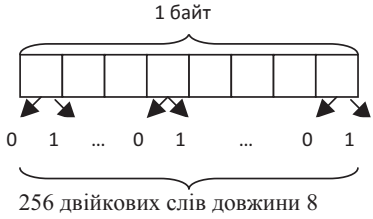
Приклад 2.5.1. Перетворити числа 375_8 і $3F_{16}$ з восьмиричної й шістнадцятиричної систем числення відповідно у двійкову систему числення.

Розв'язок. Перетворимо число 375_8 з восьмиричної системи числення у двійкову. Двійкове представлення довжини 3 для числа 3 має вигляд 011, двійкове представлення довжини 3 для числа 7 має вигляд 111, двійкове представлення довжини 3 для числа 5 має вигляд 101. Таким чином, $375_8=011111101_2$.

Перетворимо число $3F_{16}$ із шістнадцятиричної системи числення у двійкову. Двійкове представлення довжини 4 числа 3 має вигляд 0011, двійкове представлення довжини 3 числа 15 (F) має вигляд 1111. Таким чином, $3F_{16}=00111112_2$.

Представлення цілих чисел в комп'ютері

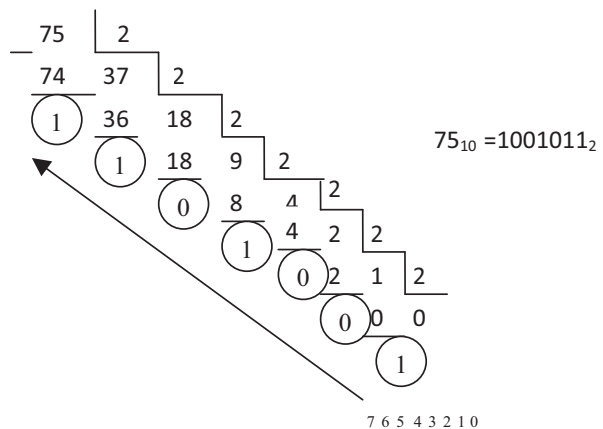
Розрізняють цілі числа без знака й зі знаком. Для представлення цілих чисел у пам'яті комп'ютера використовують звичайно один байт (вісім біт) або два байти (шістнадцять біт). За допомогою 8 біт можна записати $2^8=256$ двійкових слів довжини 8.



При представленні цілих чисел без знака в пам'яті комп'ютера за допомогою 8 біт можна «закодувати» 256 чисел у діапазоні від 0 до $2^8-1=255$. При представленні цілих чисел без знака в пам'яті комп'ютера за допомогою 2 байт або 16 біт можна «закодувати» 2^{16} або 65536 цілих чисел у діапазоні від 0 до $2^{16}-1=65535$.

Приклад 2.5.2. Представити число 75_{10} в однобайтовому і двобайтовому форматі.

Розв'язок. Використовуємо правило перетворення з десятикової системи числення в q -ічну систему числення, де $q=2$.



Число 75_{10} в однобайтовому форматі має вигляд: 01001011_2 .

Число 75_{10} у двобайтовому форматі має вигляд: 0000000001001011_2 .

містить інформацію про знак числа. Якщо в старшому розряді записаний нуль, то число позитивне або нуль, якщо в старшому розряді записана одиниця, то число негативне. Такий запис двійкового числа називається *прямим кодом*. За допомогою 8 біт або 1 байта також можна «закодувати» 256 чисел, перша половина яких буде використана для представлення негативних чисел, а друга половина – для представлення нуля й позитивних цілих чисел, тобто чисел у діапазоні від -2^7 або -128 до $+2^7-1$ або $+127$. Аналогічно, за допомогою 16 біт або 2 байт можна представити в пам'яті комп'ютера числа від -2^{15} або -32768 до $+2^{15}-1$ або $+32767$. За допомогою 32 біт або 4 байт у пам'яті комп'ютера можна представити цілі числа від -2^{31} або -2147483648 до $+2^{31}-1$ або $+2147483647$.

Приклад 2.5.3. Записати прямий код чисел $+75$ і -75 в однобайтовому форматі.

Розв'язок. У прикладі 2.5.2 було отримане представлення числа 75 в однобайтовому форматі. Для запису прямого коду числа слід пам'ятати, що в старшому (крайньому лівому) розряді «кодується» знак числа, а для вистави абсолютної величини використовується сім розрядів, що залишилися. Таким чином, прямий код числа $+75$ в однобайтовому форматі має вигляд:

$$\begin{array}{cccccccc}
 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\
 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\
 \hline
 + & \text{двійкове представлення } | +75 |
 \end{array}$$

Прямий код числа -75 в однобайтовому форматі має вигляд:

$$\begin{array}{cccccccc}
 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\
 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\
 \hline
 - & \text{двійкове представлення } |-75|
 \end{array}$$

Для спрощення структури арифметико-логічного обладнання в багатьох комп'ютерах операція віднімання відсутня. Вона замінюється операцією додавання, при цьому негативне число повинне бути представлено у вигляді зворотного або додаткового коду. Розглянемо віднімання як додавання з протилежним числом $A - B = A + (-B)$. У комп'ютерах можуть здійснюватися два типи додавання: з використанням зворотного й додаткового кодів. *Зворотний код* – це прямий код, у якому усі цифри, крім знакового розряду, замінені на протилежні (оскільки йдеться про двійкові

цифри, то 0 замінюється на 1, а одиниця – на нуль). Зворотний код негативного числа в знаковому розряді містить одиницю.

Алгоритм додавання з використанням зворотного коду:

1. позитивний доданок записати в прямому коді;
2. негативний – у зворотному;
3. коди підсумувати, включаючи знакові розряди. Якщо в знаковому розряді виникає 1 перенесення, то його слід додати до молодшого розряду суми кодів;
4. відповідь розглядати як позитивне число, якщо в знаковому розряді 0;
5. якщо в знаковому розряді 1, то відповідь – зворотний код шуканої суми.

Приклад 2.5.4. Обчислити а) $25-5$; б) $-25+5$ с) $-25-5$, застосувавши алгоритм додавання з використанням зворотного коду.

Розв'язок. а) $+25-5$

1. Прямий код $+25$ має вигляд $0\ 0011001$

2. Запишемо зворотний код для -5 . Прямий код -5 має вигляд $1\ 0000101$, тоді зворотний код -5 буде отриманий інвертуванням усіх цифр прямого коду за винятком знакового розряду, тобто має вигляд $1\ 1111010$.

3. Виконаємо підсумовування кодів, включаючи знакові розряди:



4. У знаковому розряді отриманий нуль, тому результатом додавання є позитивне число $00010100_2 = 2^4 + 2^2 = +20_{10}$.

b) $-25+5$

1. Прямий код $+5$ має вигляд $0\ 0000101$

2. Запишемо зворотний код для -25 . Прямий код -25 має вигляд $\bar{1}\ 0011001$, тоді зворотний код -25 буде отриманий інвертуванням усіх цифр прямого коду за винятком знакового розряду, тобто має вигляд $\bar{1}\ 1100110$.

3. Виконаємо підсумовування кодів, включаючи знакові розряди:

-											
+	1	1	1	0	0	1	1	0			
	+										
	0	0	0	0	0	1	0	1			
	-										
	1	1	1	0	1	0	1	1			

зворотний код -25

прямий код $+5$

5. У знаковому розряді отримана одиниця, тому відповідь – зворотний код шуканої суми $\bar{1}1101011$, тоді прямий код результату має вигляд $\bar{1}0010100_2 = -(2^4 + 2^2) = -20_{10}$.

с) $-25-5$

Тому що обидва числа негативні, переходимо відразу до 2-го кроку алгоритму.

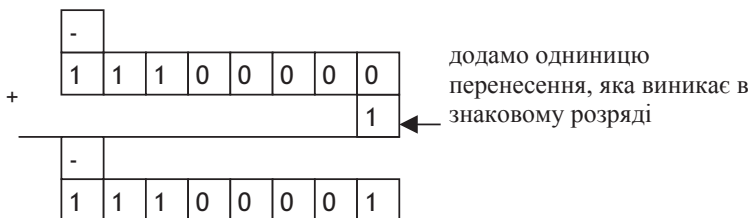
2. Зворотний код для -25 має вигляд $\bar{1}\ 1100110$, а для -5 – $\bar{1}\ 1111010$.

3. Виконаємо підсумовування кодів, включаючи знакові розряди:

+	1	1	1	0	0	1	1	0			
	-										
	1	1	1	1	1	0	1	0			
	-										
	1	1	1	0	0	0	0	0			

зворотний код -25

зворотний код -5



5. У знаковому розряді отримана одиниця, тому відповідь – зворотний код шуканої суми 11100001 , тоді прямий код результату має вигляд $10011110_2 = -(2^4 + 2^3 + 2^2 + 2^1) = -30_{10}$.

Додатковий код – це зворотний код, складений з 1.

Алгоритм додавання з використанням додаткового коду

1. позитивний доданок – у прямому коді;
2. негативний – у зворотньому;
3. виконується підсумок, включаючи знакові розряди. Якщо зі знакового розряду виникає 1 перенесення, її слід відкинути.
4. якщо в знаковому розряді суми 0, то отриманий прямий код шуканої суми (результат позитивний);
5. якщо в знаковому розряді результату 1, то отриманий додатковий код негативного результату. Для перетворення в прямий код треба значення всіх розрядів (крім знакового розряду) замінити протилежними значеннями (інвертувати), після чого до молодшого розряду додати одиницю.

Приклад 2.5.5. Обчислити а) $25-5$; б) $-25+5$ в) $-25-5$, застосувавши алгоритм додавання з використанням додаткового коду.

Розв'язок. а) $+25-5$

1. Прямий код $+25$ має вигляд $0\ 0011001$
2. Запишемо додатковий код для -5 . Для цього до зворотного коду -5 , який має вигляд $1\ 1111010$ додамо одиницю.



Додатковий код -5 має вигляд $\bar{1}1111011$.

3. Виконаємо підсумовування кодів, включаючи знакові розряди:



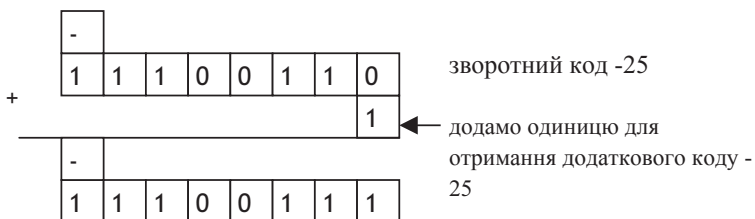
4. У знаковому розряді суми отриманий нуль, тому результатом додавання є

прямий код позитивного числа $00010100_2 = 2^4 + 2^2 = +20_{10}$.

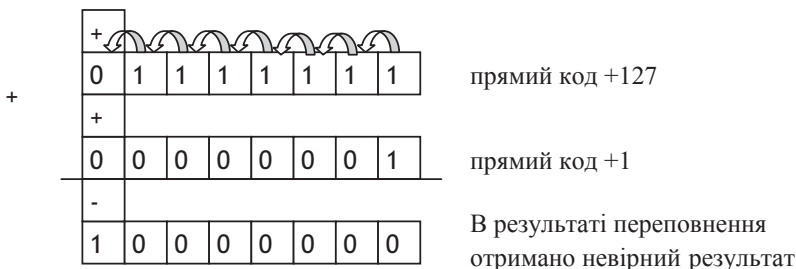
b) -25+5

1. Прямий код +5 має вигляд $0^+0000101$

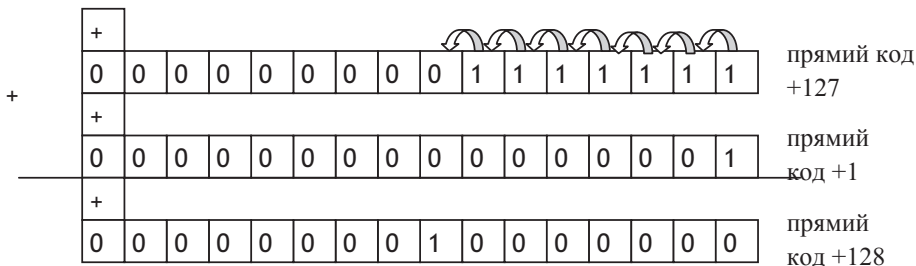
2. Запишемо додатковий код для -25. Для цього додамо до зворотного коду -25, який має вигляд $\bar{1}1100110$ одиницю.



Розглянемо приклад переповнення розрядної сітки однобайтового формату числа. За допомогою одного байта в пам'яті комп'ютера можна зберігати числа в діапазоні від -2^7 (або -128) до $+2^7-1$ (або $+127$). Це означає, що, наприклад, $+128$ або будь-яке ціле число не із цього діапазону представити в однобайтовому форматі не можна. Дійсно, при додаванні одиниці до $+127$ в однобайтовому форматі виникне переповнення розрядної сітки.



При використанні двобайтового формату для одержання суми $+127+1$ ситуація переповнення розрядної сітки не виникає.



При додаванні чисел зі знаком переповнення розрядної сітки відбувається або коли сума позитивних чисел, або коли сума абсолютних величин негативних чисел не менше 2^{n-1} .

На закінчення відзначимо переваги й недоліки використання розглянутих форм представлення (кодування) цілих чисел зі знаком.

1. Оскільки додатковий код можна отримати зі зворотного шляхом додавання одиниці до молодшого розряду, то комп'ютер затрачає більше часу на кодування негативного числа в додатковий код, ніж на його представлення у зворотному коді.

2. На додавання негативних чисел у додаткових кодах комп'ютер затрачає менше часу, ніж на додавання цих же чисел у зворотних кодах. Це пов'язане з тим, що при додаванні чисел у додаткових кодах не треба додавати одиницю перенесення, що виникла в знаковому розряді, до молодшого розряду суми.

Представлення дійсних чисел у комп'ютері

У комірці пам'яті дійсні числа представлені за певним шаблоном:

знак числа	мантиса	знак порядку	порядок
------------	---------	--------------	---------

Таке число називається *нормалізованим*, його *мантиса* є правильним дробом. Всяке число в нормалізованій формі має вигляд: $A_q = M_q \cdot q^p$ де q - основа, M_q - мантиса, p - порядок. Мантису і порядок *q-їчного* числа прийнято записувати в системі числення з основою q , а сама основа — в десятковій системі. Наприклад, десяткове число 475,68 в нормалізованому вигляді виглядає як $0,47568 \cdot 10^3$, а двійкове число 11,0101 в нормалізованому вигляді виглядає як $0,110101 \cdot 2^{10}$, причому порядок $p=10_2=2_{10}$.

При цьому сама мантиса при записі в пам'ять не має знаків «0» і «,», тобто набуває вигляду цілого числа. Знак числа також зберігається як двійкова цифра: для позитивного числа в полі знаку записується нуль, а для негативного – одиниця. Такий запис двійкового числа називається прямим кодом.

$$A = -0,101011_2$$

$$A_{np} = 1101011_2$$

$$B = 0,11101_2$$

$$B_{np} = +011101_2$$

Для виконання комп'ютером дій над числами проводяться наступні перетворення:

1. Десяткові числа переводяться в шістнадцятиричну систему числення згідно з правилами переведення.

2. З шістнадцятиричної системи числа перетворюються в двійкову (кожна цифра замінюється чотирма двійковими).
3. Двійкове число нормалізується, утворюючи мантису і порядок.
4. Якщо складаються два числа, то вирівнюються їх порядки, перетворюючи менший порядок до більшого (наприклад, скласти $0,5 \cdot 10^3$ і $0,2 \cdot 10^5$, можна тільки, якщо порядки рівні: $0,005 \cdot 10^5 + 0,2 \cdot 10^5$ і дістанемо можливість скласти мантиси, результат буде $0,205 \cdot 10^5$).
5. Залежно від знаків чисел мантиси складаються або з використанням алгоритму для зворотних кодів, або – для додаткових.
6. Результат складання переводиться в десяткову систему числення.

Приклад 2.5.6. Обчислити $29_{10} - 75_{10}$ з використанням зворотнього коду.

Рішення.

$$A = 29_{10} = 1D_{16} = +11101_2 = 0,11101 \cdot 10^{101}$$

$$B = -75_{10} = -4B_{16} = -1001011_2 = -0,1001011 \cdot 10^{111}$$

$$P_A \neq P_B$$

Треба вирівняти порядки:

$$A = 0,11101 \cdot 10^{101} = 0,0011101 \cdot 10^{111}$$

Складасмо мантиси як цілі числа

$$\begin{array}{r}
 A_{np} = \overset{+}{0} \mid 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 1 \\
 + \\
 B_{obp} = \overset{-}{1} \mid 0 \ 1 \ 1 \ 0 \ 1 \ 0 \ 0 \\
 \hline
 \phantom{A_{np}} \mid 1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \text{ обр. код отриц. числа} \\
 \phantom{A_{np}} \mid - \ 0 \ 1 \ 0 \ 1 \ 1 \ 1 \ 0 \text{ прямой код}
 \end{array}$$

$$\hat{A} = -0,0101110 \cdot 10^{111} = -101110 = -2\hat{A}_{16} = -46_{10}$$

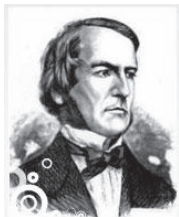
Слід зауважити, що дійсні числа в комп'ютерах різних типів записуються по-різному. Перерахуємо стандартні формати представлення дійсних чисел в комп'ютері:

1. одинарний формат – 32-розрядне нормалізоване число зі знаком;
2. подвійний формат – 64-розрядне нормалізоване число зі знаком;
3. розширений формат – 80-розрядне нормалізоване число зі знаком.

Питання і завдання для самостійної роботи

1. Перевести числа з восьмиричної системи числення а) 127; б) 307; в) 2365; д) 55701; е) 332201 і шістнадцятиричної системи числення ф) 6D11A; ж) 7A89; з) 203F; и) 9084; г) 543288 в двійкову систему числення.
2. Перевести числа а) 101100111010011; б) 1110001101001; в) 000110101101; д) 010111001110101001; е) 00101110100101; ф) 1000100110; ж) 0100011101001; з) 100001111100000101; и) 0111000111010110; г) 101010001101 з двійкової системи числення у восьмерічну і шістнадцятиричну системи числення.
3. Записати прямий, зворотний і додатковий коди чисел а) -126; б) -115; в) 47; д) 107; е) -53; ф) 79; ж) -63; з) 105; и) -29; г) -17 у однобайтовому форматі.
4. Знайдіть десяткові представлення чисел, записаних в додатковому коді: а) 1100110001101011; б) 01011110; в) 11001010; д) 1001110101011111; е) 0110001001011101; ф) 1010010010010000; ж) 0011000110110001; з) 10011110; и) 00111011; г) 1111010001001000.
5. Знайдіть десяткові представлення чисел, записаних в зворотному коді: а) 0111111011000010; б) 01011110; в) 111111100101010; д) 10000000; е) 0111110100111101; ф) 0010101111001001; ж) 10100011; з) 1111000001100011; и) 0000000111100110; г) 1001001000110001.
6. Виконайте складання чисел в зворотних кодах з використанням однобайтового формату. Вкажіть, в яких випадках виникає переповнювання розрядної сітки: а) -45-13; б) -78+15; в) -98-33; д) 125-67; е) -59+23; ф) -26-105; ж) -38+119; з) 61-112; и) -73-28.
7. Виконайте складання чисел в додаткових кодах з використанням однобайтового формату. Вкажіть, в яких випадках виникає переповнювання розрядної сітки: а) 38+57; б) 78-13; в) -112-93; д) 256-14; е) -347+17; ф) -229-46; ж) 244-26; з) 452+74; и) -65-34; г) -228+37.

2.6. Зв'язок між логічними та арифметичними операціями



Джордж Буль (1815 — 1864) – видатний англійський математик і логік, який розробив алгебру логіки – основу функціонування комп'ютерів. Алгебра логіки – це розділ математики, що вивчає логічні висловлювання і операції над ними.

Основними елементами арифметико-логічного пристрою, що виконує дії з двійковими числами, є логічні елементи, реалізовані у вигляді електронних пристроїв. Технічні реалізації цих елементів постійно удосконалюються, процесори кожні півтора роки подвоюють свою швидкість, але принципова основа реалізації арифметичних дій за допомогою логічних елементів (званих в літературі також логічними вентилями) залишається. Логічними елементами комп'ютерів є електронні схеми {I, АБО, НЕ} {I, НЕ} {АБО, НЕ}. За допомогою цих схем можна реалізувати будь-яку логічну функцію, що описує роботу пристроїв комп'ютера. Нижче розглянемо схему однорозрядного двійкового суматора, що складає два однорозрядні двійкові числа.

Логічні елементи виконують три операції, які мають наступні назви: логічне множення (операція «I» - кон'юнкція, позначається також символом \wedge або $\&$), логічне складання (операція «АБО» - диз'юнкція, позначається також символом \vee) і логічне заперечення (операція «НЕ» - інверсія, позначається також рискою над ім'ям змінної). Дані операції виконуються над висловлюваннями – оповідними реченнями, про які можна сказати, достеменно вони або помилкові. При цьому істину можна позначити 1, а неправду – 0.

Наприклад, одне висловлювання: «Йде сніг», а друге – «Сьогодні літо». Якщо ми знаходимося в північній півкулі Землі, то зв'язавши ці два вислови операцією «I» отримаємо явну неправду, а операцією «АБО» – правду.

Розглянемо основні логічні операції над висловлюваннями.

1. Операція «**константа 0**» відповідає тотожно хибному висловлюванню.

2. Операція «**константа 1**» відповідає тотожно правдивому висловлюванню.

3. Логічною операцією «**НЕ**» називається таке висловлювання \bar{A} , яке є правдивим тоді і тільки тоді, коли висловлювання A є хибним.

4. Логічною операцією «**I**» називається таке висловлювання $A \wedge B$, яке є правдивим тоді і тільки тоді, коли обидва висловлювання A і B одночасно правдиві і хибним, коли хоч би одне з висловлювань є хибним.

5. Логічною операцією «**АБО**» називається висловлювання $A \vee B$, яке є хибним тоді і тільки тоді, коли обидва висловлювання A і B одночасно хибні і правдивим, коли хоч би одне з висловлювань є правдивим.

6. Логічною операцією «**ЕСЛИ-ТО**» або імплікацією називається висловлювання $A \rightarrow B$, яке є хибним тоді і тільки тоді, коли A – правдиве висловлювання, а B – хибне висловлювання.

7. Логічною операцією «**Тоді і тільки тоді**» або еквівалентність називається висловлювання $A \sim B$, яке є правдивим тоді і тільки тоді, коли обидва висловлювання A і B є одночасно правдивими або одночасно хибними.

8. Логічною операцією «**ЩО ВИКЛЮЧАЄ АБО**» або сума по модулю два називається висловлювання $A \oplus B$, яке є правдивим тоді і тільки тоді, коли одне висловлювання є правдивим, а інше – хибним.

9. Логічною операцією «**ШТРИХ ШЕФЕРА**» називається висловлювання $A|B$, яке є хибним тоді і тільки тоді, коли висловлювання A і B є одночасно правдивими.

10. Логічною операцією «**СТРІЛКА ПІРСУ**» називається висловлювання $A \downarrow B$, яке є правдивим тоді і тільки тоді, коли висловлювання A і B є одночасно хибними.

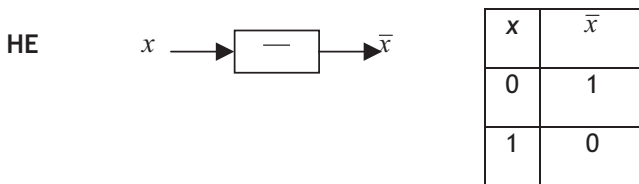
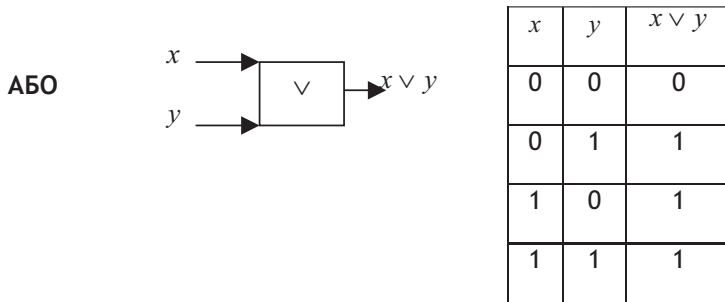
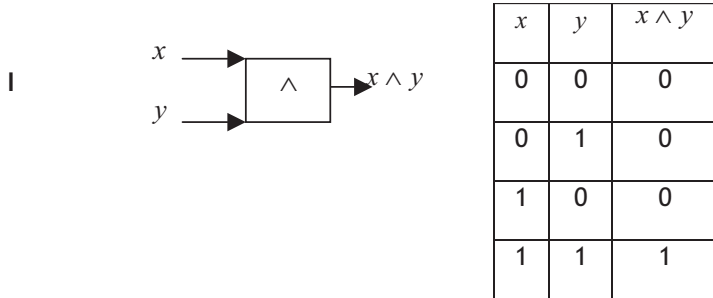
Тотожно перетворити будь-яке логічне вираження до вигляду з логічними елементами I, АБО, НЕ можна за допомогою основних законів алгебри логіки. Хай A і B – висловлювання, тоді справедливі наступні закони алгебри логіки, представлені в таблиці 2.6.1.

Таблиця 2.6.1. Основні закони алгебри логіки

Назва закону	Закон для АБО	Закон для І
Комутативний	$A \vee B = B \vee A$	$A \wedge B = B \wedge A$
Асоціативний	$(A \vee B) \vee C = A \vee (B \vee C).$	$(A \wedge B) \wedge C = A \wedge (B \wedge C).$
Дистрибутивний	$A \wedge (B \vee C) =$ $= (A \wedge B) \vee (A \wedge C).$	$A \vee (B \wedge C) =$ $= (A \vee B) \wedge (A \vee C).$
Ідемпотентність	$A \vee A = A.$	$A \wedge A = A.$
Поглинання	$A \vee (A \wedge B) = A.$	$A \wedge (A \vee B) = A.$
Склеювання	$(A \wedge B) \vee (\overline{A \wedge B}) = B.$	$(A \vee B) \wedge (\overline{A \vee B}) = B.$
Правила де Моргана	$\overline{A \vee B} = \overline{A} \wedge \overline{B}.$	$\overline{A \wedge B} = \overline{A} \vee \overline{B}.$
Операція вислову з його запереченням	$A \vee \overline{A} = 1.$	$A \wedge \overline{A} = 0.$
Властивість одиниці	$A \vee 1 = 1.$	$A \wedge 1 = A.$
Властивість нуля	$A \vee 0 = A.$	$A \wedge 0 = 0.$
Подвійне заперечення	$\overline{\overline{A}} = A.$	

Виконання логічних операцій над невеликою кількістю висловлювань зручно представляти у вигляді таблиць істинності, по яких можна судити про результат.

Таблиці істинності і відповідні їм функціональні елементи мають вигляд:



При виконанні складання двох однорозрядних двійкових чисел можливі наступні комбінації:

$$\begin{array}{r}
 0 \\
 + 0 \\
 \hline
 00
 \end{array}
 \quad
 \begin{array}{r}
 0 \\
 + 1 \\
 \hline
 01
 \end{array}
 \quad
 \begin{array}{r}
 1 \\
 + 0 \\
 \hline
 01
 \end{array}
 \quad
 \begin{array}{r}
 1 \\
 + 1 \\
 \hline
 10
 \end{array}$$

Побудуємо схему двійкового суматора. Двійковий суматор – це електронна логічна схема, що виконує підсумовування двійкових чисел. Позначимо значення першого розряду результату через S_2 , а другого – S_1 .

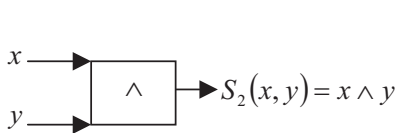
x	y	S_2	S_1
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Виведемо логічну формулу для кожного розряду. Для S_2 це досить просто – він дорівнює 1, тільки коли обидва доданки дорівнюють 1, тобто досить виконати для них операцію «І». Для другого – складніше, послідовне виконання над операндами отриманої формули представлене в наступній таблиці істинності, а зображення її у вигляді схеми логічних елементів, відповідних операціям наведено нижче.

$$S_2(x, y) = x \wedge y$$

$$S_1(x, y) = (\bar{x} \wedge y) \vee (x \wedge \bar{y})$$

Схема для першої цифри S_2 має вигляд:

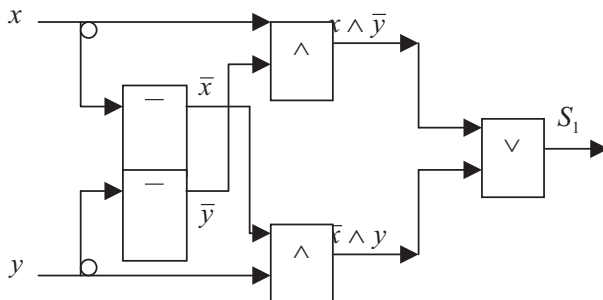


x	y	S_2
0	0	0
0	1	0
1	0	0
1	1	1

Таблиця істинності для другої цифри результату $S_1 = (x \wedge \bar{y}) \vee (\bar{x} \wedge y)$

x	y	\bar{x}	\bar{y}	$\bar{x} \wedge y$	$x \wedge \bar{y}$	$S_1(x, y)$
0	0	1	1	0	0	0
0	1	1	0	1	0	1
1	0	0	1	0	1	1
1	1	0	0	0	0	0

Схема для другої цифри результату складання має вигляд:



Загальна об'єднана схема однорозрядного двійкового суматора запропонована на рисунку 2.6.1.

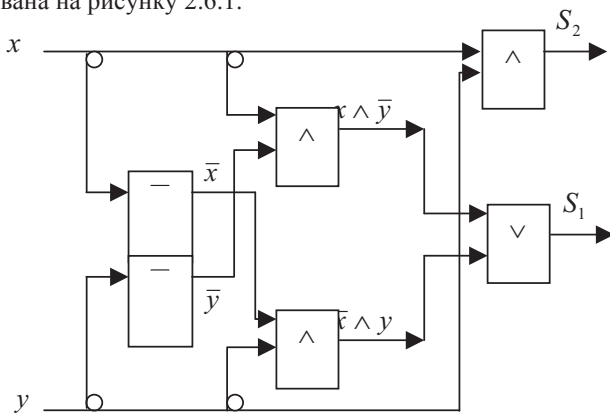


Рис. 2.6.1. Схема однорозрядного двійкового суматора

Питання і завдання для самостійної роботи

- Які з наступних стверджувальних речень є логічними висловлюваннями?
 - « $2 \times 2 = 5$ »;
 - «Сімферополь – столиця Криму»;
 - «якщо відрізок AB паралельний відріжку CD , то відрізок CD паралельний відріжку AB »;
 - Економіка повинна бути економною;
 - Систему рівнянь можна розв'язати за методом Крамера або за методом Гаусса;
 - Кожен студент економічного факультету здатен розв'язати задачу лінійного програмування;
 - Інформатика для економістів – це обов'язковий курс навчання в університеті;
 - Фахівець з економіки повинен володіти знаннями про сучасні інформаційні системи і технології.
- Побудувати таблиці істинності для логічних формул:
 - $(A \oplus B) \downarrow (\bar{B} \rightarrow A)$;
 - $\overline{A \sim B} (\bar{A} \vee B)$;

- c) $(A \vee \bar{C}) \wedge (\bar{B} \vee C) \rightarrow \bar{A}$;
 d) $(\bar{A} \downarrow \bar{B}) \oplus (\bar{C} \sim (A \vee \bar{B}))$;
 e) $(\bar{A} \wedge (\bar{B} | C)) \rightarrow (A \vee (B \oplus C))$;
 f) $C \rightarrow (\bar{A} \rightarrow \bar{B}) \downarrow (B \sim C)$;
 g) $((\bar{A} \oplus \bar{B}) \rightarrow C) | (A \rightarrow (\bar{B} \oplus C))$;
 h) $(B \sim (\bar{A} \vee C) \rightarrow \bar{B}) \oplus (C \oplus \bar{A})$.

3. Опишіть побудову схеми однорозрядного двійкового суматора
 4. Перерахуйте основні закони алгебри логіки і вкажіть, у чому їхня сутність

Список літератури

1. Борисенко О.А. Лекції з дискретної математики (множини і логіка): Навчальний посібник. 3-є вид., випр. і доп. – Суми: ВТД «Університетська книга», 2002. – 180 с.
2. Шауцукова Л.З. Інформатика, 10-11: Підручник. – М.: Вид-во «Освіта», 2004. – 420 с.
3. Бардачов Ю.М., Соколова Н.А., Ходакова В.Є. Дискретна математика.- К.: Вища шк., 2002. – 287 с.
4. Новиков Ф.А. Дискретная математика для программистов. – СПб: Питер, 2000. – 304 с.
5. Матросов В.Л., Стеценко В.А. Лекции по дискретной математике. Учебное пособие для магистратов математических факультетов педагогических университетов. М.: МПГУ, 1997, 220 с.
6. Акимов О.Е. Дискретная математика: логика, группы, графы. – М.: Лаборатория Базовых Знаний, 2001. – 352 с.

Розділ III. ОРГАНІЗАЦІЯ І ПРИНЦИПИ ФУНКЦІОНУВАННЯ КОМП'ЮТЕРІВ

У цьому розділі наведено основні відомості про будову комп'ютерів, описано класичну архітектуру Джона фон Неймана та інші види архітектур, перелічено основні функціональні блоки комп'ютера і їхнє призначення.

3.1. Архітектура комп'ютера

Дж. фон Нейман в 1946 році розробив схему функціональної будови комп'ютера. Більшість сучасних процесорів для персональних комп'ютерів засновані на тій або іншій версії циклічного процесу послідовної обробки інформації, винайденого Джоном фон Нейманом.

Найважливіші етапи процесу функціонування комп'ютера приведені нижче. У різній архітектурі і для різних команд можуть бути потрібні додаткові етапи. Наприклад, для арифметических команд можуть бути потрібні додаткові звернення до пам'яті, під час яких проводиться прочитування операндів і запис результатів. Відмітною особливістю архітектури фон Неймана є те, що інструкції і дані зберігаються в одній і тій же пам'яті.

1. Прочитування команди з оперативної пам'яті за адресою з «лічильника команд».

2. «Дешифровка» команди і прочитування з пам'яті даних, які є операндами.

3. Виконання операції з даними.

4. Збереження в пам'яті даних, отриманих в результаті виконання команди.

5. «Лічильник команд» збільшується на довжину команди для переходу до наступної команди. У випадку якщо виконувана команда була командою переходу, то «лічильник команд» змінюється на задане нею значення. Відбувається повернення до першого етапу.

Даний цикл виконується незмінно, і саме він називається *процесом* (звідки і сталася назва пристрою - *процесор*).

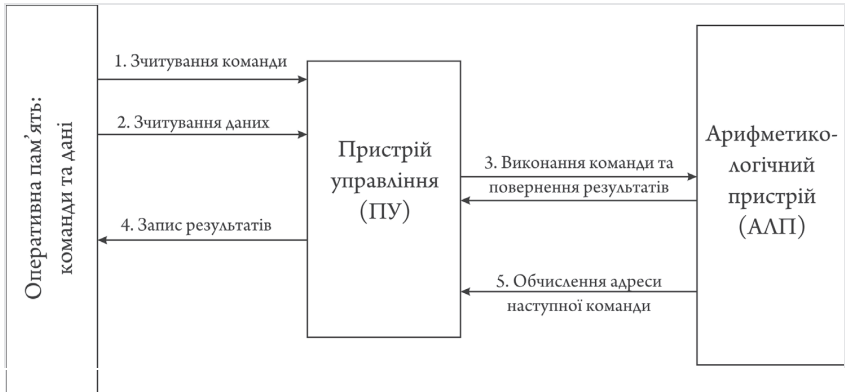


Рис.3.1.1. Функціональна схема комп'ютера

Етапи циклу виконання:

Під час процесу процесор виконує послідовність команд, що містяться в пам'яті. Така послідовність команд називається *програмою* і представляє алгоритм корисної роботи процесора. Черговість прочитування команд змінюється у випадку, якщо процесор отримує команду зупинки (завершення програми) або перемикається в режим обробки апаратного переривання (реакцію на запити інших пристроїв комп'ютера).

Команди центрального процесора є самим нижнім рівнем управління комп'ютером, тому виконання кожної команди неминуче і безумовно. Не проводиться ніякої перевірки на допустимість виконуваних дій, зокрема, не перевіряється можлива втрата цінних даних. Щоб комп'ютер виконував тільки допустимі дії, команди мають бути відповідним чином організовані у вигляді необхідної програми.

Швидкість переходу від одного етапу циклу до іншого визначається тактовим генератором. Тактовий генератор виробляє імпульси, які служать ритмом для центрального процесора. Частота тактових імпульсів називається тактовою частотою.

Конвеєрна архітектура. Конвеєрна архітектура (*pipelining*) була застосована для побудови центрального процесора з метою підвищення швидкодії. Зазвичай для виконання кожної команди потрібно здійснити деяку кількість однотипних операцій, наприклад: вибірка команди з оперативної пам'яті, дешифрування команди, пошук операнда в оперативній пам'яті, вибірка операнда з оперативної пам'яті, виконання команди, запис результату в оперативну пам'ять. Кожну з цих операцій зіставляють одному ступеню

конвеєра. Наприклад, конвеєр мікропроцесора з архітектурою MIPS-I містить чотири стадії:

- отримання і декодування інструкції (Fetch);
- адресація і вибірка операнда з оперативної пам'яті (Memory access);
- виконання арифметичних операцій (Arithmetic Operation);
- збереження результату операції (Store).

Після звільнення чергового ступеня конвеєра вона відразу приступає до роботи над наступною командою. Таким чином стадії конвеєра можуть одночасно виконувати декілька команд на різних етапах. Але в тому випадку, якщо одна з команд в конвеєрі вимагає для свого виконання результатів роботи іншої команди, яка теж ще не завершена в конвеєрі, або якщо зустрічається команда переходу (цю проблему вдається згладити, використовуючи передбачення переходів), то тоді деякі із ступенів конвеєра можуть простоювати.

Деякі сучасні процесори мають більше 30 ступенів в конвеєрі, що збільшує продуктивність процесора, проте призводить до великого часу простою (наприклад, в разі помилки в передбаченні умовного переходу.)

У сучасних комп'ютерах рідко згадують окремо арифметико-логічний пристрій і пристрій управління. Вони об'єднані словом «процесор».

Центральний процесор (ЦП; CPU — англ. *central processing unit*, дослівно — **центральный обчислювальний пристрій)** — **процесор машинних інструкцій**, частина апаратного забезпечення комп'ютера, що відповідає за виконання арифметичних операцій, які задаються програмами операційної системи, і координують роботу всіх пристроїв комп'ютера.

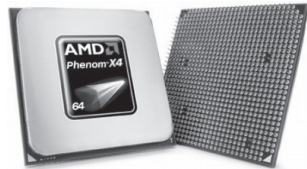


Рис. 3.1.2. Центральний процесор

Сучасні ЦП, виконувані у вигляді окремих мікросхем (чіпів), що реалізують всі особливості, властиві даного роду пристроям, називають мікропроцесорами. З середини 1980-х останні практично витіснили інші види ЦП, унаслідок чого термін став все частіше і частіше сприйматися як звичайний синонім слова «мікропроцесор». Проте, це не так: центральні процесорні пристрої деяких суперкомп'ютерів навіть сьогодні є складними комплексами великих (ВІС) і надвеликих (НВІС) інтегральних схем.

У ранніх комп'ютерах різних виробників використовувалися елементи пам'яті різного розміру. Проте, в середині 60-х років, коли фірма ІВМ випустила сімейство обчислювальних машин третього покоління, склався стандарт одиниці виміру пам'яті — байт.

Байт — це система з 8 двійкових елементів (бітів), кожен з яких може знаходитися в двох станах, що позначаються 0 і 1.

Кожен байт має «адресу» — порядковий номер комірки. Байт може зберігати числа і символи. Для запам'ятовування символів раніше повсюдно використовувався код ASCII (American Standard Code for Information Interchange), в таблиці кодування якого кожному символу відповідає шифр з 8 бітів (нулів і одиниць), що дозволяє кодувати 256 символів. На сьогодні, з урахуванням широкого розповсюдження комп'ютерної техніки все більшого поширення набуває система кодування символів **Unicode** (скорочення Universal Code – англ. універсальний код), в якій кожному символу відповідає код з 16 бітів (2 байти), що дозволяє включити в таблицю 65 536 символів, що достатньо для представлення символів майже всіх мов.

Для запам'ятовування чисел використовуються спеціальні шаблони, відповідні типові числа. Для цілого числа використовується, як правило, два байти, для дійсного — чотири або шість, залежно від системи програмування. Команди також бувають різної довжини. Команди рідко містять дані для обробки у вигляді конкретних чисел, в них записані адреси даних. Залежно від того, скільки адрес дозволяла записати команда, комп'ютери називалися одно-, двух- і триадресними машинами.

Із збільшенням об'ємів комп'ютерної пам'яті почали застосовувати кратні одиниці виміру об'єму пам'яті:

- 1 кілобайт (Кб) – 1024 байти
- 1 мегабайт (Мб) – 1024 кілобайти
- 1 гігабайт (Гб) – 1024 мегабайти
- 1 терабайт (Тб) – 1024 гігабайти.

Програмне забезпечення персональних комп'ютерів дозволяє користувачеві не думати про те, який формат команди машинної мови застосовується, де саме в пам'яті знаходиться його програма і дані. Такі відомості необхідні розробникам комп'ютерів і операційних систем.

3.2. Функціональні блоки комп'ютера

Функціональні блоки комп'ютера можуть бути реалізовані у вигляді різних фізичних пристроїв. У персональному комп'ютері процесор — це одна мікросхема, тому він називається *мікропроцесором*.

Оперативна пам'ять

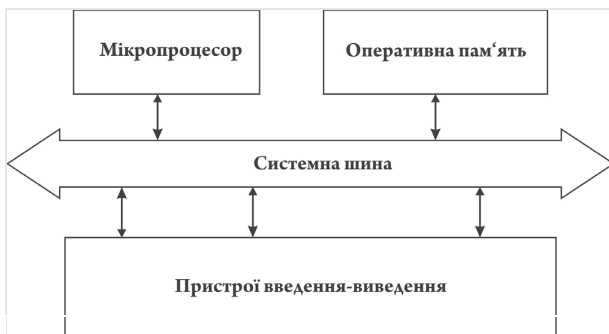


Рис 3.2.1. Проста схема взаємодії оперативної пам'яті з процесором

Оперативна пам'ять (також **оперативний запам'ятовувальний пристрій, ОЗП**) — це частина запам'ятовувальної системи ЕОМ, у яку процесор може звернутися за одну операцію (jump, move і т.і.). Призначена для тимчасового зберігання даних і команд, що необхідні процесору для виконання їм операцій. Оперативна пам'ять передає дані безпосередньо процесору. Кожна комірка оперативної пам'яті має свою індивідуальну адресу.



Рис. 3.2.2. Плата оперативної пам'яті



Рис. 3.2.3. Накопичувач на жорстких магнітних дисках

У сучасних обчислювальних пристроях оперативна пам'ять виконана за технологією *динамічної пам'яті з довільним доступом* (англ. *dynamic random access memory, DRAM*). Поняття пам'яті з довільним доступом передбачає, що поточне звернення до пам'яті не враховує порядок попередніх операцій і розташування даних в ній. ОЗП може виготовлятися як окремий блок, або входити в конструкцію однокристалльної ЕОМ або мікроконтролера.

Крім «внутрішньої» в комп'ютері є також «зовнішня» пам'ять, яка інакше має назву «енергонезалежної пам'яті». Це означає, що такі пристрої зберігають дані незалежно від того, увімкнено чи или ні комп'ютер. До неї належать накопичувачі на жорстких магнітних дисках, оптичні (лазерні)

диски і накопичувачі на основі флеш-пам'яті. Вони дозволяють здійснювати як ввід, так і вивід даних, тобто виконують дві функції. Накопичувач на жорстких магнітних дисках, НЖМД (HDD – Hard Disk Drive), іноді має назву «вінчестера» і являє собою конструкцію, яка складається з алюмінієвих пластин, покритих магнітним шаром, магнітних голівок і приводячих моторів в герметичному корпусі. Він монтується в корпусі комп'ютера і може бути заміненим у випадку несправності або за бажанням користувача.

НЖМД, хоча і належить до зовнішньої пам'яті, фактично є вбудованим пристроєм комп'ютера (рис. 3.2.3).

Важливу роль в перенесенні значних об'ємів інформації між комп'ютерами грають *змінні* накопичувачі даних.

Флеш-пам'ять (англ. *Flash-Memory*)

— різновид твердотільної напівпровідникової енергонезалежної перезаписуваної пам'яті. Дані з будь-якої комірки флеш-накопичувача можуть бути прочитані скільки завгодно разів, але запис в окрему комірку може здійснюватися лише обмежене (хоча і досить велике) число разів. Не дивлячись на те, що таке обмеження є, кількість циклів перезапису, як правило, перевищує таку кількість для магнітної дискети або перезаписуваного лазерного диску CD-RW. Флеш-пам'ять не містить рухливих частин, так що, на відміну від жорстких дисків, більш надійна і компактна.

Завдяки своїй компактності, дешевизні і низькій потребі в електроенергії флеш-пам'ять також широко використовується в портативних пристроях, що працюють на батареях і акумуляторах, — цифрових фото- і відеокамерах, цифрових диктофонах, MP3-плеєрах, кишенькових ПК (КПК) мобільних телефонах, а також смартфонах. Крім того, вона використовується для зберігання вбудованого програмного забезпечення в різних периферійних пристроях (маршрутизаторах, МІНІ-АТС, комунікаторах, принтерах, сканерах).



Рис. 3.2.4. USB флеш-накопичувач



Рис. 3.2.5. Пристрій для читання лазерних дисків

Персональний комп'ютер може також працювати з оптичними (лазерними) дисками, CD-ROM (скорочення від англ. *Compact Disc read-only memory*) — це компактний оптичний диск, що містить дані доступні для комп'ютера. Запис і зчитування даних з таких дисків здійснюється з використанням світлового

лазеру. Споконвічно диск розроблювався для зберігання і відтворення музичних записів, згодом його було допрацьовано для зберігання цифрових даних. Лазерні диски — зручний засіб для поширення програмного забезпечення, комп'ютерних ігор, мультимедійних програм.

Існує декілька стандартів оптичних дисків, що розрізняються довжиною хвилі (кольором) лазерного променя, використовованого для читання/запису даних, а також можливістю/відсутністю перезапису даних на диску. Так існує стандарт CD, що дозволяє зберігати на одному диску до 800 Мбайт інформації, DVD – до 8,5 Гбайт і BD – до 54 Гбайт на одному диску. У кожному із стандартів існують диски, призначені для однократного запису, диски з можливістю перезапису даних і диски «тільки для читання», записувані при їх виготовленні.

Перераховані компоненти комп'ютера знаходяться в основному його блоці, званому «системний блок». Головним єдиним елементом для всіх вищеперелічених пристроїв є так звана «**материнська плата**».

Материнська плата — це складна багатшарова друкована плата, на якій устанавлюються основні компоненти персонального комп'ютера (центральный процесор, контролер ОЗП і, власне, ОЗП, завантажувальний ПЗП, контролери базових інтерфейсів вводу-вивода). Як правило, материнська плата містить гнізда (слоти) для підключення додаткових контролерів з використанням шин USB, PCI і PCI-Express, а також дискових накопичувачів.

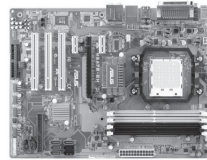


Рис. 3.2.6. Материнська плата

Назва походить від англ. motherboard, інколи використовується скорочення MB або слово mainboard – головна плата.

Основні компоненти, встановлені на материнській платі:

- центральний процесор (ЦПУ)
- набір системної логіки (англ. chipset), набір мікросхем, що забезпечують підключення процесора до оперативної пам'яті і контролерів периферійних пристроїв. Як правило, сучасні набори системної логіки будуються на базі двох мікросхем, званих «північний міст» і «південний міст» відповідно їх розташуванню на материнській платі.

На передній панелі системного блоку розташовані вмикач, приймальні панелі оптичних дисків, пристрої для читання флеш-карт, USB-гнізда і індикатори. На задній панелі є гнізда, що дозволяють підключати різні пристрої — дисплей, клавіатуру, принтер, мишу, сканер, а також USB-гнізда і гнізда для підключення до комп'ютерної мережі.

ПРИСТРОЇ ВВЕДЕННЯ-ВИВОДУ ІНФОРМАЦІЇ



Рис. 3.2.7. Рідкокристалічний монітор

Монітор або дисплей — пристрій відображення даних. Більшість дисплеїв можуть подавати дані в різних формах — тексти, графічні зображення. Це залежить від можливостей комп'ютера, наявності відповідної апаратної частини — відеопідсистеми. Основний елемент монітора — дисплей (екран), на якому за допомогою тієї чи іншої технології

формується зображення (алфавітно-цифрове або графічне).

Існує декілька видів моніторів, що розрізняються своєю будовою, пристроєм формування зображення:

- ЕПТ-монітор — на основі електронно-променевої трубки (англ. CRT cathode ray tube) ;
- РК — рідкокристалічні монітори (англ. LCD — liquid crystal display);
- Плазмовий — на основі плазменної панелі;
- Проекційний — відеопроєктор і екран, розміщені окремо або об'єднані в одному корпусі (як варіант через зеркало або систему дзеркал);
- OLED-монітор - монітор, заснований на технології OLED - Organic Light-Emitting Diode або органічний світловипромінюючий діод.

Найбільш поширеним видом моніторів на сьогодні є РК-монітори завдяки співвідношенню компактності, якості зображення, безпеки і ціни. Проте найбільшого поширення у міру здешевління технології виготовлення набувають OLED-монітори.

Основні характеристики монітора:

- розмір діагоналі і співвідношення сторін
- роздільна здатність – кількість відображуваних крапок, званих пікселі, і кількість кольорів, що передаються окремим пікселем
- швидкість оновлення зображення, тобто швидкість перемикання окремого пікселя з одного кольору в інший.

Пристрої введення інформації. Найбільш поширеними є такі пристрої як клавіатура і миша.



Рис. 3.2.8. Маніпулятор «миша»

Маніпулятор «миша» — одне з вказівних пристроїв введення (англ. pointing device), що забезпечують інтерфейс користувача з комп'ютером.

При переміщенні миші в робочій площині (звичайно на ділянці поверхні столу), програма, що працює на комп'ютері, утворює на екрані дію,



Рис. 3.2.9. Клавіатура

що відповідає напрямку і відстані цього переміщення. В універсальних інтерфейсах (наприклад, у віконних) за допомогою миші користувач керує спеціальним курсором — покажчиком-маніпулятором елементами інтерфейсу. Іноді використовується ввід команд за допомогою миші без участі видимих елементів інтерфейсу програми: за допомогою аналізу рухів миші. На додаток

до детектору переміщення миша має від однієї до трьох (або більше) кнопок, а також додаткові елементи управління (колеса прокручування, потенціометри, джойстики, трекболи, клавіші і т.д.), дія котрих звичайно пов'язується з поточним розміщенням курсору (або складових специфічного інтерфейсу).

Серед інших видів пристроїв введення інформації в комп'ютер все більшого поширення набуває технологія сенсорного екрану.

Сенсорний екран – пристрій введення-виводу інформації, що є екраном, здатним відображувати текстово-графічну інформацію, реагуючі на дотик до нього.

Використання такого екрану дозволяє управляти роботою комп'ютера без використання клавіатури (клавіші для набору змальовуються на екрані), миші (курсором можна управляти безпосередньо натисненнями на потрібні точки екрану) і будь-яких інших пристроїв. Таким чином сенсорний екран є одночасно і пристроєм виводу і пристроєм введення інформації.

ПЕРИФЕРІЙНІ ПРИСТРОЇ

До периферійних пристроїв прийнято відносити такі пристрої, які не є необхідними для функціонування комп'ютера, але дозволяють виконувати ті або інші функції, необхідні користувачеві:

- принтер – пристрій виведення зображень і тексту на папір або інші носії (тканина, плівка);
- сканер – пристрій введення зображень в комп'ютер з паперових і інших носіїв;
- акустична система – пристрій (набір пристроїв) для відтворення звуку;
- мікрофон – пристрій введення звукової інформації в комп'ютер;
- модем – сімейство пристроїв, що забезпечують зв'язок комп'ютера з іншими комп'ютерами і комп'ютерними мережами через середовище інших засобів зв'язку.

Комп'ютерний принтер (англ. printer — друкар) — пристрій друку цифрової інформації на твердий носій, зазвичай на папір.

В залежності від технології нанесення зображення на папір розрізняють принтери *матричні, струменеві, лазерні і сублімаційні (фотопринтери)*, а за кольором друку — *багатокольорові і монохромні*.

Матричні принтери, не дивлячись на те, що багато хто вважає їх застарілими, все ще активно використовуються для друку, (в основному з використанням безперервної подачі паперу, в рулонах) в лабораторіях, банках, бухгалтеріях, в бібліотеках для друку на картках, для друку на багатошарових бланках (наприклад, на авіаквитках), а також в тих випадках, коли необхідно отримати другий екземпляр документа через копірку (обидві копії підписуються через копірку одним підписом для запобігання внесення несанкціонованих змін у фінансовий документ).

Принтери безконтактного друку — це струменеві і електрографічні (лазерні). Струменевий принтер будує зображення з крапельок кольорового чорнила, що розпилюється декількома друкарськими голівками з чорнилом різних кольорів. Послідовність роботи кожної чорнильниці регулюється мікропроцесором принтера. При нанесенні близько 100 чорнильних крапок на квадратний сантиметр паперу виходить високоякісний кольоровий друк.

Також високоякісний, але чорно-білий друк, дають електрографічні принтери. Під дією променя лазера намагнічуються ділянки металевого барабана, до яких прилипає дрібний порошок. При цьому формується дзеркальне зображення всієї сторінки. Папір, що має протилежний заряд, проходить через пристрій і притягує порошок, який тут же закріплюється під дією тепла і тиску. Перед друком наступної сторінки барабан нейтралізується і очищається від порошку.

Сканер (*англ. scanner*) — пристрій, який, аналізуючи який-небудь об'єкт (звичайно зображення або текст), створює цифрову копію зображення об'єкту. Процес перенесення зображення в цифрову форму, отримання цифрової копії, називається скануванням.

Розглянемо принцип дії планшетних сканерів, як найбільш поширених моделей. Сканований об'єкт кладеться на скло планшета сканованою поверхнею вниз. Під склом розташовується рухлива лампа, рух якої регулюється шаговим двигуном. Світ, відображений від об'єкту, через систему дзеркал потрапляє на чутливу матрицю, далі на алфавітно-цифровий процесор і передається в комп'ютер. За кожен крок двигуна сканується смужка об'єкту, далі всі смужки об'єднуються програмним забезпеченням у спільне зображення.



Рис. 3.2.10. Струменевий принтер

Останнім часом широкого поширення набули так звані багатофункціональні пристрої (БФП), в яких в одному корпусі об'єднані принтер, сканер, копір і інколи факс. Таке об'єднання раціональне, технічне і зручне в роботі.

Модем (аббревіатура, складена із слів **модулятор-демодулятор**) пристрій, що застосовується в системах зв'язку і виконує функцію модуляції і демодуляції, тобто перетворення цифрових даних у форму сигналів, що передаються по каналах зв'язку. Окремим випадком модему є широко вживаний периферійний пристрій для комп'ютера, що дозволяє йому зв'язуватися з іншим комп'ютером, обладнаним модемом, через телефонну мережу (телефонний модем), мережі стільникового мобільного зв'язку (GSM-модем) або мережу кабельного телебачення (кабельний модем).

Найбільш поширена функція модему – підключення до глобальної комп'ютерної мережі Інтернет.

Питання і завдання для самостійної роботи

1. Хто і коли сформулював ідею програмно керованої обчислювальної машини?
2. Що розуміється під архітектурою комп'ютера?
3. Перерахуйте основні функціональні блоки комп'ютера і вкажіть їх призначення.
4. Поясніть, як ви розумієте сформульований Дж. фон Нейманом принцип єдиної лінійної пам'яті.
5. Чим відрізняється мікросхема від мікропроцесора?
6. Скільки байт містить кілобайт, мегабайт, гігабайт?
7. Який компонент комп'ютера забезпечує взаємодію всіх інших його компонентів?
8. Що таке накопичувач на жорстких магнітних дисках і для чого він призначений?
9. Які види зовнішніх носіїв даних використовує персональний комп'ютер?
10. Перерахуйте пристрої введення інформації в персональний комп'ютер.
11. Перерахуйте пристрої персонального комп'ютера, використовувані для виведення інформації.
12. Який пристрій визначає швидкість роботи комп'ютера?
13. Перерахуйте відомі Вам види моніторів.
14. Що таке носій інформації?

15. Яким чином записується інформація на лазерних дисках?
16. Як називається друкарський носій інформації?
17. Який принцип дії сканера?
18. Який принцип дії лазерного принтеру?
19. Що таке роздільна здатність екрану?
20. Які пристрої можуть використовуватися для підключення комп'ютера до мережі Інтернет?

3.3. Види комп'ютерів і їхня архітектура

Комп'ютери паралельних обчислень. Паралелізм для обчислювальних систем може бути здійснений на різному рівні – мови програмування, представлення даних, але самим нижнім, базовим, рівнем є наявність в самій системі фізичних можливостей організації паралельних обчислень.

У історії обчислювальної техніки паралелізм в обробці інформації процесорами був реалізований у декілька етапів.

Перший етап – це паралельна робота з розрядами слів пам'яті, коли інформація передавалася з одного слова пам'яті (групи байтів) не побітно, а цілком.

Другий етап – це паралелізм з перекриттям, одночасно з роботою центрального процесора було організовано введення і виведення даних, для цього існував периферійний процесор, що має свої команди і буферну пам'ять.

Третій етап – поява кеш-пам'яті, вибір наступної команди з оперативної пам'яті і даних для неї одночасно з виконанням поточної команди.

Четвертий етап – паралельний вибір інформації з оперативної пам'яті (розшарування пам'яті).

Це попередні етапи, паралельні обчислення в «чистому» вигляді зачалися з того моменту, коли з'явилися матричні, конвеєрні і асоціативні обчислювальні пристрої.

У **матричних** комп'ютерах був реалізований безпосередній паралелізм, операції здійснювалися одночасно над декількома даними. Наприклад, при складанні елементів двох векторів, необхідно попарно скласти всі числа, створюючі вектора. До таких же векторних операцій

відносяться обчислення скалярного твору, знаходження максимального елемента, пересилка масивів та інші.

У **конвеєрних** процесорах одночасно з роботою одного такту поступає інформація для роботи іншого такту. Паралелізм повною мірою здійснюється в разі заповнення всього конвеєра (магістралі). Прикладом таких обчислень може служити обчислення значення багаточлена в деякій точці. Кожен процесорний елемент конвеєра обчислює свій одночлен (доданок), що представляє ступінь з множником. Отриманий результат передається по магістралі, кожен процесорний елемент підсумовує набутого значення зі своїм результатом і передає далі. На виході виходить шукана сума:

$$R_n(x) = a_0 \cdot x^n + a_1 \cdot x^{n-1} + \dots + a_{n-1} \cdot x + a_n.$$

Асоціативні обчислювальні системи володіють асоціативною пам'яттю, в якій пошук інформації здійснюється по її змісту, при цьому процесорний елемент може обслуговувати групу слів пам'яті або тільки одне слово.

Комп'ютери потоків даних. Метою розробки поточкових обчислювальних систем було підвищення швидкодії на вже наявній елементній базі.

У машинах потоків даних (data flow) не процесор прочитує заздалегідь готову команду, а сама команда формується в процесі рішення задачі. Елемент пам'яті є деякою структурою, в поля якої записуються ті, що вчинили під час введення або отримані в процесі обробки дані і, коли вся структура заповнена, виконується вказана в одному з полів структури операція. У простому випадку елемент такої пам'яті має вид запису, представленого на рисунку 3.3.1.

Гніздо для першого операнда	Гніздо для другого операнда	Готовність комірки	Код операції	Адреси для розсилки результату
-----------------------------	-----------------------------	--------------------	--------------	--------------------------------

Рис.3.3.1. Елемент пам'яті машини потоків даних

В разі готовності, про що сигналізує вміст одного з полів, комірка обробляється вільним в той момент процесорним елементом. На схемі, у вигляді якої змальовується програма, така комірка позначається кружком з двома входами і одним виходом, усередині якого записана операція. Така комірка називається «актором», а дані, що переміщуються від одного актора до іншого, – фішками. Коли актор готовий до виконання, він вважається за

«зведеного», а коли його обробка закінчується – він «вистрілює», передаючи дані в інші комірки.

Принципову схему роботи комп'ютера потоків даних можна представити як кільце, по якому рухаються пакети (створювані у вічках структури). «Зведені» операндні пакети вибираються із структурованої пам'яті і передаються на виконання процесорному елементу. З процесорних елементів виходять результуючі пакети, що містять як оброблені дані, так і адреси їх розсилки. Для передачі пакетів на обробку використовується спеціальний розподільний пристрій, а для подальшої розсилки результатів – збираючий пристрій.

Машини потоків даних реалізують той природний паралелізм, який є в методі рішення даної задачі. Якщо число процесорів достатньо велике, то досягається деякий гранично можливий паралелізм.

Комп'ютери потоків даних як комп'ютери нової архітектури були складовою частиною проекту обчислювальних систем п'ятого покоління.

Список літератури

1. Войтюшенко Н. М. Інформатика і комп'ютерна техніка: Навч. пос. з баз. підготовки для студ. екон. і техн. спеціальностей ден. і заоч. форми навчання / Н. М. Войтюшенко, А. І. Остапечю – К.: Центр навчальної літератури, 2006. – 568 с.
2. Інформатика: Комп'ютерна техніка. Комп'ютерні технології. Посібник / За ред. О. І. Пушкаря. – К.: “Академія”, 2001.
3. Мельник А. Архітектура комп'ютера. – Луцьк, Луцька обласна друкарня, 2008. – 506 с.
4. Острейковский В.А. Информатика: Учебник для вузов. – М.:Выш.шк., 2000. – 511 с.
5. Соболев Б.В., Галин А.Б., Панов Ю.В., Рашидова Е.В., Садовой Н.Н. Информатика: учебник. – Изд. 3-е, дополн. И перераб. – Ростов н/Д: Феникс, 2007. – 446 с.

Розділ IV. ОПЕРАЦІЙНІ СИСТЕМИ

У цьому розділі наводяться основні відомості про операційні системи, розглядається класифікація операційних систем і описуються принципи роботи в двох найбільш популярних операційних системах – MS DOS і Windows XP.

4.1. Поняття операційної системи

Комп'ютер є обчислювальною системою, що складається з апаратної частини (hardware) і програмного забезпечення (software). Програмне забезпечення включає як написані користувачем програми, так і велику кількість різних інших, готових до роботи, програм. Різні програми можуть використовувати різні пристрої комп'ютера, які називають *ресурсами системи*. Для того, щоб спрацював деякий пристрій, про нього необхідно знати досить багато: адреса порту, до якого він підключений, адреса самого пристрою, команди кодування даних з внутрішньої форми подання в зовнішню (або навпаки) і багато що інше. Більшості користувачів такі подробиці не потрібні, їм досить вибрати об'єкт і вказати дію, яку потрібно з цим об'єктом зробити. Система управління сама підключить необхідні допоміжні програми введення/виводу і виконає необхідну команду. Також при роботі з комп'ютером виникає потреба у ряді сервісних програм, що виконують службові функції. Для того, щоб забезпечити роботу комп'ютера, з ним постачається основне програмне забезпечення — операційна система.

➤ **Операційна система (ОС)** — це сукупність програмних засобів, що здійснюють управління ресурсами ЕОМ, запуск прикладних програм і їх взаємодію із зовнішніми пристроями і іншими програмами, а також що забезпечують діалог користувача з комп'ютером.

Операційна система завантажується при включенні комп'ютера. Вона забезпечує діалог з користувачем, управління комп'ютером, його *ресурсами* (оперативною пам'яттю, місцем на дисках тощо), запускає інші *прикладні програми* на виконання.

➤ *Ресурсом* є будь-який компонент ЕОМ і можливості, що надаються ним: центральний процесор, оперативна або зовнішня пам'ять, зовнішній пристрій, програма і так далі

Операційна система забезпечує користувачеві і прикладним програмам зручний спосіб спілкування (*інтерфейс*) з пристроями персонального комп'ютера. Інтерфейс при цьому може бути апаратним, програмним і призначеним для користувача.

- *Апаратний інтерфейс* – це сукупність дротів, що зв'язують два пристрої комп'ютера.
- *Програмний інтерфейс* — це сукупність засобів, що забезпечують взаємодію пристроїв і програм в рамках обчислювальної системи.
- *Призначений для користувача інтерфейс* — це програмні і апаратні засоби взаємодії користувача з програмою або ЕОМ. У свою чергу, призначений для користувача інтерфейс може бути командним або об'єктно-орієнтованим.
- *Командний інтерфейс* передбачає введення користувачем команд з клавіатури при виконанні дій з управління ресурсами комп'ютера.
- *Об'єктно-орієнтований інтерфейс* — це управління ресурсами обчислювальної системи за допомогою здійснення операцій над об'єктами, що представляють файли, каталоги (папки), дисководи, програми, документи і так далі

Більшість операційних систем постійно модифікуються і удосконалюються у напрямі виправлення помилок і включення нових можливостей. В цілях збереження спадкоємності нова модифікація операційної системи не перейменовується, а набуває назви версії. Версії ОС позначаються (як правило) «десятковим дробом» вигляду 6.00, 2.1, 3.5 і так далі. При цьому збільшення цифри до крапки відображає істотні зміни, що вносяться до операційної системи, а збільшення цифр, що стоять після крапки — незначні зміни (наприклад, виправлення помилок). Чим більше номер версії, тим більшими можливостями володіє система.

Операційні системи обов'язково потрібні, якщо:

- обчислювальна система використовується для різних завдань, причому програми, що виконують ці завдання, потребують збереження даних і обміну ними. З цього виходить необхідність універсального механізму збереження даних; у переважній більшості випадків ОС включає реалізацію *файлової системи*. Сучасні ОС, крім того, надають можливість безпосередньо «пов'язати» виведення однієї програми з введенням іншої, минувши відносно повільні дискові операції;

- різні програми потребують виконання одних і тих же рутинних дій. Наприклад, просте введення символу з клавіатури і відображення його на екрані може зажадати виконання сотень машинних команд, а дискова операція – тисяч. Щоб не програмувати їх кожного разу наново, ОС надають *системні бібліотеки* – набори часто використовуваних підпрограм (функцій);

- між програмами і користувачами системи необхідно розподіляти повноваження, щоб користувачі могли захищати свої дані від несанкціонованого доступу, а можлива помилка в програмі не викликала збоїв в роботі самої системи і інших програм;
- необхідна можливість імітації «одночасного» виконання декількох програм на одному комп'ютері (що навіть містить лише один процесор) – *багатозадачності* – здійснюваної за допомогою прийому, відомого як «розділення часу». При цьому спеціальний компонент, званий планувальником, розділяє процесорний час на короткі відрізки і надає їх по черзі різним виконуваним програмам (процесам);
- оператор повинен мати можливість, так або інакше, управляти процесами виконання окремих програм. Для цього служить *операційне середовище*, одне з яких - оболонка і набір стандартних застосувань - є часткою ОС (інші, такі, як графічна операційна середа, утворюють незалежні від ОС прикладні платформи).

4.2. Еволюція операційних систем і основні ідеї

За попередника ОС слід вважати службові програми (завантажувачі і монітори), а також бібліотеки часто використовуваних підпрограм, що почали розроблятися з появою універсальних комп'ютерів 1-го покоління (кінець 1940-х років). Службові програми мінімізували фізичні маніпуляції оператора з устаткуванням, а бібліотеки дозволяли уникнути багатократного програмування одних і тих же дій (здійснення операцій введення-виводу, обчислення математичних функцій і т. і.).

У 1950-60-х роках сформувалися і були реалізовані основні ідеї, що визначають функціональність ОС: пакетний режим, розділення часу і багатозадачність, розділення повноважень, реальний масштаб часу, файлові структури і файлові системи.

Необхідність оптимального використання дорогих обчислювальних ресурсів привела до появи концепції «пакетного режиму» виконання програм. *Пакетний режим* передбачає наявність черги програм на виконання, причому ОС може забезпечувати завантаження програми із зовнішніх носіїв даних в оперативну пам'ять, не чекаючи завершення виконання попередньої програми, що дозволяє уникнути простою процесора.

Пакетний режим в своєму розвиненому варіанті вимагає розділення процесорного часу між виконанням декількох програм. Необхідність в розділенні часу (багатозадачності, мультипрограмуванні) виявилася ще

сильніше при розповсюдженні телетайпів (а пізніше, терміналів з електронно-променевими дисплеями) (1960-і роки) як пристроїв введення-виводу. Оскільки швидкість клавіатурного введення (і навіть читання з екрану) даних оператором багато нижче, ніж швидкість обробки цих даних комп'ютером, використання комп'ютера в «монопольному» режимі (з одним оператором) могло привести до простою дорогих обчислювальних ресурсів.

Розділення часу дозволило створити «багатокористувальницькі» системи, в яких один (як правило) центральний процесор і блок оперативної пам'яті з'єднувався з багаточисельними терміналами. При цьому частина завдань (таких, як введення або редагування даних оператором) могла виконуватися в *режимі діалогу*, а інші завдання (такі, як масивні обчислення) – в *пакетному режимі*.

Розповсюдження багатокористувальницьких систем зажадало рішення задачі розділення повноважень, що дозволяє уникнути можливості модифікації виконуваної програми або даних однієї програми в пам'яті комп'ютера іншою (що містить помилку або зловмисно підготовленою) програмою, а також модифікації самої ОС прикладною програмою.

Реалізація *розділення повноважень* в ОС була підтримана розробниками процесорів, що запропонували архітектуру з двома режимами роботи процесора – «реальним» (у якому виконуваний програмі доступний весь адресний простір комп'ютера) і «захисним» (у якому доступність адресного простору обмежена діапазоном, виділеним при запуску програми на виконання).

Застосування універсальних комп'ютерів для управління виробничими процесами зажадало реалізації *«реального масштабу часу»* («реального часу») – синхронізації виконання програм із зовнішніми фізичними процесами.

Включення функції реального масштабу часу в ОС дозволило створювати системи, одночасно обслуговуючі виробничі процеси і вирішуючі інші завдання (у пакетному режимі і (або) в режимі розділення часу).

4.3. Класифікація операційних систем

Операційні системи класифікуються по:

- кількості одночасно працюючих користувачів: однокористувацькі, багатокористувацькі;
- числу процесів, що одночасно виконуються під управлінням системи: однозадачні, багатозадачні;

- кількості підтримуваних процесорів: однопроцесорні, багатопроцесорні;
- розрядності коду ОС: 8-розрядні, 16-розрядні, 32-розрядні, 64-розрядні;
- типові інтерфейсу: командні (текстові) і об'єктно-орієнтовані (графічні);
- типові доступу користувача до ЕОМ: з пакетною обробкою, з розділенням часу, реального часу;
- типові використання ресурсів: мережеві, локальні.

Відповідно до першої ознаки класифікації багатокористувацької операційної системи, на відміну від однокористувацьких, підтримують одночасну роботу на ЕОМ декількох користувачів за одним і тим же терміналом в різний час або за різними терміналами.

Друга ознака передбачає ділення ОС на багатозадачні і однозадачні. Поняття багатозадачності означає підтримку паралельного виконання декількох програм, що існують в рамках однієї обчислювальної системи, в один момент часу. Однозадачні ОС підтримують режим виконання тільки однієї програми в окремий момент часу.

Відповідно до третьої ознаки багатопроцесорні ОС, на відміну від однопроцесорних, підтримують режим розподілу ресурсів декількох процесорів для вирішення того або іншого завдання.

Четверта ознака підрозділяє операційні системи на 8-, 16-, 32- і 64-розрядні. При цьому мається на увазі, що розрядність операційної системи не може перевищувати розрядності процесора.

Відповідно до п'ятої ознаки класифікації ОС за типом призначеного для користувача інтерфейсу діляться на об'єктно-орієнтовані (як правило, з графічним інтерфейсом) і командні (з текстовим інтерфейсом). Згідно з шостою ознакою ОС підрозділяються на системи:

- пакетної обробки, в яких з програм, що підлягають виконанню, формується пакет (набір) завдань, що вводяться в ЕОМ і виконуваних в порядку черговості з можливим пріоритетності;
- розділення часу (TSR), що забезпечують одночасний діалоговий (інтерактивний) режим доступу до ЕОМ декількох користувачів на різних терміналах, яким по черзі виділяються ресурси машини, що координується операційною системою відповідно до заданої дисципліни обслуговування;
- реального часу, що забезпечують певний гарантований час відповіді машини на запит користувача з управлінням ним якими-небудь зовнішніми по відношенню до ЕОМ подіями, процесами або об'єктами.

Відповідно до сьомої ознаки класифікації ОС діляться на мережеві і локальні. Мережеві ОС призначені для управління ресурсами комп'ютерів, об'єднаних в мережу з метою спільного використання даних, і надають потужні засоби розмежування доступу до даних в рамках забезпечення їх

цілісності і збереження, а також безліч сервісних можливостей по використанню мережевих ресурсів.

В більшості випадків мережеві операційні системи встановлюються на один або декілька достатньо потужних комп'ютерів-серверів, що виділяються виключно для обслуговування мережі і спільно використовуваних ресурсів. Всі останні ОС вважатимуться локальними і можуть використовуватися на будь-якому персональному комп'ютері, а також на окремому комп'ютері, підключеному до мережі як робоча станція або клієнт.

Основні критерії підходу до вибору операційної системи

В даний час є велика кількість операційних систем, і перед користувачем виникає завдання визначити, яка операційна система краща за інші (по тих або інших критеріях). Очевидно, що ідеальних систем не буває, будь-яка з них має свої достоїнства і недоліки. Вибираючи операційну систему, користувач повинен представляти, наскільки та або інша ОС забезпечить йому вирішення його завдань.

Щоб вибрати ту або іншу ОС, необхідно знати:

- на яких апаратних платформах і з якою швидкістю працює ОС;
- яке периферійне апаратне забезпечення ОС підтримує;
- як повно задовольняє ОС потреби користувача, тобто які функції системи;
- який спосіб взаємодії ОС з користувачем, тобто наскільки наочний, зручний, зрозумілий і звичний користувачеві інтерфейс;
- чи існують інформативні підказки, вбудовані довідники і т.п.;
- яка надійність системи, тобто її стійкість до помилок користувача, відмов обладнання і т.д.;
- які можливості надає ОС для організації мереж;
- чи забезпечує ОС сумісність з іншими операційними системами;
- які інструментальні засоби має ОС для розробки прикладних програм;
- чи здійснюється в ОС підтримка різних національних мов;
- які відомі пакети прикладних програм можна використовувати при роботі з даною системою;
- як здійснюється в ОС захист інформації і самої системи.

Найбільш відомі сімейства операційних систем: MS-DOS, MS Windows, UNIX, Linux, Mac OS і ОС реального часу.

Найпоширенішою в світі операційною системою є сімейство розроблених фірмою Microsoft операційних систем WINDOWS.

Розроблена фірмою Microsoft операційна система WINDOWS (версії 95, 98, 2000, XP, Vista, 7) враховує перераховані недоліки і реалізує ідею «поверхні письмового столу», коли все, що там знаходиться, видно одночасно

і досить «протягнути руку» і «узяти», щоб використовувати потрібний предмет. Система WINDOWS є багатозадачною, багатокористувальницькою, з об'єктно-орієнтованим інтерфейсом і має як 32-бітові так і 64-бітову версії.

Зазвичай всі версії WINDOWS ділять на декілька «груп».

Графічні інтерфейси і розширення для DOS

Ці версії WINDOWS не були повноцінними операційними системами, а лише надавали інтерфейс до можливостей операційної системи MS-DOS. До них належать операційні системи WINDOWS версій 1.02.0, 3.0, 3.1 і 3.11.

Сімейство Windows 9x

Сімейство ОС, розроблених спеціально для процесорів з 32-бітровою архітектурою на ринку позиціювали як 32-розрядні ОС, проте у складі даних продуктів була присутня деяка кількість 16-біттового коду. До складу програмних продуктів Windows 9x входила ОС MS-DOS.

Основним стратегічним завданням розробки операційних систем сімейства Windows 9x було переведення користувачів на нові 32-бітові програми при збереженні спадкоємності програм, написаних для MS-DOS.

Саме завдяки операційним системам сімейства Windows 9x комп'ютер став доступний домашнім користувачам і почав використовуватися в бізнес-процесах практично всіх комерційних підприємств і державних організацій.

Сімейство Windows NT

Операційні системи цього сімейства (Windows NT, 2000, XP, Vista) є повністю 32-бітовими операційними системами, і, на відміну від попередніх версій, не потребують підтримки з боку MS-DOS. Всі вони також є ОС з об'єктно-орієнтованим графічним інтерфейсом, що не вимагає від користувача знання спеціальних команд і уміння користуватися спеціальними програмами. Тільки у цьому сімействі представлені операційні системи для серверів.

У основу сімейства Windows NT покладено розділення адресних просторів між процесами. Кожен процес має можливість працювати тільки з виділеною йому пам'яттю. Завдяки чому істотно зросли стабільність і безпека роботи операційної системи і всього комп'ютера. Унаслідок чого дане сімейство операційних систем швидко витіснило всі попередні версії ОС WINDOWS.

Інтегровані програмні продукти

Додаткова перевага операційним системам Microsoft Windows забезпечує те, що вони включають деякі стандартні застосування, такі як інтернет-браузер (Internet Explorer), поштовий клієнт (Outlook Express), мультимедіа-програвач (Windows Media Player). За допомогою технологій COM і OLE (раніше використовувалося DDE (основоположна частка OLE)) їх

компоненти можуть вбудовуватися в інші програми, у тому числі і сторонніх виробників. Ці продукти безкоштовні і вільно розповсюджуються разом з операційною системою.

UNIX

UNIX — група *переносимих багатозадачних багатокористувацьких* операційних систем, побудованих на основі *командного інтерфейсу*, але включають як компонент об'єктно-орієнтований інтерфейс.

Перша система UNIX була розроблена в 1969 р. в підрозділі Bell Labs компанії AT&T. З тих пір було створено велику кількість різних UNIX-систем. Юридично лише деякі з них мають повне право називатися «UNIX»; останні ж, хоча і використовують схожі концепції і технології, об'єднуються терміном «UNIX-подібні» (англ. Unix-like).

Деякі відмітні ознаки UNIX-систем включають:

- використання простих текстових файлів для налаштування і управління системою;
- широке застосування утиліт, що запускаються в командному рядку;
- взаємодія з користувачем за допомогою віртуального пристрою-терміналу;
- представлення фізичних і віртуальних пристроїв і деяких засобів міжпроцесової взаємодії як файлів;
- використання конвеєрів з декількох програм, кожна з яких виконує одне завдання.

UNIX, на відміну від більшості операційних систем, була написана на мові високого рівня, а не на мові машинних кодів, що зробило доступним її використання на будь-якій апаратній платформі, на будь-якому комп'ютері.

UNIX першою надала можливість використання протоколів TCP/IP на недорогих комп'ютерах, що привело до швидкого зростання Інтернету.

В даний час ОС UNIX використовується в основному на серверах, а також як вбудовані системи для різного обладнання. На ринку ОС для робочих станцій і домашнього застосування UNIX поступилися іншим операційним системам, насамперед Microsoft Windows, хоча існуючі програмні рішення для Unix-систем дозволяють реалізувати повноцінні робочі станції як для офісного, так і для домашнього використання.

Найбільшого поширення серед всіх UNIX-подібних систем набули різноманітні версії операційної системи Linux. Вони краще всього дозволяють реалізувати повноцінні робочі станції як для офісного, так і для домашнього використання. Операційні системи Linux розглядаються як основний конкурент операційних систем Microsoft Windows. Головна конкурентна перевага систем, побудованих на основі Linux, – безкоштовність

їх використання і відкритість початкових кодів, що дозволяє легко їх розвивати і модифікувати.

LINUX

Linux – вільно поширювана система Unix, розробку якої веде велике число добровольців. Linux володіє всіма властивостями сучасної Unix-системи, включаючи справжню багатозадачність, розвинену підсистему управління пам'яттю і мережеву підсистему. *Ядро* Linux, що поставляється разом з вільно поширюваними прикладними і системними програмами, утворює повнофункціональну універсальну операційну систему.

ОС Linux розповсюджується для користувачів у вигляді інсталяційних пакетів, званих також «дистрибутивами», які створюються різними комерційними і некомерційними фірмами і організаціями.

На сьогоднішній день існує безліч різних дистрибутивів Linux, які можна розділити на дистрибутиви спільного призначення і спеціалізовані. Не дивлячись на відмінності в дистрибутивах спільного призначення, всі вони утворюють ОС Linux такою, якою її знають більшість користувачів ОС. На відміну від ядра, дистрибутиви можуть містити комерційні компоненти, і тому їх вільне розповсюдження може бути обмежено. У такому разі автори дистрибутива роблять доступними всі вільні компоненти. Зазвичай під словосполучкою «ОС Linux» розуміють дистрибутиви Linux спільного призначення.

Велика частина ядра ОС Linux написана на мові Cі, завдяки чому система достатньо легко переноситься на різну апаратну архітектуру. Крім того, існує багато варіантів Linux, поширюваних окремо від офіційного ядра. ОС Linux здатна працювати на багатопроесорних системах, забезпечуючи ефективне використання всіх процесорів, так само як і на малопотужних мобільних пристроях.

Ядро ОС Linux спочатку розроблялося як багатокористувацька, багатозадачна операційна система. Вона побудована за модульним принципом – кожен компонент операційної системи представлений окремою програмою або комплектом програм. Файлова система, командний інтерпретатор (призначений для користувача інтерфейс, що сприймає команди користувача і організує їх виконання), графічне операційне середовище, офісний пакет, система управління базами даних – все це є самостійними модулями, які можуть створюватися різними розробниками і, більш того, різні команди програмістів можуть створювати свої власні варіанти цих модулів. Цим пояснюється наявність декількох файлових систем (ext3fs reiserfs jfs), створених спеціально для ОС Linux, разом з реалізацією підтримки більше 20 сторонніх файлових систем (у тому числі і використовуваних в ОС Windows – FAT і NTFS).

Безпосередньо після завантаження ОС Linux функціонує в командному текстовому режимі. Кожна дія здійснюється шляхом набору команди з клавіатури з подальшим виведенням тексту на екран. Графічне операційне середовище розробляється і використовується як окремий програмний комплекс. Запускається графічна оболонка на вимогу користувача (або включається в команди автозавантаження) і дозволяє здійснювати управління операційною системою і програмами в звичному інтуїтивно зрозумілому графічному інтерфейсі.

Найбільш поширені графічні віконні оболонки для ОС Linux KDE і GNOME. Працювати в графічній оболонці можна з використанням миші: вибір піктограм, командних кнопок, маніпулювання іншими елементами об'єктно-орієнтованої операційної системи.

Невід'ємною частиною ОС Linux є інтернет-браузери (декілька проектів, що розробляються різними командами програмістів), програми для організації роботи з електронною поштою, обміну миттєвими повідомленнями, телеконференцій і інших засобів телекомунікацій, а також пакети офісних програм для роботи з документами.

Найбільш популярним офісним пакетом для ОС Linux є офісний пакет OpenOffice.org. Для графічних віконних оболонок KDE і GNOME розробляються свої версії офісних пакетів, відповідно KOffice і GnomeOffice. Окрім них доступні комерційні офісні пакети від багатьох інших виробників. Всі вони, як правило, дають можливість повноцінної роботи з текстовими і табличними документами, створення презентацій, доступу до баз даних, виведення результатів на друк.

Для Linux існують спеціальні версії всіх поширених систем управління базами даних, систем управління підприємством, пакетів програм для роботи з графікою. Існують комплекти мультимедіа-програм для відтворення аудіо- і відеоінформації.

Велика частина прикладного програмного забезпечення не розробляється виключно для Linux, а працює і на інших Unix-подібних системах.

Завдяки безкоштовності і можливості вільної модифікації початкових кодів Linux добре підходить для використання в навчальному процесі, для установки на недорогі комп'ютери. Існують також версії для мобільних пристроїв і вбудовувані версії для використання з різним обладнанням.

Mac OS

Mac OS (*Macintosh Operating System*) — сімейство операційних систем з графічним інтерфейсом, розроблене корпорацією «Apple» (раніше Apple Computer) для своєї лінійки комп'ютерів Macintosh. Популяризація графічного інтерфейсу користувача у сучасних операційних системах часто

вважається за заслугу Mac OS. Вона була вперше представлена в 1984 году разом з оригінальним Macintosh 128K.

«Apple» свідомо зменшувала значення існування операційної системи в ранні роки існування Макінтоша, щоб інтерфейс сприймався як більш дружній до користувача і віддалити Mac OS від інших операційних систем, таких як MS DOS, які представлялися заплутаними і важкими в освоєнні і використанні. «Apple» переслідувала мету, щоб Макінтош представлявся як комп'ютер «для всіх останніх» («for the rest of us»). Сам термін «Mac OS» насправді не існував до тих пір, поки не був офіційно використаний в середині 1990-х років. З тих пір термін застосовується до всіх версій операційних систем Макінтоша як зручний спосіб виділення їх в контексті інших операційних систем.

Ранні версії Mac OS були сумісні тільки з комп'ютерами Макінтош. З недавніх пір Mac OS X стала сумісна з архітектурою Intel x86. Але політика фірми Apple така, що вона дозволяє встановлювати систему Mac OS тільки на комп'ютери Apple.

Операційна система Mac OS стала першою ОС, в якій користувач управляв своїм комп'ютером не лише командами, що вводилися з клавіатури, і інструкціями, але і за допомогою нового в ті часи пристрою, названого мишею. Миша дозволяла управляти покажчиком екрані монітора. Крім того, операційна система Mac OS використовувала новий «віконний» графічний інтерфейс для подання і організації інформації.

Останні новітні версії операційної системи Mac OS побудовані на ядрі операційної системи LINUX.

Питання і завдання для самостійної роботи

1. Дайте визначення операційної системи.
2. Для чого використовуються операційні системи?
3. Що є пакетним режимом виконання програм?
4. По яких ознаках класифікуються операційні системи?
5. Які основні критерії підходу до вибору ОС?
6. Перерахуйте найбільш відомі сімейства операційних систем.
7. Охарактеризуйте об'єктно-орієнтований графічний інтерфейс.
8. Який зв'язок існує між сімействами ОС UNIX і LINUX?
9. Яку ОС розробила корпорація «Apple»?
10. Перерахуйте сімейства MS Windows і їх хаактерні особливості?

11. Перерахуйте інтегровані програмні продукти ОС Windows.
12. Які Ви знаєте операційні системи в залежності від типу інтрефейсу?

4.4. Операційна система MS DOS

Найпоширенішою в світі операційною системою є розроблена фірмою Microsoft в 1981 році разом з комп'ютером IBM PC дискова операційна система MS DOS. Ця однозадачна, однокористувальницька операційна система стала попередником ОС Windows.

Розглянемо структуру MS DOS, представлену на рисунку 4.4.1.

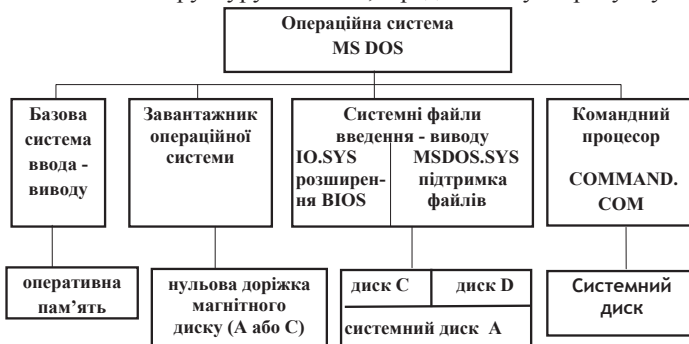


Рис. 4.4.1. Компоненти MS-DOS і їх розміщення в пам'яті комп'ютера до початку роботи операційної системи

Операційна система завантажується в пам'ять комп'ютера і робить його готовим до роботи в наступних випадках:

- 1) при включенні комп'ютера;
- 2) при натисненні кнопки **RESET**.

Процес завантаження відбувається таким чином. **BIOS** — базова система введення-виводу, розташована в ПЗП (постійному запам'ятовувальному пристрої) або в оперативній пам'яті, перевіряє працездатність пам'яті і підключеного до комп'ютера обладнання. Потім вона шукає програму початкового завантаження системи і передає їй управління.

Розглянемо роботу BIOS детальніше. Окрім програм, що забезпечують запуск операційної системи, до складу базової системи введення-виводу входять програми обслуговування зовнішніх пристроїв. Їх робота організована через систему переривань: звернення до пристрою з якої-небудь програми викликає своє переривання. *Переривання* — це примусова передача управління від виконуваної програми до системи. Кожне переривання має свій номер, згідно якому викликається сервісна програма, що виконує декілька функцій. Так, програма по роботі з дискетою виконує шість функцій: переводить накопичувач на гнучкому магнітному диску в початковий стан, отримує код стану накопичувача з докладною про нього інформацією, читає і записує один або декілька секторів на доріжці, перевіряє дані після операції читання або запису, формує доріжку дискети. Всі ці функції, або деякі з них, виконуються по одній команді копіювання файлів або команді виводу даних на дискету.

Завантажувач операційної системи — програма початкового завантаження — розміщується завжди в першому секторі нульової доріжки дискети (BOOT — секторі). Якщо дискета не знайдена, програма шукається на такій же доріжці твердого диску. Програма початкового завантаження прочитується засобами BIOS в оперативну пам'ять, завжди в одне і те ж місце. BIOS запускає цю програму, і вона починає переписувати в пам'ять (завантажувати) файли введення-виводу системи.

Системні файли введення-виводу — програми додаткових засобів введення-виводу, після завантаження вони знаходяться постійно в оперативній пам'яті.

IO.SYS дозволяє врахувати особливості конкретної операційної системи, усунути помилки введення-виводу і організувати роботу з новими зовнішніми пристроями. Коли ця програма починає працювати, вона шукає файл з допоміжною інформацією CONFIG.SYS. У цьому файлі указується, які пристрої мають бути підключені при завантаженні системи. Для кожного пристрою існує своя програма підтримки, звана *драйвером*. Драйвери стандартних пристроїв складають файл IO.SYS. Імена драйверів, що зберігаються в інших файлах, команди по їх запуску, а також установка інших параметрів системи містяться у файлі CONFIG.SYS. Якщо цей файл не знайдений, параметри системи встановлюються за умовчанням. Комплекс підпрограм, що містяться у файлі **MSDOS.SYS**, дозволяє управляти оперативною і дисковою пам'яттю, організацією обчислювального процесу. У персональних комп'ютерах головним є завдання розподілу дискового простору. При цьому реалізуються наступні принципи: 1) всі файли даних зберігаються в секторах стандартного розміру; 2) сектори виділяються за мірою необхідності; 3) забезпечується логічний зв'язок між секторами, що

дозволяє користувачеві не піклуватися про те, на скільки частин система поділила його файл, і де ці частини розташовані; 4) кожен диск містить кореневий каталог, в якому зберігається перелік розташованих на диску файлів. Підпрограми другого системного файлу введення-виводу також управляють проходженням програми користувача: завантажують, контролюють виконання і завершення.

Командний процесор **COMMAND.COM** прочитує команди системи, що вводяться з клавіатури, і виконує їх. Командний процесор розбитий на три частини. Перша прочитується услід за системними файлами введення-виводу і залишається постійно в оперативній пам'яті. Друга — зберігається тимчасово і використовується для пошуку і виконання командного файлу **AUTOEXEC.BAT**. У ньому вказані програми і команди, що виконуються при кожному запуску комп'ютера, наприклад, драйвер клавіатури для роботи з російськими або українськими літерами. Після виконання файлу **AUTOEXEC.BAT** процес завантаження завершується. Третя частина командного процесора містить таблицю команд системи і їх інтерпретатор, і може бути видалена іншою програмою з пам'яті. В цьому випадку вона, при необхідності, повторно завантажується з диска. Команди системи діляться на внутрішні і зовнішні. Внутрішні команди містяться в таблиці, зовнішні можуть бути іменами файлів з розширеннями **COM**, **EXE** або **BAT**. Якщо командний процесор не знаходить команду в таблиці, він вважає її зовнішньою і шукає файл з таким ім'ям і можливим розширенням. Файл **AUTOEXEC.BAT** може містити команду **nc** і тоді завантажується система **NORTON COMMANDER**.

Для зручності роботи з операційною системою **MS DOS** фірма **Peter Norton Computing** розробила одну з найпопулярніших програм-оболонок — систему **NORTON COMMANDER**. Вона дозволяє наочно зображувати зміст каталогів на дисках, копіювати, перейменовувати і пересилати файли, переглядати і редагувати створені в різних програмних системах файли, виконувати команди **DOS** натисненням тільки одної клавіші і робити багато що інше.

Робота з файлами і каталогами в MS DOS. Розглянемо деякі основні команди, які може використовувати користувач для ведення діалогу з операційною системою **MS DOS**. Коли **OC MS DOS** готова до діалогу, користувач бачить на екрані запрошення **MS DOS**, що полягає, як правило, з інформації про поточний дисковод і, можливо, поточний каталог. Наведемо приклади запрошення **MS DOS**:

C:\> приклад запрошення **MS DOS**, що містить інформацію про поточний дисковод

C:\WIN> приклад запрошення **MS DOS**, що містить інформацію про поточний дисковод і поточний каталог

Формат запису команд в MS DOS має на увазі написання імені команди або ключового слова (наприклад, cd, md, dir) після запрошення MS DOS, при необхідності завдання параметрів команди – об'єктів, над якими виконується команда (наприклад, ім'я файлу або ім'я каталога) і ключів – режимів виконання команди (наприклад, /p, /w). Ключ складається з косої риси і букви латинського алфавіту. Параметри і ключі є необов'язковими складовими команди. Існують команди, що складаються з одного ключового слова (наприклад, cls, time, date). Після перевірки правильності синтаксису команди користувач повинен підтвердити введення команди натисканням на клавішу ENTER. До підтвердження команди користувач має можливість редагувати команду за допомогою клавіш Del і Backspace.

Команди роботи з файлами в MS DOS. Розглянемо команди створення, копіювання, перейменування, переміщення і видалення файлів.

Створення файлу в MS DOS. Створити невеликий текстовий файл можна за допомогою команди **copy con**. Формат запису команди має вигляд:

copy con ім'я_файлу

Після введення команди та її підтвердження користувач повинен ввести текст файлу з клавіатури, натискаючи в кінці кожного рядка клавішу ENTER. Після введення останнього рядка тексту користувач повинен натиснути функціональну клавішу F6, а потім клавішу ENTER. Ця команда інтерпретується як копіювання з файлу con (з клавіатури) тексту в деякий текстовий файл з ім'ям ім'я_файлу.

Копіювання файлу в MS DOS. Копіювання файлу здійснюється за допомогою команди **copy**. Формат запису команди має вигляд:

copy ім'я_файлу1 ім'я_файлу2

В результаті виконання такої команди створюється файл з ім'ям ім'я_файлу2 в поточному каталозі і вміст файлу з ім'ям ім'я_файлу1 в поточному каталозі копіюється у файл з ім'ям ім'я_файлу2. При виконанні цієї команди слід враховувати, що якщо в поточному каталозі вже існує файл з ім'ям ім'я_файлу2, то в результаті команди його вміст буде повністю заміщений вмістом файлу з ім'ям ім'я_файлу1 без яких-небудь попереджень. Наведемо приклади виконання команди **copy**:

copy first.txt second.txt копіювання вмісту файлу first.txt поточного каталогу в файл second.txt поточного каталогу

copy c:*.* d: копіювання всіх файлів кореневого каталогу диску C: в кореневий каталог диску D:

copy c:*.* d: копіювання всіх файлів кореневого каталога диску C: в поточний каталог диску D:

copy c:*.* копіювання всіх файлів кореневого каталога диску C: в поточний каталог поточного диску

Перейменування файлу в MS DOS. Перейменування файлу здійснюється за допомогою команди **ren**. Формат запису команди має вигляд:

ren ім'я_файлу1 ім'я_файлу2

В результаті виконання такої команди файл з ім'ям **ім'я_файлу1** в поточному каталозі отримує ім'я **ім'я_файлу2**. Наведемо приклад виконання команди **ren**.

ren first.txt second.txt файл first.txt поточного каталогу отримує ім'я second.txt

Переміщення файлу в інший каталог. Переміщення файлу в інший каталог здійснюється за допомогою команди **move**. Формат запису команди має вигляд:

move [/Y] ім'я_файлу ім'я_каталога

В результаті виконання такої команди файл з ім'ям **ім'я_файлу** переміщається з поточного каталога в каталог з ім'ям **ім'я_каталога**. Команда містить необов'язковий параметр [/Y]. При його наявності MS DOS буде видавати запит Are you sure (Ви упевнені) (Y/N)? Для виконання команди користувач повинен натиснути на клавішу Y, а потім на клавішу ENTER. Для відміни команди – клавішу N, а потім ENTER. Наведемо приклад виконання команди **move**.

move *.* d:\New переміщення всіх файлів поточного каталога в каталог New диску D:

Видалення файлу в MS DOS. Видалення файлу здійснюється за допомогою команди **del**. Формат запису команди має вигляд:

del ім'я_файлу

В результаті виконання такої команди з поточного каталога видаляється файл з ім'ям **ім'я_файлу**. Наведемо приклади виконання команди **del**:

del first.txt видалення з поточного каталога файлу first.txt

del c:*.* видалення всіх файлів з кореневого каталога диску C:, при цьому DOS видає запит Are you sure (Y/N)? Для підтвердження видалення слід натиснути клавішу Y, а потім клавішу ENTER, для скасування команди – клавішу N, а потім клавішу ENTER

З'єднання (конкатенація) файлів в MS DOS. Конкатенація файлів виконується за допомогою команди **copy**. Формат запису команди має вигляд:

copy ім'я_файлу1+ім'я_файлу2

В результаті виконання цієї команди вміст файлу з ім'ям **ім'я_файлу2** додається в кінець вмісту файлу з ім'ям **ім'я_файлу1**.

Виведення вмісту файлу на екран. Виведення вмісту файлу на екран здійснюється за допомогою команди **type**. Формат запису команди має вигляд:

type ім'я_файлу

Наведемо приклад виконання команди **type**:

type c:\first.txt вивід на екран вмісту файла first.txt, що розташований в кореневому каталозі диску C

Очищення екрану монітора. Очищення екрану здійснюється за допомогою команди **cls**. Після натиснення на клавішу ENTER екран очищається і на екрані з'являється запрошення DOS.

Виведення вмісту файлу на друк. Виведення вмісту файлу на друк здійснюється за допомогою команди **copy**. Формат запису команди має вигляд:

copy /b ім'я_файлу prn

При використанні цієї команди важливо, щоб принтер був включений і готовий до друку.

Наведемо приклад виконання команди **copy**:

copy /b c:\first.txt prn

вивід на принтер вмісту файла first.txt, що розташований в кореневому каталозі диску C

Зміна поточного дисководу. Після запрошення DOS необхідно набрати ім'я дисководу, який повинен стати поточним, ввести двокрапку і натиснути клавішу ENTER. Наприклад, для того щоб зробити поточним дисковод D слід ввести команду:

C:\>D:

Після натискання на клавішу ENTER поточним дисководом стане дисквід D

Зміна поточного каталога. Зміна поточного каталога здійснюється за допомогою команди **cd**. Формат запису команди має вигляд:

cd [дисквід:] шлях

Наприклад, для переходу в кореневий каталог поточного диска слід виконати команду: **cd**

Проглядання вмісту каталога. Виведення змісту каталога здійснюється за допомогою команди **dir**. Формат запису команди має вигляд:

dir (дисквід:) (шлях\) (ім'я_файлу) (параметри)

За допомогою параметрів користувач може установити повноекранне виведення змісту (установка параметру **/P**) або вивід в широкому форматі (установка параметру **/W**). Наведемо приклади виконання команди **dir**:

dir

вивід на екран змісту поточного каталогу

dir *.doc

вивід на екран відомостей про файли поточного каталогу з розширенням doc

dir >prn

вивід змісту поточного каталогу на принтер

Створення каталога. Створення каталога здійснюється за допомогою команди **md**. Формат запису команди має вигляд:

md (дисковід:) шлях

Наведемо приклади виконання команди **md**:

md dir1 в поточному каталозі буде створено підкаталог з ім'ям dir1

md c:\dir1 створення підкаталогу з ім'ям dir1 в кореневому каталозі диску C

Перейменування каталога. Перейменування каталога здійснюється за допомогою програми **move**, що входить в MS DOS, починаючи з 6 версії. Формат запису команди має вигляд:

move і'мя_існуючого_каталога нове_і'мя_каталога

Наведемо приклади виконання команди **move**:

move dir1 dir2 підкаталог dir1 поточного каталогу буде перейменовано в каталог з ім'ям dir2

move c:\dir1 dir2 підкаталог dir1 кореневого каталога диску C буде перейменовано в каталог з і'мям dir2

Видалення порожнього каталога. Видалення порожнього каталога здійснюється за допомогою команди **rd**. Формат запису команди має вигляд:

rd (дисковід:) шлях

Наведемо приклади виконання команди **rd**:

rd dir1 видалення порожнього підкаталогу dir1 поточного каталогу

rd c:\dir1 видалення порожнього підкаталогу dir1 кореневого каталога диску C

Видалення непорожнього каталога або файла. Видалення непорожнього каталога здійснюється за допомогою програми **deltree**, що входить в MS DOS, починаючи з 6 версії. Формат запису команди має вигляд:

deltree ім'я_файла_або_ім'я_каталога (/Y)

Відзначимо, що при використанні параметра /Y в записі команди видалення здійснюватиметься без запиту на підтвердження команди.

Наведемо приклади виконання команди **deltree**:

deltree dir1	видалення підкаталога dir1 (разом з усім його вмістом) поточного каталога або файлу з ім'ям dir1 поточного каталога
deltree /y dir1	видалення підкаталога dir1 (разом з усім його вмістом) поточного каталога або файлу з ім'ям dir1 поточного каталога без запиту на підтвердження команди

Питання і завдання для самостійної роботи

1. Що є ресурсами обчислювальної системи?
2. Для чого необхідна операційна система?
3. Перерахуйте основні компоненти операційної системи MS-DOS. Де вони знаходяться до і після завантаження системи?
4. У яких випадках операційна система завантажується в пам'ять комп'ютера?
5. Які функції виконує при завантаженні базова система введення-виводу BIOS?
6. Які програми входять до складу BIOS?
7. Що таке переривання, для чого воно використовується?
8. Де розташований завантажувач операційної системи MS DOS?
9. Яким чином операційна система завантажує сама себе?
10. Що таке драйвер?
11. Що є системними файлами введення-виводу?
12. Яку інформацію містить файл CONFIG.SYS?
13. Яке головне завдання і як вирішує комплекс програм файлу MSDOS.SYS?
14. Для чого необхідний командний процесор?

15. Зі скількох частин складається командний процесор і як вони розміщуються в пам'яті комп'ютера?
16. Які допоміжні файли використовують системні файли введення-виводу і командний процесор?
17. Як називається програмна оболонка, що дозволяє зробити роботу з операційною системою MS DOS більш ефективнішою?
18. Коли завершується процес завантаження операційної системи?
19. Що є зовнішніми командами операційної системи MS DOS?
20. Перерахуйте програми операційної системи, що постійно знаходяться в оперативній пам'яті під час роботи комп'ютера.

4.5. Операційна система WINDOWS XP

Операційна система Windows XP в російській редакції була випущена в двох версіях – Windows XP Home Edition і Windows XP Professional – як операційна система, що прийшла на зміну версіям Windows 9x/ME і Windows NT/2000. При появі кожної нової версії Windows розробники прагнуть максимально спростити діалог користувача з операційною системою, зберігши всі кращі якості попередніх версій. ОС Windows XP Professional орієнтована, насамперед, на роботу в середніх і великих мережах, тобто на роботу корпоративних користувачів, а ОС Windows XP Home Edition орієнтована на домашніх користувачів з можливістю спрощеного налаштування мережевих засобів і доступу в Інтернет, використання засобів мультимедіа і підтримки комп'ютерних ігор.

Опишемо основи роботи в операційній системі Windows XP Home Edition. Після завантаження ОС Windows XP користувач на екрані бачить робочий стіл. Подібно до звичайного робочого столу користувач може розмістити тут основні елементи, призначені для його комфортної роботи. ОС Windows XP представляє широкі можливості для зручного налаштування робочого столу, які розглянемо далі в цьому пункті. Спочатку на робочому столі розташовані тільки *Кошик*. Решта значків *Мій комп'ютер* і *Мої документи* розташовані в єдиному меню *Пуск*. У заголовку меню *Пуск* вказано ім'я, під яким користувач реєструвався в системі. Меню *Пуск* складається з лівої і правої колонки значків, наведених на рисунку 4.5.1. Ліва колонка містить програми, що найбільш часто запускаються, а права колонка – посилання на основні системні папки (наприклад, *Мій комп'ютер*, *Мої документи*, *Мої малюнки*, *Моя музика*), *Панель управління*, *Довідку* і *підтримку*, *Пошук* і *тому подібне*. У лівій колонці є можливість переглядання

списку встановлених програм за допомогою клацання лівою кнопкою миші по списку *Всі програми*.

Розглянемо деякі елементи *Панелі управління*. Для запуску *Панелі управління* користувачеві слід виконати одинарне клацання лівою кнопкою миші по пункту *Панель управління* в правій колонці меню *Пуск*. Діалогове вікно *Панель управління* представлено на рисунку 4.5.2. У лівій частині вікна є можливість зміни способу відображення елементів панелі управління – перемикання до класичного вигляду. *Панель управління* містить наступні категорії: *Оформлення і теми, Мережа і підключення до Інтернету, Установка і видалення програм, Звук, мова і аудіопристрої, Продуктивність і обслуговування, Принтери і інше обладнання, Облікові записи користувачів, Дата, час, мова і регіональні стандарти, Спеціальні можливості, Центр забезпечення безпеки*.

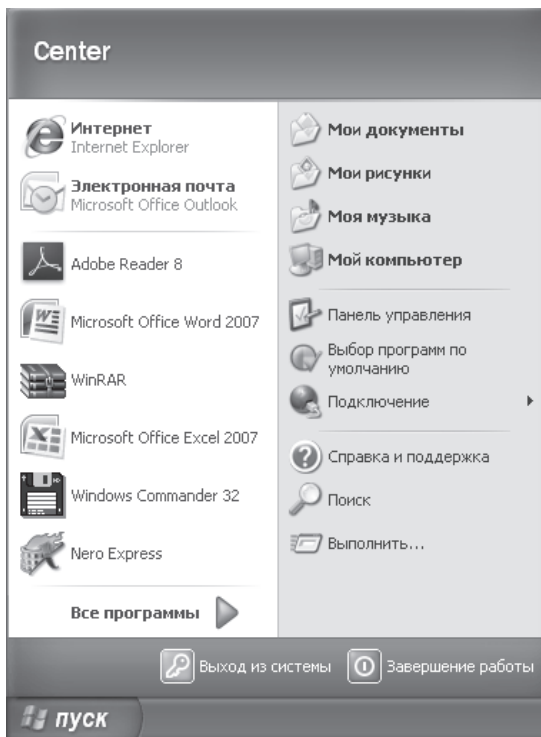


Рис.4.5.1. Меню Пуск

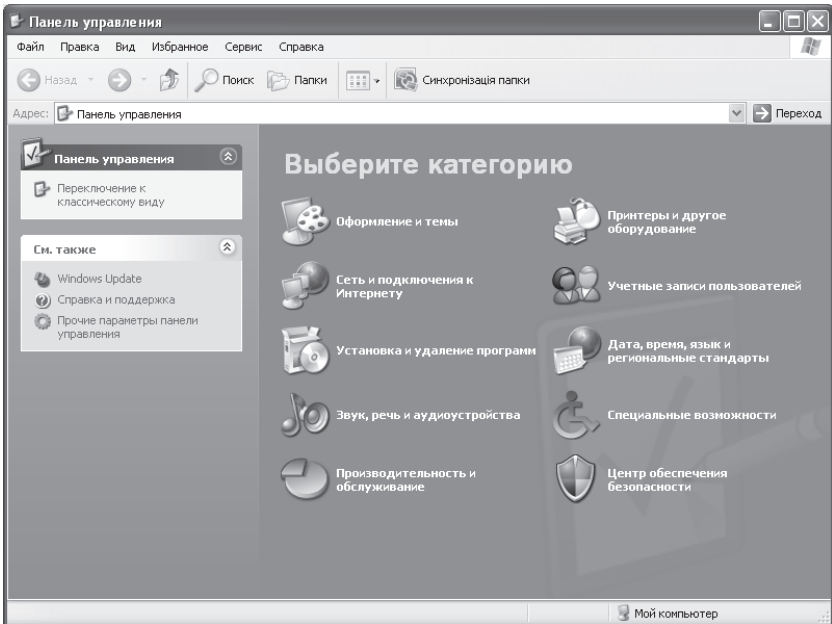


Рис.4.5.2. Категорії Панелі управління

Розглянемо деякі можливості налаштування призначеного для користувача інтерфейсу за допомогою елементів *Панелі управління*. Для графічного оформлення призначеного для користувача інтерфейсу слід використовувати категорію *Оформлення і теми Панелі управління*.

Категорія Оформлення і теми Панелі управління. Для додавання значків на робочий стіл, зміни теми робочого столу або заставки користувач повинен вибрати категорію *Оформлення і теми*. В результаті на екрані з'явиться однойменне діалогове вікно. У діалоговому вікні *Оформлення і теми* користувачеві необхідно клацнути лівою кнопкою миші по значку *Екран*. Діалогове вікно *Властивості:Екран*, що з'явилося в результаті цієї дії містить вкладки *Теми*, *Робочий стіл*, *Заставка*, *Оформлення*, *Параметри*. Викликати це діалогове вікно можна також, натиснувши праву кнопку миші у вільній області робочого столу і вибравши пункт *Властивості*.

Перемкнувшись на вкладку *Теми* (вона активна за умовчанням) діалогового вікна **Властивості:Екран**, користувач може налаштувати тему робочого столу. Тема робочого столу – це фоновий малюнок робочого столу, який може бути вибраний одинарним клацанням миші по списку Тем, що розкривається. Список, що розкривається, дозволяє вибрати як готові (стандартні) теми, так і завантажити свою тему при виборі пункту *Огляд*. Є також можливість завантаження теми з Інтернету. Для цього в списку *Теми*, що розкривається, користувачеві необхідно вибрати пункт *Теми з Інтернету*. У області *Зразок* користувач може спостерігати, як зміниться фоновий малюнок робочого столу. Після вибору теми користувачеві слід підтвердити команду натисненням на кнопку *Застосувати* або *ОК*. Діалогове вікно з активною вкладкою **Теми** представлено на рисунку 4.5.3.

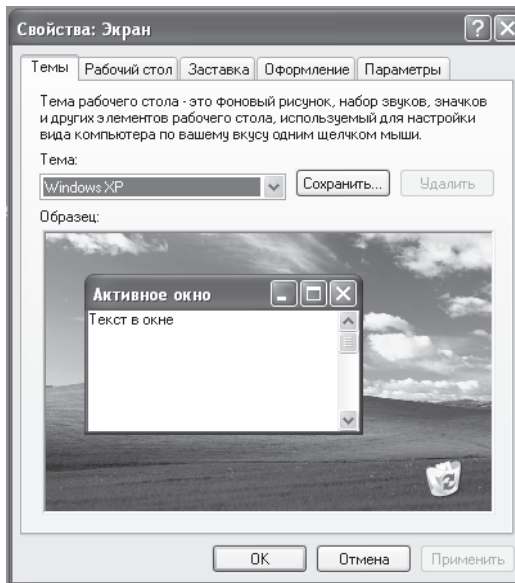


Рис. 4.5.3. Діалогове вікно Властивості: Екран з активною вкладкою Тем **Властивості:Екран**, користувач може налаштувати розташування фонового малюнка. При натисненні на кнопку *Налаштування робочого столу...* на екрані з'являється діалогове вікно **Елементи робочого столу**, в якому

користувач може вказати значки, які слід розмістити на робочому столі. Наприклад, користувач може розмістити значок *Мій комп'ютер*, встановивши прапорець напроти цього значка в розділі *Значки робочого столу*, а також змінити на свій розсуд вигляд значка. У цьому діалоговому вікні користувач також може очистити робочий стіл, тобто видалити невживані протягом певного періоду часу елементи робочого столу, або встановити автоматичне очищення робочого столу через встановлену кількість днів. Діалогове вікно Елементи робочого столу представлено на рисунку 4.5.4.

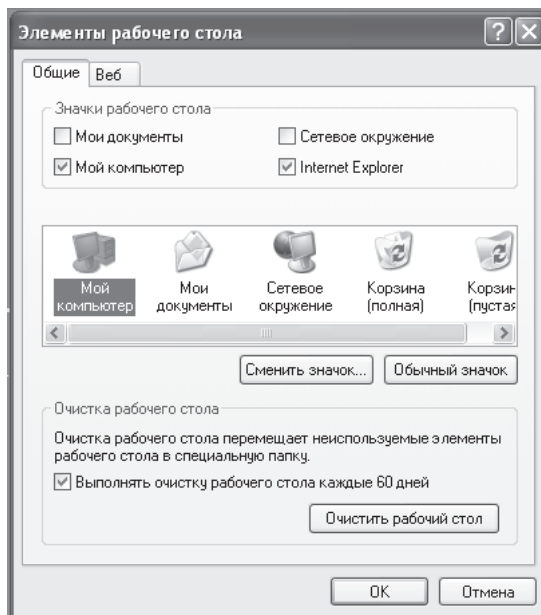


Рис.4.5.4. Діалогове вікно Елементи робочого столу з активною вкладкою Загальні

Перемкнувшись на вкладку *Заставка* діалогового вікна **Властивості:Екран**, користувач може вибрати заставку, вказати параметри, встановити часовий інтервал, а також має можливість захисту паролем. Натиснувши на кнопку Перегляд, користувач може побачити результат виконаних ним налаштувань. На вкладці *Заставка* в розділі

Енергозбереження є можливість налаштування живлення монітора. При натисненні на кнопку *Живлення...* користувач може встановити, наприклад, часовий інтервал простою, через який комп'ютер перейде в сплячий режим. Діалогове вікно *Елементи робочого столу* з активною вкладкою *Заставка* представлено на рисунку 4.5.5.

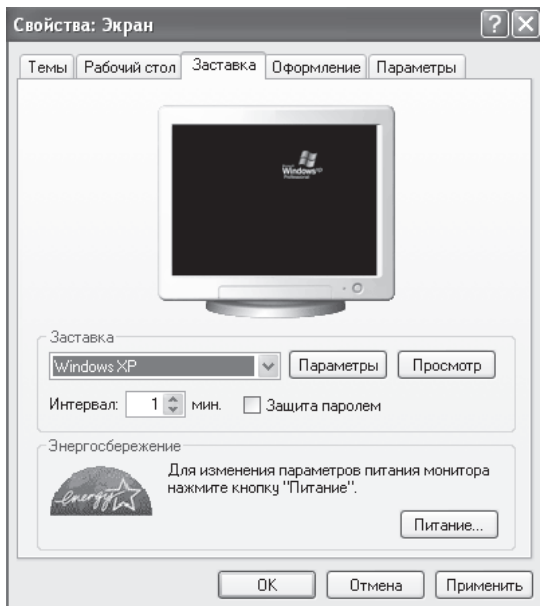


Рис.4.5.5. Діалогове вікно Властивості: Екран з активною вкладкою Заставка

Перемкнувшись на вкладку *Оформлення* діалогового вікна *Властивості: Екран*, користувач може налаштувати стиль оформлення вікон і кнопок (стиль Windows XP або класичний стиль), змінити колірну схему або розмір шрифту, які використовуються у вікнах Windows XP, а також встановити ефекти для відображення вікон.

На вкладці *Параметри* діалогового вікна *Властивості:Екран* користувач може настроїти розрішення екрану і якість перенесення кольорів.

При виборі значка *Панель завдань і меню "Пуск"* в категорії *Оформлення і теми Панелі управління* на екрані з'являється діалогове вікно *Властивості панелі завдань і меню «Пуск»*, представлено на рисунку 4.5.6. Це діалогове вікно містить дві вкладки: *Панель завдань* (вкладка активна за

умовчанням) і Меню «Пуск». На вкладці Панелі завдань користувач може налаштувати оформлення панелі завдань і області повідомлень. Зручною для користувача Windows XP є можливість закріплення панелі завдань і угруповання схожих кнопок панелі завдань. На вкладці меню «Пуск» користувач може натиснути на кнопку *Налаштувати...* і в діалоговому вікні, що з'явилось, встановити розмір значків цього меню, кількість ярликів часто використовуваних програм, перерахованих в лівій колонці меню «Пуск».

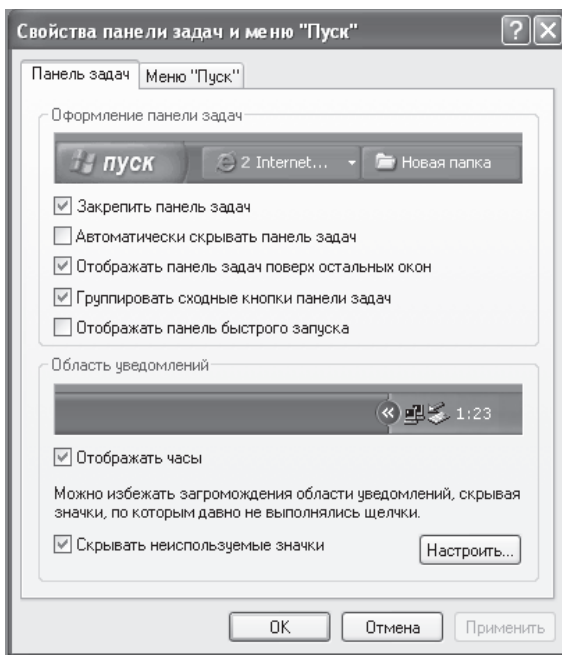



Рис.4.5.6. Діалогове вікно Властивості панелі завдань і меню «Пуск» з активною вкладкою Панель завдань

Для звукового оформлення призначеного для користувача інтерфейсу зручно використовувати категорію *Звук, мова і аудіопристрої*.

Категорія Звук, мова й аудіопристрої Панелі управління. У цій категорії, вибравши відповідне завдання або значок, користувач може

настроїти гучність, змінити звукову схему, змінити параметри настроювання динаміків і т.і. Розглянемо, як можна озвучити будь-яку операцію або змінити звук операції при роботі з операційною системою. У категорії *Звук, мова й аудіопристрої* користувачеві слід вибрати завдання *Змінити звукову схему*. У результаті на екрані з'явиться діалогове вікно **Властивості:Звуки й аудіопристрої** з активною за замовчуванням вкладкою *Звуки*, представлено на рисунку 4.5.7. Серед програмних подій слід вибрати операцію (програмну подію), звук (звукову схему) якої прагне змінити (призначити) користувач. Наприклад, призначимо звук програмній події *Друк завершено*. Для цього в однойменному списку, що розкривається, користувачеві треба вибрати звук, наприклад, *Windows XP – друк завершено* або призначити звукову схему.

Поруч зі списком *Звуки*, що розкривається розташована кнопка , яка дозволяє користувачеві (при її натисканні) послухати обраний звук.

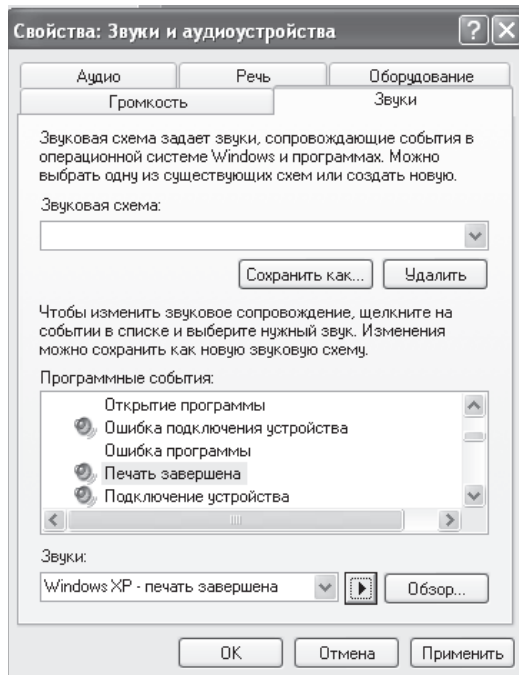


Рис. 4.5.7. Діалогове вікно Властивості: Звуки і аудіопристрої з активною вкладкою Звуки

Для збереження звукової схеми з індивідуальним налаштуванням користувачеві необхідно натиснути на кнопку *Зберегти як...* і в діалоговому вікні Збереження схеми, що розкрилося, ввести назву для створеної звукової схеми. Для підтвердження команди слід натиснути на кнопку *ОК*.

При графічному і звуковому настроюванні призначеного для користувача інтерфейсу слід пам'ятати, що зайве нагромадження звуковими і графічними ефектами призводить до витрачання ресурсів і, як наслідок, до зниження продуктивності операційної системи.

Робота в будь-якій операційній системі – це, перш за все, робота з файлами, папками (каталогами) і програмами.

➤ **Файл** – це іменована область пам'яті, що має унікальне ім'я.

Імена файлів можна побачити на екрані дисплея під час роботи з будь-яким програмним забезпеченням. Ім'я файлу часто говорить про його вміст. Файл утворює і програма, що зберігається на диску, і дані різних користувачів, і гра (яка часто включає декілька файлів). Ім'я файлу складається з двох слів, розділених при написанні крапкою або пропуском в змісті файлів. Перше слово придумує автор файлу, піклуючись про те, щоб в імені відбивався зміст файлу. Наприклад, якщо це програма — гра, то назва цієї гри. Друге слово, зване також розширенням файлу, показує, до якої категорії належить файл. Розширення складається з трьох букв. Існують стандартні розширення, по яких можна визначити, що містить файл: готову до запуску програму, опис або допоміжні дані і програми. Деякі файли мають стандартні імена, які не можна змінювати.

Перерахуємо значення розширень:

.exe, com – програма машинною мовою, готова до виконання;

.txt, doc — текстові файли, містять описи готового програмного забезпечення або створені користувачем;

.jpg, bmp, pcx – файли малюнків (зображень);

.arj, lzh – файли, поміщені в архів.

Проглядання файлів і папок за допомогою Провідника в ОС Windows XP. Для проглядання файлів і папок в ОС Windows XP можна скористатися програмою Провідник. Для запуску Провідника користувачеві слід натиснути кнопку *Пуск*, розкрити список *Всі програми*, в якому вибрати *Стандартні*, а в списку *Стандартні*, що розкрився, слід вибрати пункт *Провідник*. Вікно *Провідник* представлено на рисунку 4.5.8 і розділено на дві частини. При клацанні по одній з папок в лівій частині вікна вміст цієї папки відображується в правій частині вікна (у вигляді ескізів сторінок, плитки, значків, списку або таблиці) і одночасно список підпапок розкривається в лівій частині вікна. Вид папок в правій частині вікна **Провідник** можна настроїти, наприклад, вибравши пункт *Вигляд* головного меню.

Ліва частина вікна **Провідник** може бути настроєна користувачем за допомогою пункту *Панель оглядача* з меню *Вигляд*. Наприклад, щоб в лівій частині вікна **Провідник** з'явився *Помічник з пошуку файлів*, користувачеві необхідно вибрати пункт меню *Вигляд*, потім в списку *Панелі оглядача*, що розкрився, вибрати підпункт *Пошук*.

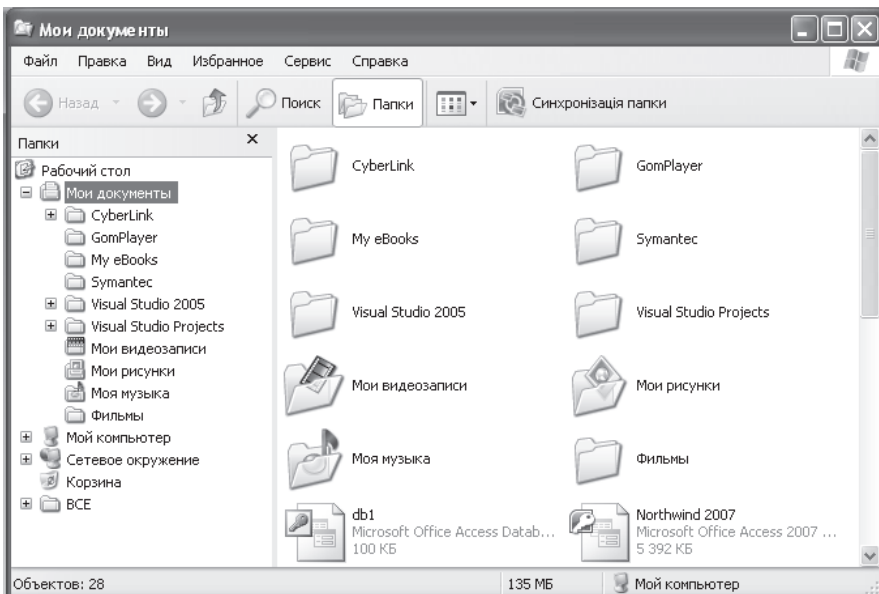


Рис.4.5.8. Вікно Провідник

Для переглядання графічних файлів за допомогою **Провідника** в Windows XP можна вибрати режим перегляду *Діафільм* в меню *Вигляд*, коли всі графічні файли розташовуються в одну низку і відображаються у вигляді мініатюр. У правій частині **Провідника** є можливість за допомогою спеціальних кнопок «перегорнути» малюнки (мініатюри), а також повертати зображення по або проти годинникової стрілки. Якщо виконати подвійне клацання миші по зображенню, то можна проглянути це зображення в збільшеному масштабі за допомогою спеціальної програми переглядання зображень і факсів. Програма переглядання зображень і факсів дозволяє не лише проглядати в збільшеному вигляді будь-які графічні файли і повертати зображення, але і здійснювати друк.

У Windows XP є можливість установки в лівій частині вікна Провідник так званої бічної панелі, що містить три підменю, які дозволяють управляти папкою і файлами, що знаходяться в даній папці. Для установки бічної панелі користувачеві слід прибрати всіх галочок біля підпунктів *Панелі оглядача* пункту меню *Вигляд*. Розглянемо функціональне призначення трьох підменю бічної панелі: *Завдання для файлів і папок*, *Інші місця*, *Детально*.

При виділенні папки в правій частині вікна **Провідник** підменю *Завдання для файлів і папок* містить наступні операції над папкою, які можна виконати одинарним клацанням лівої кнопки миші: *Перейменувати папки*, *Перемістити папку*, *Скопіювати папку*, *Опублікувати папку у веб*, *Відкрити спільний доступ до цієї папки*, *Відправити вміст цієї папки електронною поштою*, *Видалити папку*. При виборі будь-якого файлу в правій частині вікна **Провідник** в підменю *Завдання для файлів і папок* з'являється аналогічні операції над файлами: *Перейменувати файл*, *Перемістити файл*, *Копіювати файл*, *Опублікувати файл у веб*, *Відправити цей файл електронною поштою*, *Видалити файл*.

При відкритті папки *Мої малюнки* на бічній панелі з'являється підменю *Завдання для зображень*, що містить список типових завдань для файлів зображень і відповідних папок: *Проглянути як слайд-шоу*, *Замовлення знімків через інтернет*, *Друк зображень*. При відкритті папки *Моя музика* на бічній панелі з'являється підменю *Завдання для музики*, що містить список типових завдань для музичних файлів і папок: *Відтворити все*, *Пошук музики в Інтернеті* і тому подібне

Підменю *Інші місця* бічної панелі містить список папок, в які можна перейти після проглядання вмісту попередньої папки.

Підменю *Детально* бічної панелі містить інформацію про виділені в правій частині вікна **Провідник** файли або папки.

Питання і завдання для самостійної роботи

1. Що таке інтерфейс?
2. Що є апаратним інтерфейсом?
3. Що таке графічний інтерфейс?
4. Яка ідея реалізована в інтерфейсі оболонки WINDOWS?
5. Що є інтерфейсом NC?
6. Яка структура вікон системи WINDOWS?
7. З якого програмного середовища запускається WINDOWS?
8. Яка програма починає працювати після запуску WINDOWS?
9. Що таке піктограма?
10. Як перемістити на екрані вікно або піктограму?

11. Що означає «розгорнути» вікно і як це зробити?
12. Що означає «відновити» вікно і як це зробити?
13. Що означає «згорнути» вікно і як це зробити?
14. Як працювати з меню вікна?
15. Як вводиться команда в WINDOWS і його застосуваннях?
16. Чим відрізняється запуск команди без багатокрапки від команди з багатокрапкою?
17. Як ввести параметри команди WINDOWS?
18. Що треба зробити користувачеві, якщо він передумав використовувати команду?
19. Як змінити межі вікна?
20. Як дізнатися, не виходячи з застосування WINDOWS, правила роботи з цим застосуванням?
21. Як виконувати операції з папками і файлами за допомогою Провідника?
22. Що таке файл і як задати його ім'я?
23. Для чого використовується розширення в імені файлу?
24. Як задати звук для програмної події в операційній системі Windows XP?
25. Як задати тему робочого столу?
26. Які елементи робочого столу Ви знаєте?
27. Чим відрізняється Windows XP від інших версій сімейства Windows?
28. Як можна усунути годинник з панелі завдань?
29. Як налаштувати заставку екрана і встановити часовий інтервал заставки в операційній системі Windows XP?
30. Як управляти живленням монітору в ОС Windows XP?
31. Перерахуйте категорії Панелі управління та опишіть для чого вони призначені?

Список літератури

1. Андреев А.Г. Microsoft Windows XP: Home Edition і Professional. Русские версии / Под общ. ред. А.Н. Чекмарева. – СПб.: БХВ-Петербург, 2003. – 640 с.
2. Петроченков А.В. MS-DOS – не вопрос. – Смоленск: РИЦ «ТОК», 1993.
3. Пунчик Н.Н. Windows для начинающих и школьников. – Мн.: Тетрасистемс, 2003. – 160 с.
4. Таненбаум Э. Современные операционные системы. – 2-е изд. – СПб.: Питер, 2002. – 1040 с.
5. Турецкий В.Я. Математика и информатика. – 3-е изд., испр. и доп. – М.: ИНФРА-М, 2000. – 560 с.
6. Уильям Р. Станек Microsoft Windows XP Professional. Справочник администратора / Пер. с англ. – 2-е изд. – М.: Издательско-торговый дом «Русская редакция», 2003. – 448 с.
7. Фигурнов В.Э. IBM PC для пользователя. – М.: Финансы и статистика, 1994.

Розділ V. БАЗИ ДАНИХ

У цьому розділі наводяться основні відомості з теорії баз даних; розглядаються покоління систем управління базами даних; а також класифікація моделей даних.

5.1. Дані і обчислювальна техніка

Сприйняття реального світу можна співвіднести з послідовністю різних, хоча інколи і взаємопов'язаних явищ. З давніх часів люди намагалися описати ці явища (навіть тоді, коли не могли їх зрозуміти). Такий опис називають *даними*. Традиційно фіксація даних здійснюється за допомогою конкретного засобу спілкування, наприклад, за допомогою природної мови на конкретному носіїві (наприклад, папері). Зазвичай дані (факти, явища, події, ідеї або предмети) і їх інтерпретація (семантика) фіксуються спільно, оскільки природна мова достатньо гнучка для подання того і іншого.

Нерідко дані і інтерпретація розділені. Існують принаймні дві історичні причини, через які застосування ЕОМ привело до відділення даних від інтерпретації:

- 1) ЕОМ не володіли достатніми можливостями для обробки текстів на природній мові – основній мові інтерпретації даних;
- 2) вартість пам'яті ЕОМ була спочатку вельми велика.

Жорстка залежність між даними і програмами, що використовують їх, створює серйозні проблеми введення даних і робить використання їх менш гнучкими. Нерідкі випадки, коли користувачі однієї і тієї ж ЕОМ створюють і використовують в своїх програмах різні набори даних, що містять схожу інформацію.

Розробники прикладних програм (написаних, наприклад, на Бейсику, Паскалі або Сі) розміщують потрібні їм дані у файлах, організовуючи їх найбільш зручним для себе чином. При цьому одні і ті ж дані можуть мати в різних застосуваннях абсолютно різну організацію (різну послідовність розміщення в записі, різні формати одних і тих же полів і тому подібне). Узагальнити такі дані надзвичайно важко: наприклад, будь-яка зміна структури запису файлу, вироблювана одним з розробників, приводить до необхідності зміни іншими розробниками тих програм, які використовують записи цього файлу.

Активна діяльність по відшукуванню прийнятних способів узагальнення безперервно зростаючого об'єму інформації привела до створення на початку

60-х років спеціальних програмних комплексів, званих «Системи управління базами даних» (СУБД). Основна особливість СУБД – це наявність процедур для введення і зберігання не лише самих даних, але і описів їх структури.

➤ Файли, забезпечені описом даних, що в них зберігаються і знаходяться під управлінням СУБД, почали називати *банки даних*, а потім «*бази даних*» (БД).

5.2. Покоління СУБД

Прийнято виділяти три покоління СУБД:

I. Мережеві і ієрархічні системи БД, широко поширені в 70-і роки, отримали назву - системи БД першого покоління. Це були перші системи, що пропонували розвинену функціональність СУБД в рамках єдиної системи з мовами визначення і маніпулювання даними для набору записів. Назвемо деякі найбільш спільні характеристики ранніх систем:

1. Ці системи активно використовувалися протягом багатьох років, довше, ніж використовується яка-небудь з реляційних СУБД.

2. Всі ранні системи не ґрунтувалися на яких-небудь абстрактних моделях. Поняття моделі даних фактично увійшло до ужитку фахівців в області БД тільки разом з реляційним підходом. Абстрактні представлення ранніх систем з'явилися пізніше на основі аналізу і виявлення спільних ознак біля різних конкретних систем.

3. У ранніх системах доступ до БД проводився на рівні записів. Користувачі цих систем здійснювали явну навігацію в БД, використовуючи мови програмування, розширені функціями СУБД.

4. Навігаційна природа ранніх систем і доступ до даних на рівні записів примушували користувача самого проводити всю оптимізацію доступу до БД, без якої-небудь підтримки системи.

5. Після появи реляційних систем більшість ранніх систем були оснащені «реляційними» інтерфейсами. Проте в більшості випадків це не зробило їх посправжньому реляційними системами, оскільки залишалася можливість маніпулювати даними в природному для них режимі.

II. У 80-і роки системи першого покоління були істотно потіснені сучасним сімейством реляційних СУБД, званих системами БД другого покоління. Типові представники багатокористувальницьких професійних систем другого покоління – DB2, INGRES, ORACLE, Informix і ін.

У нашій країні уявлення про реляційні СУБД у більшості програмістів склалося на основі досвіду використання систем на платформі персональних комп'ютерів, таких як dBASE, FOXBASE, FoxPro, Paradox, Clipper, Clarion, а

пізніше Access. Причини такої популярності можна бачити як в широкому розповсюдженні персональних комп'ютерів, так і у відносній простоті і легкості вивчення і освоєння самих персональних СУБД. Дуже часто персональні СУБД використовувалися (та і зараз подекуди використовуються) для автоматизації таких завдань, наприклад, у фінансовій сфері, які вимагають багатокористувальницьких професійних систем.

Реляційні СУБД і зараз є найбільш популярними у сфері розробки бізнес-застосунків. Проте існує широкий клас застосунків, для яких технологія реляційних систем БД не є цілком задовільною – це технологія програмування; системи, засновані на знаннях і мультимедійні системи; системи автоматизації проектування (САПР); геоінформаційні системи (ГІС); видавничі системи; системи дистанційного навчання; системи електронної комерції та ін. Це перш за все пов'язано з примітивністю структур даних, що покладені в основу реляційної моделі даних. Плоскі нормалізовані відношення універсальні і теоретично достатні для представлення даних будь-якої наочної області. Проте в нетрадиційних застосуваннях в базі даних з'являються сотні, якщо не тисячі таблиць, над якими постійно виконуються дорогі операції з'єднання, необхідні для відтворення складних структур даних, властивих наочній області.

Іншим серйозним обмеженням реляційних систем є їх відносно слабкі можливості в частині представлення семантики застосування.

Усвідомлюючи ці обмеження і недоліки реляційних систем, дослідники в області баз даних виконують багаточисельні проекти, засновані на ідеях, що виходять за межі реляційної моделі даних

III. Термін «системи наступного (або третього) покоління» увійшов до життя після публікації групою відомих фахівців в області БД «Маніфесту систем баз даних третього покоління». В цілому можна сказати, що СУБД наступного покоління - це прями спадкоємці реляційних систем. У число вимог до СУБД третього покоління входять повнота системи типів, підтримуваних в СУБД; підтримка ієрархії і спадкоємства типів; можливість управління складними об'єктами і так далі.

Основними напрямками систем третього покоління, що володіють деякими різними характеристиками, є:

- орієнтація на розширену реляційну модель. Основні характеристики: максимальне дотримання (наскільки це можливо з урахуванням нових вимог) відомих принципів організації СУБД; абстрактні типи даних; підтримка історичної інформації і темпоральних запитів; відмова від вимог нормалізації. Однією з найбільш відомих СУБД цього напрямку є система Postgres. У Postgres реалізовано багато цікавих засобів: підтримується темпоральна

модель зберігання і доступу до даних; допускається зберігання в полях відношень даних абстрактних, визначуваних користувачами типів.

- генерація систем баз даних, орієнтованих на застосування. Основна характеристика: створення власне не системи, а генератора систем, якнайповніше відповідних потребам застосувань. Рішення досягається шляхом створення наборів модулів із стандартизованими інтерфейсами. Існують два експериментальні прототипи «генераційних» систем - Genesis і Exodus. Обидві ці системи засновані перш за все на принципах модульного і точного дотримання встановлених інтерфейсів. По суті справи, системи складаються з мінімального ядра (розвиненої файлової системи в разі Exodus) і технологічного механізму програмування додаткових модулів. У проєкті Exodus цей механізм ґрунтується на системі програмування E, яка є простим розширенням Сі++, що підтримує стабільне зберігання даних в зовнішній пам'яті. Замість готової СУБД надається набір «напівфабрикатів» з узгодженими інтерфейсами, з яких можна згенерувати систему, що максимально відповідає потребам додатка.

- системи баз даних, засновані на правилах. До цього напрямку належать дедуктивні БД. Основна характеристика: досягнення розширюваності системи та її можливість пристосовуватися до потреб конкретних застосувань шляхом використання стандартного механізму управління правилами. По суті справи, система є деяким інтерпретатором системи правил і набір модулів-дій, що викликаються відповідно до цих правил. Можна змінювати набори правил (існує спеціальна мова завдання правил) або змінювати дії, підставляючи інші модулі з тим же інтерфейсом. За визначенням, дедуктивна БД складається з двох частин: екстенціональної, що містить факти, і інтенціональної, що містить правила для логічного виведення нових фактів на основі екстенціональної частини і запиту користувача. Відзначимо лише, що зазвичай мови запитів і визначення інтенціональної частини БД є логічними (тому дедуктивні БД часто називають логічними). Є прямий зв'язок дедуктивних БД з базами знань.

- об'єктно-орієнтовані СУБД. Напрямок об'єктно-орієнтованих баз даних (ООБД) виник порівняно давно. Публікації з'являлися вже в середині 1980-х. Проте найбільш активно цей напрям розвивається останніми роками. У сьогоднішній день ведеться дуже багато експериментальних і виробничих робіт в області об'єктно-орієнтованих СУБД. На ринку представлено понад 25 систем ООБД. Серед них — система GemStone компанії Servio, ONTOS компанії Ontos, ObjectStore компанії Object Design, O2, ORION, Iris. Крім того, системи управління реляційними базами даних, розроблені компаніями Oracle, Microsoft, Borland, Informix, включали об'єктно-орієнтовані засоби. Багато які з цих продуктів з'явилися ще в другій половині 80-х років, і сьогодні, після

майже півтора десятиліть розробки вони все ще не вступили в пору зрілості; у цьому — одна з причин того, що до цього дня світовий ринок реальних застосувань не поспішає приймати системи ООБД.

Широке використання Інтернет в різних сферах діяльності ставить перед розробниками СУБД наступні проблеми:

1. Інтеграція тексту, даних, коду і потоків. В області СУБД основна увага завжди приділялася організації, зберіганню, аналізу і вибірці структурованих даних. Розвиток Web продемонстрував важливість складніших типів даних: текстів, зображень, темпоральних, аудио- і відеоданих.

2. Інтеграція інформації. Типовим підходом до інтеграції інформації в масштабах підприємства є побудова сховищ (DataWarehouse) вітрин (data mart) даних на основі витягання операційних даних, їх трансформації до єдиної схеми і завантаження даних в сховищі (процедура ETL - extraction, transformation, loading). Цей підхід придатний для використання на підприємстві з декількома десятками операційних баз даних, що знаходяться під єдиним контролем. У Інтернет потрібно проводити інтеграцію інформації між декількома підприємствами. Як правило, організації не дозволять в масовому порядку витягувати дані зі своїх операційних баз даних, до них можна буде адресувати лише одиночні запити. В результаті буде потрібно здійснювати інтеграцію, можливо, мільйонів інформаційних джерел «на льоту». У зв'язку з цим існує безліч невіршених проблем: семантична неоднорідність; неповнота і неточність даних; обмеженість доступу до конфіденційних даних і так далі.

3. Мультимедійні запити. Очевидно, що об'єм мультимедійних даних (зображення, аудіо, відео і так далі) значно зростає. Проблемою співтовариства баз даних є створення простих способів аналізу, узагальнення, пошуку і огляду електронних підбірок мультимедійної інформації, що належать до деякого об'єкту.

5.3. Термінологія в СУБД

У загальногалузевих керівних матеріалах по створенню банків даних Державного комітету з науки й техніки, виданих в 1982 р., наводяться наступні визначення основних понять:

➤ **База даних (БД)** - іменована сукупність даних, що відображає стан об'єктів та їх відношень в даній предметній області (ПО).

➤ **Предметна область (ПО)** – частина реального світу, що підлягає автоматизації з метою організації управління. Вона представлена

безліччю фрагментів, кожен з яких характеризується об'єктами, процесами і безліччю користувачів.

➤ **Банк даних (БНД)** – це система спеціальним чином організованих даних – баз даних, програмних, технічних, мовних, організаційно-методичних засобів, призначених для забезпечення централізованого накопичення і колективного багатоцільового використання даних.

➤ **Системи управління базами даних (СУБД)** – сукупність мовних і програмних засобів, призначених для створення, ведення і спільного використання БД багатьма користувачами.

Програми, за допомогою яких користувачі працюють з БД, називаються *додатками*.

СУБД повинна забезпечувати:

- можливість представлення внутрішньої структури даних;
- фізичну і логічну незалежність даних;
- мінімальну надмірність даних;
- можливість швидкого пошуку;
- ефективні мови запитів до даних;
- вимоги безпеки, надійності, конфіденційності, цілісності;
- дані мають бути захищені від перекручення, розкрадання, руйнування;
- дані мають бути відновлюваними;
- дані мають бути контрольованими;
- має бути встановлена процедура ідентифікації користувачів;
- має бути організована система санкціонованого доступу;
- має бути встановлений контроль за діями користувача з метою виявлення помилкових операцій.

СУБД повинна надавати доступ до даних будь-яким користувачам, включаючи і тих, які практично не мають і (або) не хочуть мати уявлення про:

- фізичне розміщення в пам'яті даних і їх описів;
- механізми пошуку запрошуваних даних;
- проблеми, що виникають при одночасному запиті одних і тих же даних багатьма користувачами (прикладними програмами);
- способи забезпечення захисту даних від некоректних оновлень і (або) несанкціонованого доступу;
- підтримку БД в актуальному стані і безліч інших функцій СУБД.

При виконанні основних з цих функцій СУБД повинна використовувати різні описи даних. Природно, що проект бази даних треба починати з аналізу предметної області і виявлення вимог до неї окремих

користувачів (співробітників організації, для яких створюється база даних). Докладніше цей процес розглянемо нижче, а тут відзначимо, що проектування зазвичай доручається людині (групі осіб) – адміністраторові бази даних (АБД). Ним може бути як спеціально виділений співробітник організації, так і майбутній користувач БД, досить добре знайомий з машинною обробкою даних.

➤ Об'єднуючи приватні уявлення про вміст БД, отримані в результаті опиту користувачів, і свої уявлення про дані, які можуть бути потрібні в майбутніх застосуваннях, АБД спочатку створює узагальнений неформальний опис створюваної бази даних. Цей опис, виконаний з використанням природної мови, математичних формул, таблиць, графіків і інших засобів, зрозумілих всім людям, працюючим над проектуванням бази даних, називають *інфологічною моделлю даних*.

Така людино-орієнтована модель повністю незалежна від фізичних параметрів середовища зберігання даних. Тому інфологічна модель не повинна змінюватися до тих пір, поки якісь зміни в реальному світі не будуть вимагати зміни в ній деякого визначення, щоб ця модель продовжувала відображати предметну область. Решта моделей є комп'ютерно-орієнтованими. З їх допомогою СУБД дає можливість програмам і користувачам здійснювати доступ до даних, що зберігаються, лише по їх іменах, не піклуючись про фізичне розташування цих даних. Потрібні дані відшуковуються СУБД на зовнішніх запам'ятовуючих пристроях по фізичній моделі даних.

➤ Оскільки вказаний доступ здійснюється за допомогою конкретної СУБД, то моделі мають бути описані на мові опису даних цієї СУБД. Такий опис, що створюється АБД за інфологічною моделлю даних, називають *даталогічною* або *концептуальною моделлю даних*.

Трирівнева архітектура (інфологічний, даталогічний і фізичний рівні) дозволяє забезпечити незалежність даних, що зберігаються, від програм, що використовують їх. За необхідності АБД може переписати дані, що зберігаються, на інші носії інформації і (або) реорганізувати їх фізичну структуру, змінивши лише фізичну модель даних. АБД може підключити до системи будь-яке число нових користувачів (нових застосувань), доповнивши, якщо треба, даталогічну модель. Вказані зміни фізичної і даталогічної моделі не будуть відзначені існуючими користувачами системи (виявляться «прозорими» для них), так само як не будуть відзначені і нові користувачі. Отже, незалежність даних забезпечує можливість розвитку системи баз даних без руйнування існуючих застосувань.

5.4. Класифікація моделей даних

➤ *Модель даних* – це сукупність структур даних, взаємозв'язків і операцій їх обробки.

На рисунку 5.4.1 представлена класифікація моделей даних відповідно до розглянутої раніше трирівневої архітектури.

Інфологічна модель відображає реальний світ в деякі зрозумілі людині концепції, повністю незалежні від параметрів середовища зберігання даних. Інфологічна модель відображає ПО у вигляді сукупності інформаційних об'єктів і їх структурних зв'язків. Існує безліч підходів до побудови таких моделей. Найбільш популярною з них виявилася *модель «суть-зв'язок»* (ER - Entity Relationship) або *ER-модель*. Інфологічна модель має відображувати в комп'ютерно-орієнтовану даталогічну модель, «зрозумілу» СУБД. В процесі розвитку теорії і практичного використання баз даних, а також засобів обчислювальної техніки створювалися СУБД, що підтримують різні даталогічні моделі – документальні і фактографічні.

➤ *Документальні моделі* відповідають уявленню про слабо структуровану інформацію, орієнтовану на вільні формати тексту на природній мові.

Моделі, орієнтовані на формат документа, пов'язані перш за все із стандартною спільною мовою розмітки - SCML, частіше використовуються HTML і XML

➤ *Тезаурусні моделі* засновані на принципі організації словників і містять певні мовні конструкції і принципи їх взаємодії в заданій граматиці. Ці моделі використовуються в системах – перекладачах, особливо в багатоязикових перекладачах.

➤ *Дескрипторні моделі* – найпростіші з документальних моделей і широко використовувалися на ранніх стадіях використання документальних баз даних. У цих моделях кожному документу відповідає дескриптор (описувач), що має жорстку структуру. Дескриптор описував документ відповідно до тих характеристик, які потрібні для роботи з документами в БД, що розробляється. Наприклад, для БД з описами патентів дескриптор містив назву області, до якої відносився патент, номер, дату видачі патенту і ще низку ключових характеристик. Опрацювання інформації в таких БД велася виключно по дескрипторам, а не по самому патенту.



Рис. 5.4.1. Класифікація моделей даних

- *Фізична модель даних* оперує категоріями, що стосуються організації зовнішньої пам'яті і структур зберігання, використовуваних в даному операційному середовищі. Фізична організація даних робить основний вплив на експлуатаційні характеристики БД.
- *Фактографічні моделі.* Спочатку почали використовувати ієрархічні даталогічні моделі. Простота організації, наявність заздалегідь заданих зв'язків між суттю, схожість з фізичними моделями даних дозволяли добиватися прийнятної продуктивності ієрархічних СУБД на повільних ЕОМ з велими обмеженими об'ємами пам'яті. Але, якщо дані не мали деревовидної структури, то виникало багато складнощів при побудові ієрархічної моделі.
- *Мережеві моделі* також створювалися для мало ресурсних ЕОМ. Це достатньо складні структури, що складаються з «наборів» – поименованих дворівневих дерев. «Набори» з'єднуються за допомогою «записів-зв'язок», утворюючи ланцюжки і так далі. При розробці мережевих моделей було вигадано безліч «маленьких хитрощів», що дозволяють збільшити продуктивність СУБД, але істотно ускладнили останні. Прикладний програміст повинен вивчити декілька внутрішніх мов СУБД, детально представляти логічну структуру бази даних для здійснення навігації серед різних екземплярів, наборів, записів і тому подібне.

Складність практичного використання ієрархічних і мережевих СУБД примушувала шукати інші способи представлення даних. В кінці 60-х років з'явилися СУБД на основі інвертованих файлів, що відрізняються простотою

організації й наявністю вельми зручних мов маніпулювання даними. Проте такі СУБД володіють низкою обмежень на кількість файлів для зберігання даних, кількість зв'язків між ними, довжину запису і кількість її полів.

Сьогодні найбільш поширені реляційні моделі, які детально будуть розглянуті в пункті 5.9.

Будь-яка логічна модель даних повинна містити три компоненти:

- структура даних;
- набір допустимих операцій, що виконуються на структурі даних. Модель даних передбачає наявність мови визначення даних, що описує структуру їх зберігання, і мови маніпулювання даними, що включає операції витягання і модифікації даних;
- обмеження цілісності: механізм підтримки відповідності даних ПО на основі формальних описаних правил.

Перш ніж перейти до детального і послідовного вивчення реляційних систем БД, зупинимось коротенько на логічних моделях ранніх (дореляційних) СУБД.

Для позначення типів структур даних широко використовується термінологія, запропонована CODASYL (The Conference on Data Systems Languages): елемент даних, агрегат даних, запис, база даних.

➤ *Елемент даних* (ЕД) – найменша поійменована одиниця даних, до якої СУБД може адресуватися безпосередньо, і за допомогою якої виконується побудова решти всіх структур. ЕД має ім'я і значення.

➤ *Агрегат даних* (АД) – поійменована сукупність ЕД усередині запису, який можна розглядувати як єдине ціле. АД може бути простим або складеним.

➤ *Запис* – поійменована сукупність ЕД і агрегатів. Запис – це АД, що не входить до складу іншого агрегату даних. Запис може мати складну ієрархічну структуру, оскільки допускається багатократне застосування агрегації.

➤ *База даних* – поійменована сукупність екземплярів записів різного типу, що містить застосування між записами, представлені екземплярами наборів. Опис структури БД задається її схемою.

5.5. Системи, засновані на інвертованих списках

До найбільш відомих і типових представників систем, заснованих на інвертованих списках, належать Datacom/DB компанії Applied Data Research, Inc. (ADR), орієнтована на комп'ютери фірми IBM, і Adabas компанії Software AG.

Організація доступу до даних на основі інвертованих списків використовується практично у всіх сучасних реляційних СУБД, але в цих системах користувачі не мають безпосереднього доступу до інвертованих списків (індексів).

Структури даних

База даних, організована за допомогою інвертованих списків, схожа на реляційну БД, але з тією відзнакою, що таблиці, що зберігаються, і шляхи доступу до них видно користувачам. При цьому:

- рядки таблиць впорядковані системою в деякій фізичній послідовності;
- фізична впорядкованість рядків всіх таблиць може визначатися і для всієї БД (так робиться, наприклад, в Database/DB);
- для кожної таблиці можна визначити довільне число ключів пошуку, для яких будуються індекси. Ці індекси автоматично підтримуються системою, але явно видні користувачам.

Маніпулювання даними

Підтримуються два класи операторів:

1. оператори, що встановлюють адресу записи, серед яких:
 - прями пошукові оператори (наприклад, знайти перший запис таблиці по деякому шляху доступу);
 - оператори, що знаходять запис в термінах відносної позиції від попереднього запису по деякому шляху доступу.
2. оператори над записами, що адресуються

Приклади операторів:

- LOCATE FIRST - знайти перший запис таблиці T у фізичному порядку; повертає адресу запису;
- LOCATE FIRST WITH SEARCH KEY EQUAL - знайти перший запис таблиці T із заданим значенням ключа пошуку K; повертає адресу запису;
- LOCATE NEXT - знайти перший запис, наступний за записом із заданою адресою в заданому шляху доступу; повертає адресу запису;
- RETRIVE - вибрати запис з вказаною адресою;
- UPDATE - відновити запис з вказаною адресою;
- DELETE - видалити запис з вказаною адресою;
- STORE - включити запис у вказану таблицю; операція генерує адресу запису.

Обмеження цілісності

Спільні правила визначення цілісності БД відсутні. У деяких системах підтримуються обмеження унікальності значень деяких полів, але в основному все покладається на прикладну програму.

5.6. Ієрархічні моделі

Типовим представником (найбільш відомим і поширеним) є Information Management System (IMS) фірми IBM. Перша версія з'явилася в 1968 році.

Ієрархічні структури даних

Ієрархічна модель даних (ІМД) властива багатьом реальним деревовидним структурам (класифікатори, структури управління і т. і.). Існують графічна і таблична форми представлення даних ІМД. Ієрархічна БД складається з впорядкованого набору дерев; точніше, з впорядкованого набору декількох екземплярів одного типу дерева.

➤ Дерево - зв'язний неорієнтований граф, який не містить циклів. Ієрархічна деревовидна структура, орієнтована від кореня (виділена вершина або вузол) і визначається умовами:

- ієрархія починається з кореневого вузла, який знаходиться на першому рівні ієрархії;
- на нижніх рівнях знаходяться породжені (залежні) вузли;
- кожен породжений вузол, що знаходиться на i -тому рівні, пов'язаний безпосередньо з одним початковим (батьківським) вузлом, що знаходиться на $i-1$ рівні ієрархії;
- кожен початковий вузол може мати один або декілька безпосередньо породжених вузлів;
- доступ до породженого вузла здійснюється через його початковий вузол;
- існує єдиний ієрархічний шлях доступу до вузла, починаючи від кореня дерева (рисунок 5.6.1).

Ієрархічний шлях включає всі зв'язані між собою вузли.

У ІМД використовується орієнтація деревовидної структури від кореня до листя дерева.

Приклад схеми ієрархічної БД представлений на рисунку 5.6.2.

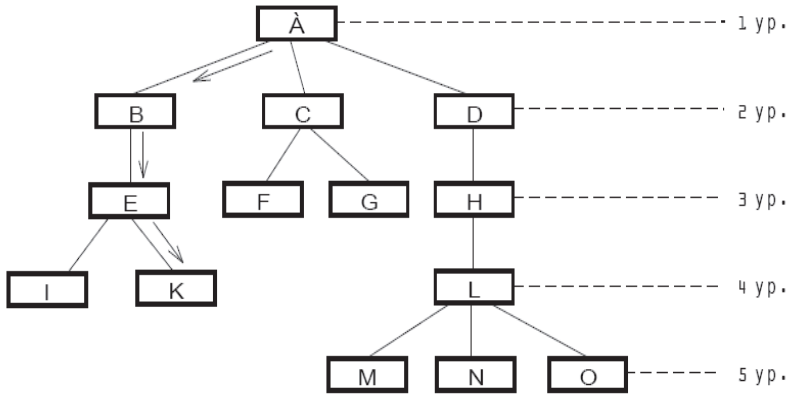


Рис.5.6.1. Приклад ієрархічної структури

Для БД визначений повний порядок обходу - зверху-вниз, зліва-направо.

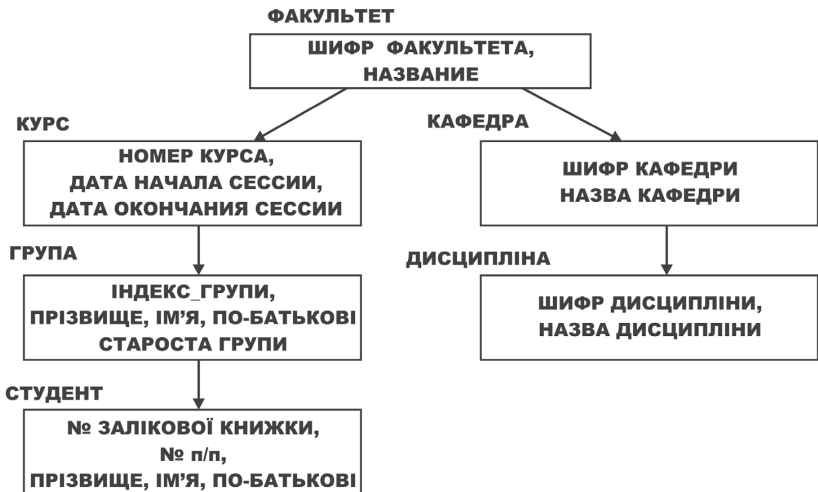


Рис. 5.6.2. Ієрархічна БД

Маніпулювання даними

Прикладами типових операторів маніпулювання ієрархічно організованими даними можуть бути наступні:

- знайти вказане дерево БД;
- перейти від одного дерева до іншого;
- перейти від одного запису до іншого усередині дерева (наприклад, від групи - до першого студента);
- перейти від одного запису до іншого в порядку обходу ієрархії;
- вставити новий запис у вказану позицію;
- видалити поточний запис.

Обмеження цілісності

Автоматично підтримується цілісність записів між предками і нащадками. Основне правило: ніякий нащадок не може існувати без свого батька.

5.7. Мережеві моделі

Типовим представником є Integrated Database Management System (IDMS) компанії Cullinet Software, Inc. Архітектура системи заснована на пропозиціях Data Base Task Group (DBTG) Комітету з мов програмування (Conference on Data Systems Languages - CODASYL). Звіт DBTG був опублікований в 1971 році, а в 70-х роках з'явилося декілька систем, серед яких IDMS.

Мережеві структури даних

Мережевий підхід до організації даних є розширенням ієрархічного. У ієрархічних структурах запис-нащадок повинен мати в точності одного предка; у мережевій структурі даних нащадок може мати будь-яке число предків.

Мережева БД складається з набору записів, відповідних кожному екземпляру об'єкту предметної області, і набору зв'язків між цими записами.

Простий приклад мережевої схеми БД наведений на рисунку 5.7.1. Для мережевих моделей допускається перетини, цикли. В деяких випадках один елемент даних може бути пов'язаний з цілою сукупністю інших елементів даних. Наприклад, один виріб може поставлятися декількома постачальниками, кожен з яких встановив свою ціну. Елемент даних ЦІНА не може бути асоційований тільки із записом ВИРІБ або тільки із записом ПОСТАЧАЛЬНИК, а має бути пов'язаний з двома цими записами. Дані, що асоціюються з сукупністю записів, називають *даними перетини*.

Циклом називається ситуація, в якій початковий вузол є в той же час породженим вузлом.

Будь-яку мережеву модель можна представити у вигляді ієрархічної шляхом введення надмірності.

Мережа перетвориться в дерево вказівкою деяких вузлів двічі.



Рис.5.7.1. Приклад мережевої схеми БД

Маніпулювання даними

Зразковий набір операцій може бути наступним:

- знайти конкретний запис в наборі однотипних записів (інженера Сидорова);
- перейти від предка до першого нащадка по деякому зв'язку (до першого співробітника відділу 310);
- перейти до наступного нащадка в деякому зв'язку (від Сидорова до Іванова);
- перейти від нащадка до предка по деякому зв'язку (знайти відділ Сидорова);
- створити новий запис;
- знищити запис;
- модифікувати запис;
- включити в зв'язок;
- виключити із зв'язку;
- переставити в інший зв'язок і так далі

Обмеження цілісності

В принципі їх підтримка не потрібна, але інколи вимагають цілісності по застосуваннях (як в ієрархічній моделі).

Достоїнства ранніх СУБД:

- розвинені засоби управління даними в зовнішній пам'яті на низькому рівні;
- можливість побудови уручну ефективних прикладних систем;
- можливість економії пам'яті за рахунок розділення підоб'єктів (у мережевих системах).

Недоліки дореляційних СУБД:

- дуже складно користуватися;
- фактично необхідні знання про фізичну організацію даних;

- прикладні програми залежать від цієї організації;
- логіка прикладних програм переобтяжена деталями організації доступу до БД.

5.8. Фізичні моделі організації баз даних

Фізичні моделі визначають спосіб розміщення даних в середовищі зберігання і способи доступу до цих даних, які підтримуються на фізичному рівні. Серед найважливіших характеристик будь-якої бази даних слід назвати продуктивність, надійність і простоту адміністрування. Знання того, як більшість СУБД фізично зберігають дані в зовнішній пам'яті, уявлення про параметри цього зберігання і відповідні методи доступу можуть дуже допомогти при проектуванні баз даних, що володіють заданою продуктивністю. Будь-яка логічна структура даних представляється на фізичному рівні у вигляді послідовності бітів.

Можна виділити наступні аспекти проблеми фізичного представлення даних:

I. Як знайти потрібний запис? Необхідно встановити відповідність між логічним записом і адресою фізичного запису. Під фізичним записом розумітимемо послідовність бітів, які можна прочитати за допомогою однієї машинної інструкції. Логічні записи знаходять по ключу або сукупності ключів.

II. Яким чином організувати дані, щоб їх пошук був ефективним, а вибірку можна було здійснити по сукупності ключів?

III. Як можна додати новий запис до даних, знищити старі записи і при цьому не порушити системи адресації і пошуку, а також самі структури даних.

Вкажемо основні чинники, що впливають на фізичну організацію даних для конкретних БД.

1. Довільна або послідовна обробка даних. Для визначення виду обробки використовують коефіцієнт активності файлу (k)

$$k = z1/z$$

де $z1$ – число записів, лічених за 1 прогін; z - число записів, проглянутих за 1 прогін.

Якщо коефіцієнт k великий, то використовують послідовну обробку, наприклад, при розрахунку заробітної плати.

2. Частота звернення до певних записів.

3. Час відповіді (поважно для систем реального часу).

4. Здібність до розширення (особливо, якщо додається записів більше, ніж знищується).

5. Можливість організації пошуку по декількох ключах.

Можна виділити наступні способи адресації (пошуку потрібного запису):

1. Послідовне сканування файлу з перевіркою ключа кожного запису. Такий метод використовується, якщо вибраний послідовний метод обробки даних або використовується файл послідовного доступу. Вимагає багато часу.

2. Блоковий пошук. Якщо записи впорядковані по ключу, то при скануванні не потрібне читання кожного запису. Прочитується перший запис блоку і її ключ порівнюється з ключем шуканого запису. А далі або є видимими всі записи даного блоку або вибирається перший запис наступного блоку.

3. Перетворення ключа в адресу - найшвидша організація пошуку. Зараз застосовується технологія хешування – технологія швидкого доступу до запису, що зберігається, на основі обчислення спеціальної функції від заданого значення деякого поля. Це значення і є адресою для запису.

4. Пошук по індексу. Первинний індекс – індекс, що використовує як вхідну інформацію первинний ключ. У індексному файлі запис складається з індексу і покажчика. Спочатку проводиться пошук в індексі, а потім по покажчику звертаємося до основного файлу із записами. Ефективно, швидко, але потрібна пам'ять для зберігання індексу.

5. Бінарний (двійковий) пошук для записів, впорядкованих по ключу.

6. Пошук по В-дереву.

Історично першими системами зберігання і доступу були файлові структури і системи управління файлами (СУФ), які фактично були частиною операційних систем. СУБД створювала над цими файлами свою надбудову, яка дозволяла організувати всю сукупність файлів так, щоб вона працювала як єдине ціле і отримувала централізоване управління від СУБД. При цьому безпосередній доступ здійснювався на рівні файлових команд, які СУБД використовувала при маніпулюванні файлами.

Проте механізми буферизації і управління файловими структурами не пристосовані для вирішення завдань власне СУБД, оскільки створювалися для традиційної обробки файлів, і із зростанням об'ємів даних, що зберігаються, вони стали неефективними для використання СУБД. Тоді поступово стався перехід від базових файлових структур до безпосереднього управління розміщенням даних на зовнішніх носіях самої СУБД. При цьому механізми, вживані у файлових системах, перейшли багато в чому і в нові системи організації даних в зовнішній пам'яті, звані частіше сторінковими системами зберігання інформації. Будь-яке впорядковане розташування

даних на диску, називається структурою зберігання. На рисунку 5.8.1 приведена класифікація структур зберігання інформації в БД.

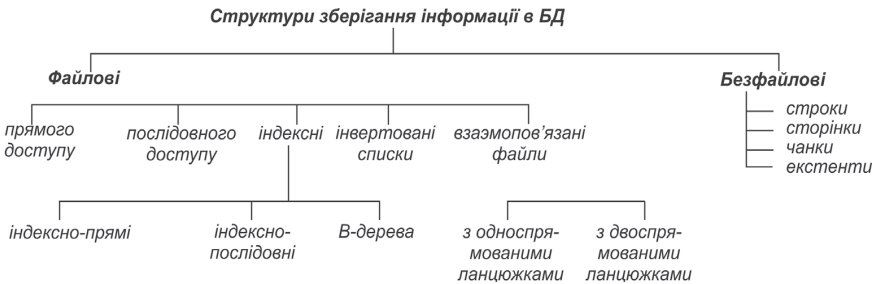


Рис. 5.8.1. Класифікація структур зберігання інформації в БД

Файлові структури, використовувані для зберігання даних в БД

На пристроях послідовного доступу (магнітофони, стримери) можуть бути організовані файли тільки послідовного доступу. Файли із змінною довжиною запису також завжди є файлами послідовного доступу і можуть бути організовані двома способами:

- кінець запису наголошується спеціальним маркером;
- на початку кожного запису записується її довжина.

Файли з постійною довжиною запису, розташовані на пристроях прямого доступу (магнітні, оптичні диски), є файлами прямого доступу. У цих файлах фізична адреса розташування потрібного запису може бути обчислена по номеру запису (NZ).

Для файлів з постійною довжиною запису адреса розміщення запису з номером K може бути обчислена за формулою:

$$VA+(K-1)*LZ+1,$$

де VA – базова адреса, LZ – довжина запису.

Файли прямого доступу забезпечують найбільш швидкий доступ до довільних записів, і їх використання є найбільш перспективним в системах БД.

Проте найчастіше в БД необхідний пошук по ключах, а не по номеру запису, і номер запису, необхідний для прямого доступу, в цьому випадку невідомий. При організації файлів прямого доступу в деяких випадках можлива побудова функції, яка за значенням ключа K однозначно обчислює номер запису (номер запису файлу) $NZ=F(K)$.

Часто не вдається побудувати взаємно-однозначну відповідність між значеннями ключа і номерами записів, тому застосовують різні методи хешування, створюють спеціальні хеш-функції.

Спільною ідеєю методів хешування є застосування до значення ключа деякої функції згортки (хеш-функції), що виробляє значення меншого розміру. Згортка значення ключа потім використовується для доступу до запису.

У найпростішому, класичному випадку, згортка ключа використовується як адреса в таблиці, що містить ключі і записи. Основною вимогою до хеш-функції є рівномірний розподіл значення згортки. При виникненні колізій (одна і та ж згортка для декількох значень ключа) утворюються ланцюжки переповнювання. Новий запис заноситься в область переповнювання на перше вільне місце, а в записі-синонімі (з тим же значенням хеш-функції), який знаходиться в основній області, робиться засування на адресу знов розміщеного запису в області переповнювання. Головним обмеженням цього методу є фіксований розмір таблиці. Якщо таблиця заповнена дуже сильно або переповнена, але виникнуть дуже багато ланцюжків переповнювання, і головну перевагу хешування - доступ до запису майже завжди за одне звернення до таблиці - буде втрачено. Розширення таблиці вимагає її повної переробки на основі нової хеш-функції (із значенням згортки більшого розміру).

В разі баз даних такі дії є абсолютно неприйнятними. Тому зазвичай вводять проміжні таблиці-довідники, що містять значення ключів і адреси записів, а самі записи зберігаються окремо. Тоді при переповненні довідника потрібна тільки його переробка, що значно простіше.

Не дивлячись на високу ефективність хеш-адресації, у файлових структурах не завжди вдається знайти відповідну функцію, тому при організації доступу по первинному ключу широко використовуються індексні файли. Індексні файли можна представити як таблиці покажчиків до основного файлу із записами, причому для одного основного файлу можна побудувати декілька індексних по різних ключах.

Розрізняють два типи індексних файлів: з щільним індексом (індексно-прямі) і з нещільним індексом (індексно-послідовні).

Структура файлів з щільним індексом має вигляд:

Значення ключа *Номер запису в основному файлі*

Записи в основному файлі розташовані в довільному порядку. Такі файли будуються для первинних ключів і в них не може бути двох записів, що мають однакові значення первинного ключа. Всі записи в індексному файлі впорядковані за значенням ключа, тому для пошуку в індексному файлі можна застосувати бінарний або двійковий пошук.

Файли з нещільним індексом будуються для основних файлів, в яких записи впорядковані по ключу і структура індексних файлів має вигляд:

Значення ключа першого запису блоку Номер блоку з цим записом в основному файлі

У індексному файлі шукається потрібний блок по заданому значенню первинного ключа. Оскільки всі записи впорядковані, то значення першого запису блоку дозволяє швидко визначити, в якому блоці знаходиться шуканий запис. Решта всіх дій з пошуку минає в основному файлі.

Найбільш популярним підходом до організації індексів в базах даних є використання техніки В-дерев. З погляду зовнішнього логічного уявлення В-дерево - це збалансоване дерево в зовнішній пам'яті. Збалансованість означає, що довжина шляху від кореня дерева до будь-якого його листа одна і та ж. Побудова В-дерев пов'язана з простою ідеєю побудови індексу над вже побудованим індексом. Якщо побудувати файл з нещільним індексом, то, розглядаючи його як основний файл, над яким треба знову побудувати файл з нещільним індексом, а потім знову над новим індексом будемо наступний і так до того моменту, поки не залишиться всього один індексний блок.

Якщо індексні файли використовуються для прискорення доступу по первинному ключу, то для прискорення доступу по вторинному ключу використовуються структури, звані інвертованими списками. Вторинними ключами є атрибут або набір атрибутів, якому відповідає декілька шуканих записів. Наприклад, для таблиці «Книги» вторинним ключем може служити місце видання, рік видання. Безліч книг можуть бути видані в одному місці, і безліч книг можуть бути видані в одному році.

Інвертований список в спільному випадку – це трирівнева індексна структура. На першому рівні знаходиться файл або частина файлу, в якій впорядковано розташовані значення вторинних ключів. Кожен запис з вторинним ключем має засылання на номер першого блоку в ланцюжку блоків, що містять номери записів з даним значенням вторинного ключа. На другому рівні знаходиться ланцюжок блоків, що містять номери записів з одним і тим же значенням вторинного ключа. При цьому блоки другого рівня впорядковані по значенням вторинного ключа. На третьому рівні знаходиться основний файл із записами. Представимо механізм доступу до записів по вторинному ключу:

Шаг_1. В області першого рівня шукається задане значення вторинного ключа;

Шаг_2. По засыланню прочитуються блоки другого рівня, що містять номери записів із заданим значенням вторинного ключа;

Шаг_3. У робочу область користувача прямим доступом завантажується вміст всіх записів із заданим значенням вторинного ключа.

Для одного основного файлу може бути створене декілька інвертованих списків по різних вторинних ключах. Проте при модифікації основного файлу потрібно внести зміни до всіх інвертованих списків. Тому можна стверджувати, що побудову інвертованих списків прискорює процес доступу тільки в тому випадку, якщо БД стабільна і її вміст не змінюється.

Для моделювання зв'язків на файлових структурах використовується принцип організації ланцюжків записів усередині файлу і застосування номерів записів для декількох взаємозв'язаних файлів.

Ланцюжок – це сукупність записів, розташованих в різних місцях і зв'язаних послідовністю покажчиків. Структура файлу з ланцюжком може бути умовно представлена у вигляді:

Ключ Запис Заслання-покажчик на наступний запис

Для моделювання відношення один-до-багатьох зв'язуються два файли, наприклад F1 і F2, причому передбачається, що один запис у файлі F1 може бути пов'язана з декількома записами у файлі F2. Структура файлу F1 може бути умовно представлена:

Ключ Запис Заслання-покажчик на перший запис у файлі F2, з якою починається ланцюжок записів файлу, пов'язаних з даним записом файлу F1

Структура запису файлу F2 має вигляд:

Покажчик на наступний запис в ланцюжку Вміст запису

Моделі сторінкової організації даних в сучасних БД

Реляційні СУБД зберігають наступні різновиди об'єктів в зовнішній пам'яті БД:

- рядки таблиць - основна частина БД;
- структури, що управляють, - індекси, що створюються за ініціативою користувача (адміністратора) з міркувань підвищення ефективності виконання запитів;
- журнальна інформація, підтримувана для задоволення потреби в надійному зберіганні даних;
- службова інформація, підтримувана для задоволення внутрішніх потреб нижнього рівня системи (наприклад, інформація про вільну пам'ять).

Зберігання даних в зовнішній пам'яті у відомих СУБД (Oracle, IBM DB2, Microsoft SQL Server, Sybase і Informix та ін.) організоване дуже схожим чином. Основними одиницями фізичного зберігання є блок даних, екстент, чанк. Логічний рівень представлення інформації включає простори (або табличні простори). Блок даних (block) або сторінка (page) є одиницею обміну із зовнішньою пам'яттю. Розмір сторінки фіксований для бази даних (Oracle) або встановлюється при створенні.

Розмір блоку надає великий вплив на продуктивність бази даних — при великих розмірах швидкість операцій читання/запису росте (особливо це

характерно для повних переглядів таблиць і операцій інтенсивного завантаження даних), проте зростають накладні витрати на зберігання (база збільшується) і знижується ефективність індексних переглядів. Менший розмір блоку дозволяє економніше витрачати пам'ять, але разом з тим відносно дорогий. Довгі блоки (16, 32 або 64 Кбайт) краще використовувати для великих об'єктів даних: повнотекстові фрагменти, мультимедіа-об'єкти, довгі рядки і тому подібне. Короткі блоки (2 або 4 Кбайт) краще лічать для значень числових типів, недовгих рядків, значень дати і часу. Слід також враховувати розмір блоку ОС, він має бути кратний розміру блоку бази даних.

Простором зовнішньої пам'яті, відведеним адміністратором, СУБД управляє за допомогою екстентів (extent), тобто безперервних послідовностей блоків (сторінок). Інформація про наявність екстентів для об'єкту схеми даних знаходиться в спеціальних структурах, що управляють, реалізація яких залежить від СУБД. На управління екстентами (виділення простору, звільнення, злиття) витрачаються певні ресурси, тому для досягнення ефективності потрібно правильно визначати їх параметри. СУБД від Oracle, IBM, Informix дозволяють визначати параметри цих структур, а в Sybase екстенти мають постійний розмір, рівний 8 сторінкам. Зменшення розміру екстента сприятиме ефективнішому використанню пам'яті, проте при цьому зростають накладні витрати на управління великою кількістю екстентов, що може уповільнити операції вставки великої кількості рядків в таблицю. У Informix існує ще одна одиниця фізичного зберігання, проміжна між файлом (або розділом диска) і екстентом, — це «чанк» (від англійського chunk, що дослівно перекладається як «ємність»). Чанк дозволяє гнучкіше управляти дуже великими масивами зовнішньої пам'яті. У одному розділі диска або файлі адміністратор може створити декілька чанков.

Основною одиницею здійснення обміну даних є сторінка даних. Всі дані зберігаються посторінково. При табличному зберіганні дані на одній сторінці є однорідними, тобто Сторінка може містити тільки дані або тільки індекси. Всі сторінки даних мають однакову структуру, представлену на рисунку 5.8.2.

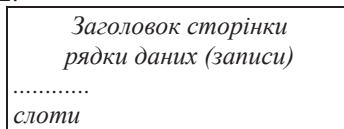


Рис.5.8.2. Структура сторінки даних

Заголовок сторінки містить наступну інформацію: логічний номер сторінки; логічні номери наступної і попередньої сторінок в ланцюжку;

ідентифікатор об'єкту даних, якому належить сторінка; номер наступного вільного рядка на сторінці; рівень індексу (для даних – 0); ідентифікатор індексу (для даних – 0); тип сторінки.

Слоти характеризують розміщення рядків даних на сторінці. Слот – це 4-байтове слово, 2 байти відповідають зсуву рядка даних на сторінці і 2 байти – довжині рядка. У БД кожен рядок має унікальний ідентифікатор в рамках всієї бази даних, часто званий RID – номер рядка, він має розмір 4 байти і складається з номера сторінки і номера рядка на сторінці. При впорядкуванні рядків на сторінках не відбувається фізичного переміщення рядків, всі маніпуляції відбуваються із слотами. При переповнюванні сторінок створюється спеціальний вид сторінок, званих сторінками залишку. Рядки, що не уміщаються на основній сторінці, зв'язуються зі своїм продовженням на сторінках залишку за допомогою заслань-показчиків, які містять номер сторінки і номер слота на сторінці.

Для СУБД MS SQL SERVER 6.5 розмір сторінок – 2 Kb, заголовок сторінки – 32 байти.

Етапи доступу до БД

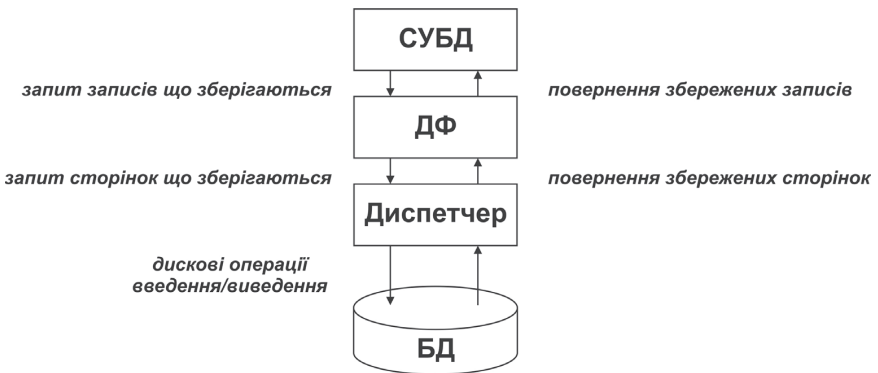


Рис.5.8.3. Схема доступу до БД

Опишемо послідовність дій при доступі до БД:

1. Спочатку в СУБД визначається шуканий запис, а потім для її витягання запрошується диспетчер файлів (ДФ).
2. Диспетчер файлів одним із розглянутих способів адресації визначає сторінку, на якій знаходиться шуканий запис, а потім для її витягання запрошується диспетчер дисків (ДД).

3. Диспетчер дисків визначає фізичне положення шуканої сторінки на диску і посилає запит на введення – виведення даних (сторінка вже може знаходитися в ОЗУ).

З погляду СУБД база даних виглядає як набір записів, які можуть бути видимими за допомогою ДФ. З погляду ДФ БД виглядає як набір сторінок, які можуть бути видимими за допомогою ДД.

ДД часто буває компонентом ОС, за допомогою якого виконуються всі операції введення/виводу, використовуючи фізичні адреси записів. Проте ДФ не обов'язково знати фізичні адреси записів, досить розглядувати диск як набір сторінок фіксованого розміру з унікальним ідентифікатором набору сторінок.

Сторінка усередині набору володіє унікальним ідентифікаційним номером сторінки.

Відповідність фізичних адрес на диску і номера сторінок досягається за допомогою ДД.

Переваги сторінкової організації - всі компоненти високого рівня не залежать від конкретного диска.

Диск – це набір файлів, що зберігаються. Файл – набір однотипних записів, що зберігаються. У спільному випадку файл, що зберігається, може зберігатися в пам'яті різними способами:

- на одному томі пам'яті (диску);
- на декількох томах;
- фізично впорядкованим відповідно до значення деякого поля, що зберігається;
- впорядкованим за допомогою одного або декількох індексів;
- впорядкованим за допомогою ланцюжків покажчиків;
- до нього може бути забезпечений доступ методом хеш-адресації;
- записи, що зберігаються, можуть бути об'єднані в блоки (декілька логічних записів в одному фізичному записі).

Набір сторінок може містити декілька файлів, що зберігаються. Кожен файл, що зберігається, має ім'я або ідентифікаційний номер (file ID), унікальний в даному наборі сторінок. А кожен запис (логічний), що зберігається, володіє ідентифікаційним номером (record ID).

ДФ виконує наступні операції з файлами.

1. Витягувати запис g , що зберігається, з файлу f , що зберігається;
2. Замінити запис g , що зберігається, у файлі f , що зберігається;
3. Видалити запис g , що зберігається, з файлу f , що зберігається;
4. Додати новий запис g , що зберігається, у файл f , що зберігається;
5. Створити новий файл f , що зберігається;
6. Видалити файл f , що зберігається.

У одних СУБД ДФ – компонент ОС, а в інших – СУБД.

Всі сторінки диска діляться на незв'язані набори. Один з наборів, набір порожніх сторінок, - вільний простір на диску.

Операції, виконувані ДД з наборами сторінок.

1. Витягувати сторінку Р з набору S;
2. Замінити сторінку Р з набору S;
3. Додати нову сторінку в набір S (витягувати її з набору порожніх сторінок і додати в набір S);
4. Видалити сторінку Р з набору S (помістити її в набір порожніх сторінок).

Приклад 5.8.1. Розглянемо БД «Замовлення деталей», яка містить таблиці ПОСТАЧАЛЬНИКИ (Р1, Р2, Р3, Р4, Р5); ДЕТАЛІ (Д1, Д2, Д3, Д4, Д5, Д6); ПОСТАЧАННЯ(Рд1, Рд2, Рд3, Рд4, Рд5, Рд6).

Для розміщення БД буде створений набір сторінок:

0	1	2	3	4	5
	Р1	Р2	Р3	Р4	Р5
6	7	8	9	10	11
Д1	Д2	Д3	Д4	Д5	Д6
12	13	РД2	14	15	17
РД1		РД3	РД4	РД5	РД6
18	19	порожні сторінки на диску			

На сторінці з номером 0 зберігається інформація про структуру БД: кількості записів в таблиці; їх розподілі по сторінках; про номери і кількість порожніх сторінок. Виконаємо дії з модифікації БД.

Додати запис про постачальника Р6. Для цього ДФ вставляє новий запис, що зберігається, а ДД шукає першу порожню сторінку (18), а потім додає її до набору сторінок постачальників. Видалити запис про постачальника Р2. ДФ видаляє запис, а ДД повертає сторінку 2 в набір порожніх сторінок.

Додати новий запис про деталь Д7. Для цього ДФ вставляє новий запис, що зберігається, а ДД шукає першу порожню сторінку (2), а потім додає її до набору сторінок про деталі.

Після виконання дій з модифікації не можна гарантувати, що логічно близькі записи фізично розташовуватимуться поруч. Тому логічну послідовність сторінок в даному наборі слід задавати за допомогою покажчиків.

Для деякого файлу, що зберігається, завжди можна здійснити послідовний доступ до всіх записів, що зберігаються, зазвичай в порядку зростання RID (під терміном «послідовний» розуміємо доступ згідно послідовності записів усередині сторінки і послідовності сторінок усередині набору сторінок). Така послідовність називається фізичною, хоча вона не

завжди відповідає фізичному розташуванню даних на диску. Це найбільш простий спосіб доступу до даних - послідовне сканування.

Для прискорення пошуку використовуються технології хешування, індексування, пошуку з використанням В-дерев.

5.9. Базові поняття реляційних баз даних

Реляційні (від англійського слова relation – відношення) моделі були розроблені Е. Коддом на початку 70-х років. Основними поняттями реляційних баз даних є тип даних, домен, атрибут, кортеж, ключі, відношення, схема відношення.

➤ *Атрибут* – це найменша поименована одиниця даних, до якої СУБД може адресуватися безпосередньо і за допомогою якої виконується побудова решти всіх структур. Атрибут має ім'я і значення.

Спершу покажемо значення цих понять на прикладі відношення СПІВРОБІТНИКИ, що містить інформацію про співробітників деякої організації (рис. 5.9.1).

➤ *Тип даних.* Поняття тип даних в реляційній моделі даних повністю адекватно поняттю типу даних в мовах програмування. Зазвичай в сучасних реляційних БД допускається зберігання даних наступних типів: символьних, числових, бітових, спеціалізованих числових даних (таких як «гроші», «темпоральних» даних (дата, час, часовий інтервал). Достатньо активно розвивається підхід до розширення можливостей реляційних систем абстрактними типами даних (відповідними можливостями володіють, наприклад, системи сімейства Ingres/Postgres). У нашому прикладі ми маємо справу з даними трьох типів: рядки символів, цілі числа і «гроші».

➤ *Домен* – допустима потенційна безліч значень простого типу даних. Поняття домена більш специфічно для баз даних, хоча і має деякі аналогії з підтипами в деяких мовах програмування. У найзагальнішому вигляді домен визначається завданням деякого базового типу даних, до якого належать елементи домена, і довільного логічного вираження, вживаного до елемента типу даних. Якщо обчислення цього логічного виразу дає результат «істина», то елемент даних є елементом домена.

Наприклад, домен «Імена» в нашому прикладі визначений на базовому типі рядків символів, але в число його значень можуть входити тільки ті рядки, які можуть представляти ім'я (зокрема, такі рядки не можуть починатися з м'якого знаку).

Слід зазначити також семантичне навантаження поняття домена: дані вважаються за порівнянні тільки у тому випадку, коли вони належать до

одного домена. У нашому прикладі значення доменів «Номера пропусків» і «Номера відділів» належать до типа цілих чисел, але не є порівняними. Відзначимо, що в більшості реляційних СУБД поняття домена не використовується, хоча в Oracle V.7 воно вже підтримується.

➤ **Схема відношення, схема бази даних.** *Схема відношення* - це іменована безліч пар { ім'я атрибуту – ім'я домена (або типа, якщо поняття домена не підтримується)}. Ступінь або «арність» схеми відношення - потужність цієї безлічі. Ступінь відношення СПВРОБІТНИКИ рівна чотирьом, тобто воно є 4-арним. Якщо всі атрибути одного відношення визначені на різних доменах, осмислено використовувати імена відповідних доменів для іменування атрибутів (не забуваючи, звичайно, про те, що це є всього лише зручним способом іменування і не усуває відмінності між поняттями домену і атрибуту). *Схема БД* (у структурному сенсі) - це набір іменованих схем стосунків

➤ **Кортеж, відношення, ключі.** *Кортеж, відповідний даній схемі відношення* - це безліч пар {ім'я атрибуту - значення}, яка містить одне входження кожного імені атрибуту, що належить схемі відношення. «Значення» є допустимим значенням домена даного атрибуту (або типа даних, якщо поняття домена не підтримується). Ступінь або «арність» кортежу, тобто число елементів в ньому, збігається з «арністю» відповідної схеми відношення. *Кортеж* - це набір іменованих значень заданого типа.

Відношення - це безліч кортежів, відповідних одній схемі відношення. Інколи, щоб не плутатися, говорять «відношення-схема» і «відношення-екземпляр», інколи схему відношення називають *заголовком відношення*, а відношення як набір кортежів - *тілом відношення*. Насправді, поняття схеми відношення щонайближче до поняття структурного типа даних в мовах програмування. Було б цілком логічно дозволяти окремо визначати схему відношення, а потім одне або декілька відношень з даною схемою. Проте в реляційних базах даних це не прийнято. Ім'я схеми відношення в таких базах даних завжди збігається з ім'ям відповідного відношення-екземпляра. У класичних реляційних базах даних після визначення схеми бази даних змінюються тільки відношення-екземпляри. У них можуть з'являтися нові і видалятися або модифікуватися існуючі кортежі. Проте в багатьох реалізаціях СУБД допускається і зміна схеми бази даних: визначення нових і зміна існуючих схем відношення. Це прийнято називати *еволюцією схеми бази даних*.

Звичайним представленням відношення є таблиця, заголовком якої є схема відношення, а рядками - кортежі відношення-екземпляра; в цьому випадку іменами атрибутів іменують стовпці цієї таблиці. Тому інколи говорять «стовпець таблиці», маючи на увазі «атрибут відношення». Коли ми

перейдемо до розгляду практичних питань організації реляційних БД і засобів управління, ми використовуватимемо цю життєву термінологію. Цієї термінології дотримуються в більшості комерційних реляційних СУБД.

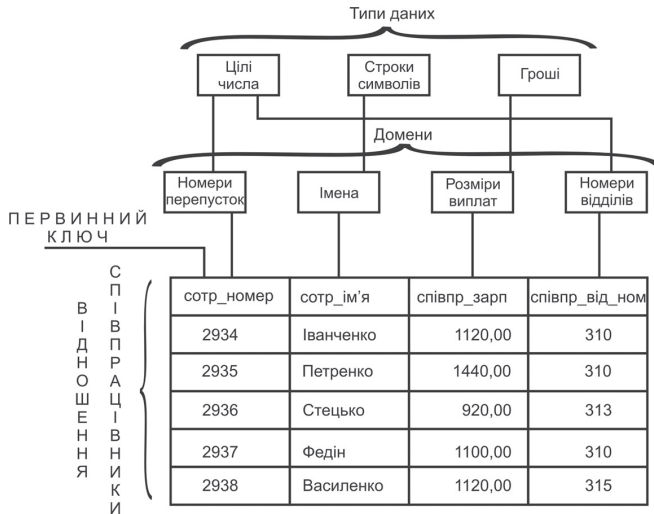


Рис. 5.9.1. Приклад відношення СПІВРОБІТНИКИ

Терміни, якими оперує реляційна модель даних, мають відповідні «табличні» синоніми, представлені в таблиці 5.9.1.

Таблиця 5.9.1. Відповідність термінів

Реляційний термін	Відповідний «табличний» термін
База даних	Набір таблиць
Схема бази даних	Набір заголовків таблиць
Відношення	Таблиця
Заголовок відношення	Заголовок таблиці
Тіло відношення	Тіло таблиці
Атрибут відношення	Стовпчик (колонка) таблиці
Кортеж відношення	Строка таблиці
Ступінь (-арність) відношення	Кількість стовпчиків таблиці
Потужність відношення	Кількість строк у таблиці
Домени і типи даних	Типи даних в клітинках таблиці

Реляційна база даних - це набір стосунків, імена яких збігаються з іменами схем стосунків в схемі БД.

➤ *Ключ* – набір атрибутів, значення яких однозначно ідентифікують кортежі. Відношення може мати декілька ключів, але завжди один з ключів оголошується первинним і його значення не можуть оновлюватися. Решта всіх ключів називається *можливими ключами*. Атрибути, що є копіями ключів інших відношень, називаються *зовнішніми ключами*.

Як видно, основні структурні поняття реляційної моделі даних (якщо не рахувати поняття домена) мають дуже просту інтуїтивну інтерпретацію, хоча в теорії реляційних БД всі вони визначаються абсолютно формально і точно.

5.10. Властивості відношень

Зупинимось тепер на чотирьох важливих властивостях відношень, які виходять з наведених раніше визначень.

Відсутність кортежів-дублікатів. Та властивість, що відношення не містять кортежів-дублікатів, виходить з визначення відношення як безлічі кортежів. У класичній теорії безлічі за визначенням кожна безліч складається з різних елементів.

З цієї властивості витікає наявність біля кожного відношення так званого *первинного ключа* - набору атрибутів, значення яких однозначно визначають кортеж відношення. Для кожного відношення принаймні повний набір його атрибутів володіє цією властивістю. Проте при формальному визначенні первинного ключа потрібне забезпечення його «мінімальності», тобто в набір атрибутів первинного ключа не повинні входити такі атрибути, які можна відкинути без збитку для основної властивості, - однозначно визначати кортеж. Поняття первинного ключа є виключно важливим у зв'язку з поняттям цілісності баз даних.

Відсутність впорядкованості кортежів. Властивість відсутності впорядкованості кортежів відношення також є наслідком визначення відношення-екземпляра як безлічі кортежів. Відсутність вимоги до підтримки порядку на безлічі кортежів відношення дає додаткову гнучкість СУБД при зберіганні баз даних в зовнішній пам'яті і при виконанні запитів до бази даних. Це не суперечить тому, що при формулюванні запиту до БД, наприклад, на мові SQL можна зажадати сортування результуючої таблиці відповідно до значень деяких стовпців. Такий результат, взагалі кажучи, не відношення, а деякий впорядкований список кортежів.

Відсутність впорядкованості атрибутів. Атрибути відношень не впорядковані, оскільки за визначенням схема відношення є безліч пар {ім'я атрибуту – ім'я домена}. Для зазначення значення атрибуту в кортежі відношення завжди використовується ім'я атрибуту. Ця властивість теоретично дозволяє, наприклад, модифікувати схеми існуючих відношень не лише шляхом додавання нових атрибутів, але і шляхом видалення існуючих атрибутів. Проте в більшості існуючих систем така можливість не допускається, і хоча впорядкованість набору атрибутів відношення явно не потрібна, часто як неявний порядок атрибутів використовується їх порядок в лінійній формі визначення схеми відношення.

Атомарність значень атрибутів. Значення всіх атрибутів є атомарними. Це витікає з визначення домена як потенційної безлічі значень простого типу даних, тобто серед значень домена не може міститися безліч значень (відношення). Прийнято говорити, що в реляційних базах даних допускаються тільки нормалізовані відношення або відношення, представлені в першій нормальній формі. Нормалізовані відношення складають основу класичного реляційного підходу до організації баз даних. Вони володіють деякими обмеженнями (не будь-яку інформацію зручно представляти у вигляді плоских таблиць), але істотно спрощують маніпулювання даними.

5.11. Характеристика реляційної моделі даних

Найбільш поширене трактування реляційної моделі даних, мабуть, належить Дейту К., який відтворює її (з різними уточненнями) практично у всіх своїх книгах. Згідно Дейту реляційна модель складається з трьох частин, що описують різні аспекти реляційного підходу: структурної частини, маніпуляційної частини і цілісної частини.

➤ У *структурній частині реляційної моделі даних* фіксується, що єдиною структурою даних, використовуваною в реляційних БД, є нормалізоване n -арне відношення. По суті справи, в попередніх двох параграфах ми розглядали саме поняття і властивості структурної складової реляційної моделі.

➤ У *маніпуляційній частині реляційної моделі даних* затверджуються два фундаментальні механізми маніпулювання реляційними БД - реляційна алгебра і реляційне числення.

Перший механізм базується в основному на класичній теорії безлічі (з деякими уточненнями), а другий - на класичному логічному апараті числення предикатів першого порядку.

➤ У *цілісній частині реляційної моделі даних* фіксуються дві базові вимоги цілісності, які повинні підтримуватися в будь-якій реляційній СУБД.

Перша вимога називається *вимогою цілісності суті*. Об'єктам або суті реального світу в реляційних БД відповідають кортежі відношень. Конкретна вимога полягає в тому, що будь-який кортеж будь-якого відношення відрізняється від будь-якого іншого кортежу цього відношення, тобто іншими словами, будь-яке відношення повинне володіти первинним ключем. Як видно з попереднього розділу, ця вимога автоматично задовольняється, якщо в системі не порушуються базові властивості відношень.

Друга вимога називається *вимогою цілісності по засланнях* і є складнішою. Очевидно, що при дотриманні нормалізованості відношень складна суть реального світу представляється в реляційній БД у вигляді декількох кортежів декількох відношень. Наприклад, уявимо, що нам потрібно представити в реляційній БД суть ВІДДІЛ з атрибутами Отд_номер (номер відділу), Отд_кол (кількість співробітників) і Отд_сотр (набір співробітників відділу). Для кожного співробітника потрібно зберігати Сотр_номер (номер співробітника), Сотр_ім'я (ім'я співробітника) і Сотр_зарп (заробітна плата співробітника). При правильному проектуванні відповідної БД в ній з'являться два відношення: ВІДДІЛИ (Отд_номер, Отд_кол) первинний ключ - Отд_номер і СПІВРОБІТНИКИ (Сотр_номер, Сотр_ім'я, Сотр_зарп, Сотр_отд_ном) первинний ключ - Сотр_номер.

Як видно, атрибут Сотр_отд_ном з'являється у відношенні СПІВРОБІТНИКИ не тому, що номер відділу є власною властивістю співробітника, а лише для того, щоб мати можливість відновити при необхідності повну суть ВІДДІЛ. Значення атрибуту Сотр_отд_ном в будь-якому кортежі відношення СПІВРОБІТНИКИ повинно відповідати значенню атрибуту Отд_ном в деякому кортежі відношення ВІДДІЛИ. Атрибут такого роду називається зовнішнім ключем, оскільки його значення однозначно характеризують суть, представлену кортежами деякого іншого відношення (тобто задають значення їх первинного ключа). Говорять, що відношення, в якому визначений зовнішній ключ, посилається на відповідне відношення, в якому такий же атрибут є первинним ключем.

Вимога цілісності по засланнях, або вимога зовнішнього ключа полягає в тому, що для кожного значення зовнішнього ключа, що з'являється у відношенні, що посилається, у відношенні, на яке веде заслання, повинен знайтися кортеж з таким же значенням первинного ключа, або значення зовнішнього ключа має бути невизначеним (тобто ні на що не указувати). Для нашого прикладу це означає, що якщо для співробітника вказаний номер відділу, то цей відділ повинен існувати. Обмеження цілісності суті і по засланнях повинні підтримуватися СУБД. Для дотримання цілісності суті досить гарантувати відсутність в будь-якому відношенні кортежів з одним і тим же значенням первинного ключа. З цілісністю по засланнях справи йдуть

декілька складніше. Зрозуміло, що при оновленні відношення (вставці нових кортежів або модифікації значення зовнішнього ключа в існуючих кортежах), що посилається, досить стежити за тим, щоб не з'явилися некоректні значення зовнішнього ключа. Але як бути при видаленні кортежу з відношення, на яке веде заслання?

Існують три підходи, кожен з яких підтримує цілісність по засланнях:

1) забороняється проводити видалення кортежу, на який існують заслання (тобто спочатку потрібно або видалити кортежі, що посилаються, або відповідним чином змінити значення їх зовнішнього ключа);

2) при видаленні кортежу, на який є заслання, у всіх кортежах, що посилаються, значення зовнішнього ключа автоматично стає невизначеним;

3) каскадне видалення, що полягає в тому, що при видаленні кортежу з відношення, на яке веде заслання, з відношення, що посилається, автоматично віддаляються всі кортежі, що посилаються.

У розвинених реляційних СУБД зазвичай можна вибрати спосіб підтримки цілісності по засланнях для кожної окремої ситуації визначення зовнішнього ключа. Звичайно, для ухвалення такого рішення необхідно аналізувати вимоги конкретної прикладної області.

Питання і завдання для самостійної роботи

1. Що розуміється під веденням даних?
2. Чи можна використовувати терміни «база даних» і «банк даних» як еквівалентні?
3. Які функції по відношенню до користувача виконує СУБД?
4. Що включають вимоги надійності і безпеки БД?
5. Чим характеризуються БД першого покоління?
6. Чим даталогічні документальні моделі відрізняються від фактографічних?
7. Наведіть приклади даталогічних документальних моделей.
8. Які компоненти входять в структуру логічної (даталогічної) моделі?
9. Назвіть структури даних ієрархічних моделей.
10. Що включає фізична модель даних?
11. Чим характеризується послідовний доступ до даних?
12. Чим характеризується прямий (довільний) доступ до даних?
13. Які методи адресації використовуються для прискорення доступу до даних?
14. Дайте характеристику методу хешування.
15. Опишіть алгоритм адресації з використанням індексно-послідовного файлу?

16. Що таке сторінка даних? Опишіть її структуру.
17. Вкажіть послідовність дій доступу до даних.
18. Як зв'язані сторінки даних в набори?
19. Чим відрізняється домен від типа даних?
20. Що таке ступінь відношення?
21. У чому відзнака схеми відношення від відношення?
22. Чи можна вважати будь-яку прямокутну таблицю даних за відношення?
23. У чому, на вашу думку, основа популярності реляційної моделі?
24. Наведіть приклад БД і вкажіть, які обмеження цілісності в ній повинні підтримуватися.
25. Які з реляційних операцій узяті з теорії множин?

Список літератури

1. Горев А., Ахаян Р., Макашаринов С. Эффективная работа с СУБД. – СПб.: Питер, 1997.
2. Пасичник В. В., Резниченко В. А. Организация баз данных и знаний. – К.: Издательская группа ВНУ, 2006. – 384 с.
3. Стасьшин В.М. Введение в проектирование реляционных баз данных: Учебное пособие по курсу «Базы данных». – Новосибирск, НГТУ, 1999.
4. Сытник Н. В. Проектирование баз и хранилищ данных: Учеб. пособие. – К.: КНЕУ, 2004. – 348 с.
5. Фуфаев Э.В. Базы данных: учеб. пособие для студ. Сред. Проф. Образования / Э.В. Фуфаев, Д.Э. Фуфаев. – 3-е изд., стер. – М.: Издательский центр «Академия», 2007. – 320 с.
6. Хомоненко А.Д., Цыганков В.М., Мальцев М.Г. Базы данных: Учебник для высших учебных заведений / Под ред. А.Д. Хомоненко. – 4-е изд., доп. И перераб. – СПб.: КОРОНА принт, 2004. – 736 с.

Розділ VI. Офісний пакет програм Microsoft Office 2007

При влаштуванні на роботу сьогодні неодмінною вимогою до фінансиста, бухгалтера, аналітика або економіста є знання офісного програмного забезпечення, що включає офісний пакет Microsoft Office, 1С Підприємство, основи VBA і таке інше.

Розглянемо мінімальний набір завдань, що виникають щодня перед працівником офісу, а також програми, що входять до складу офісного пакету Microsoft Office, що дозволяють вирішувати ці завдання.

1. *Складання, форматування і друк документів різного ступеня складності.* Для вирішення цього завдання працівник офісу може скористатися спеціальною програмою, званою текстовим процесором або текстовим редактором. У офісний пакет Microsoft Office входить популярний текстовий редактор Microsoft Word.
2. *Складання таблиць даних, проведення розрахунків і аналіз отриманих результатів.* Для вирішення цього завдання офісний працівник може скористатися табличним редактором Microsoft Excel.
3. *Представлення результатів аналізу в наочному і готовому для демонстрації вигляді, наприклад, у вигляді електронної презентації.* Для вирішення цього завдання офісний працівник може скористатися програмою Power Point з пакету Microsoft Office.
4. *Організація зберігання великих об'ємів даних у вигляді таблиць спеціального вигляду, званих базами даних, і виконання обробки цих даних.* Для вирішення цього завдання офісний працівник може скористатися вбудованою в пакет Microsoft Office системою управління базами даних Microsoft Access.
5. *Для відправки електронних листів колегам, постачальникам або замовникам офісний працівник може скористатися програмою Microsoft Outlook пакету Microsoft Office.*



Далі будуть викладені деякі основи роботи в кожній програмі пакету MS Office, що не претендують на повноту викладу. Для глибокого вивчення можливостей кожної програми пакету читачеві пропонується вивчити спеціальну літературу, вказану наприкінці цього розділу.

У цьому підручнику ми зупинимося на вивченні нового офісного пакету MS Office 2007. Істотною відзнакою пакету MS Office 2007 від

офісних пакетів попередніх років (Microsoft Office 2000, 2002 і 2003) є наявність *стрічкового інтерфейсу* у всіх програмах пакету, головні достоїнства якого – наочність і доступність. Функціональність всіх програм офісного пакету MS Office 2007 максимально відповідає всім завданням, які можуть виникнути перед сучасним користувачем.

Домовимося під вибором деякого пункту або натисненням на кнопку панелі інструментів розуміти одинарне клацання лівою кнопкою миші по назві пункту або кнопки відповідно. Вважатимемо також, що всі програми пакету MS Office встановлені на комп'ютері читача.

6.1. Опис стрічкового інтерфейсу і виконання основних операцій в програмах пакету MS Office 2007

В верхньому лівому куті розташована кнопка Microsoft Office . З правого боку від кнопки Microsoft Office розташована панель швидкого доступу з можливістю її налаштування .

Стрічковий інтерфейс програм офісного пакету реалізований у вигляді стрічки, що містить вкладки. Перехід від однієї вкладки до іншої здійснюється клацанням лівої кнопки миші по їх назвах. Кожна вкладка має своє призначення і містить групу або групи інструментів, призначені для реалізації певного класу завдань. Вкладки, розташовані над стрічкою інструментів, по суті, замінюють пункти основного меню, використовуюваного в попередніх версіях офісного пакету MS Office 2007. Перерахуємо основні вкладки програм пакету MS Office 2007.

Вкладка **Основне** призначена для виконання багатьох основних операцій, пов'язаних з форматуванням і редагуванням тексту (операції роботи з буфером обміну, установка параметрів шрифту (зображення, розмір), команди вирівнювання тексту, установки стилів, пошуку і заміни тексту).

Вкладка **Вставлення** призначена для вставки різних елементів в документ таких як графіка, фігури, малюнок, діаграма.

Вкладка **Розмітка сторінки** (Word 2007, Excel 2007) призначена для установки і налаштування різних параметрів сторінки (поля, орієнтація, розмір сторінки).

Вкладка **Рецензування** призначена для вставки приміток, перевірки орфографії і тому подібне

Вкладка **Вигляд** призначена для установки режиму проглядання документа.

Створення, відкриття, збереження і вивід на друк документа – основні операції для роботи з будь-яким документом пакету MS Office. Перераховані операції майже у всіх програмах офісного пакету MS Office виконуються однаково.

1. *Створення нового документа.* У програмах офісного пакету створення нового документа виконується за допомогою натиснення на кнопку Office. В результаті відкриється вікно, в якому перераховані всі основні команди роботи з документом (*Створити, Відкрити, Зберегти, Зберегти як..., Друк, Підготувати, Надіслати, Опублікувати, Закрити*). Для створення нового документа слід вибрати пункт *Створити*. На екрані з'явиться діалогове вікно **Створення документа**, представлене на рисунку 6.1.1. У цьому вікні слід вибрати шаблон *Чисті та недавні* (вибраний за умовчанням), а в ньому – *Новий документ* і натиснути на кнопку *Створити*.

Створити новий документ можна за допомогою використання комбінації клавіш Ctrl+N. Діалогове вікно **Створення документа** містить такі шаблони як, наприклад, візитні картки, вітальні листівки, які можна отримати з сайту Microsoft Office, підключившись до мережі Internet.

2. *Відкриття раніше створеного документа.* Для відкриття документа слід виконати одинарне клацання лівою кнопкою миші по кнопці *Office*, потім в списку команд, що розкрився, вибрати команду *Відкрити*. В результаті на екрані з'явиться діалогове вікно **Відкриття документа**, представлене на рисунку 6.1.2.

У цьому діалоговому вікні слід вказати шлях до текстового файлу і після вибору цього файлу натиснути на кнопку *Відкрити*, яка стане активною.

Викликати діалогове вікно **Відкриття документа** можна простіше – за допомогою натиснення комбінації клавіш Ctrl+O.

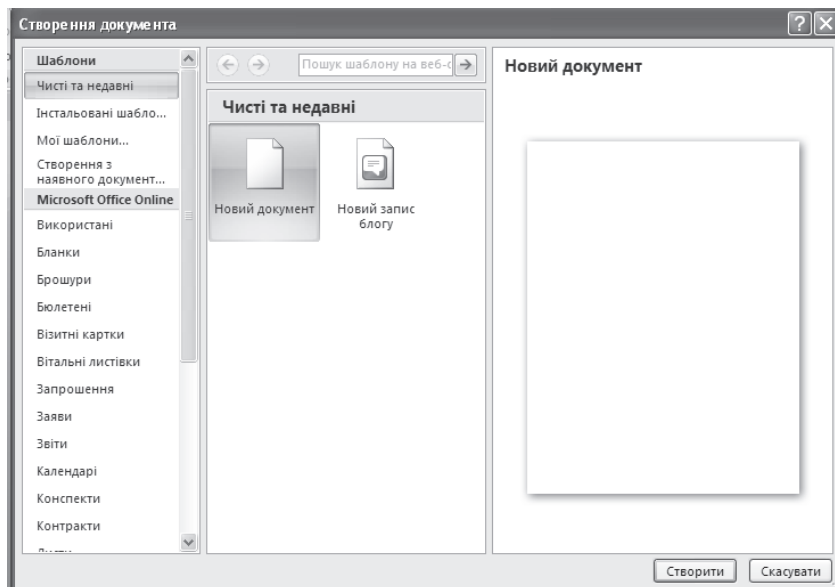


Рис.6.1.1. Діалогове вікно *Створення документа*

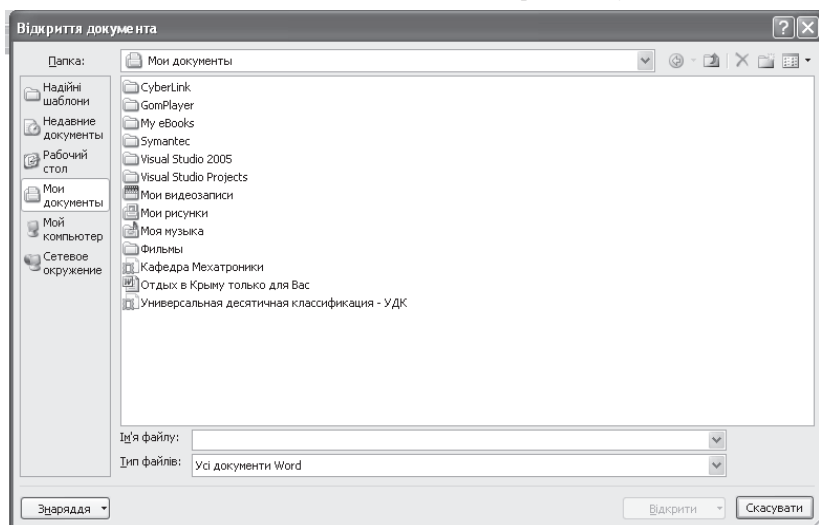


Рис.6.1.2. Діалогове вікно *Відкриття документа*

3. *Збереження документа*. Для збереження документа слід виконати одинарне клацання лівою кнопкою миші по кнопці Office, потім в списку команд, що розкрився, вибрати пункт *Зберегти* або *Зберегти як...*. При збереженні нового документа принципової різниці між цими командами немає. Для збереження існуючого документа під іншим ім'ям слід вибрати команду *Зберегти як...*. В результаті на екрані з'явиться діалогове вікно *Збереження документа*, представлене на рисунку 6.1.3. У цьому діалоговому вікні слід вказати диск і директорію, в яку слід зберегти файл, а також в рядку Ім'я файлу ввести з клавіатури ім'я файлу.

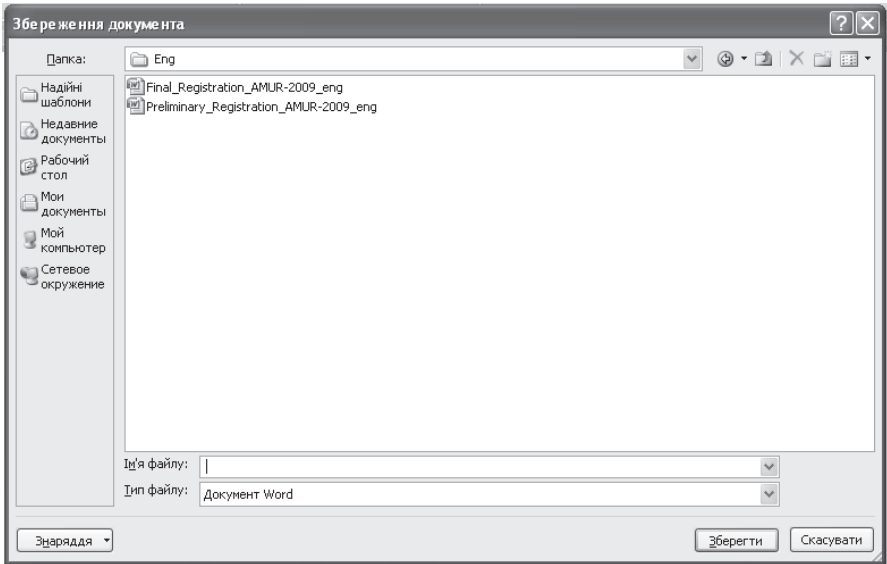


Рис.6.1.3. Діалогове вікно Збереження документа

Викликати діалогове вікно *Збереження документа* для збереження нового документа можна за допомогою комбінації клавіш Ctrl+S.

4. *Закриття документа*. Закрити документ можна або натисненням комбінації клавіш Alt+F4 або, вибравши пункт *Закрити* в списку кнопки Office, що розкривається.

6.2. Текстовий процесор Microsoft Office Word 2007

Текстовий редактор Microsoft Word 2007 – це програма офісного пакету MS Office 2007, призначена для введення, форматування, перевірки правильності написання, виводу на друк, відправки факсом або по електронній пошті, збереження у вигляді веб-сторінок текстів різного ступеня складності, зокрема, що містять графіку, формули, таблиці, списки, діаграми, відеокліпи, звукові файли і тому подібне, ведення інтернет-щоденника за допомогою оновлення блога безпосередньо з редактора.

6.2.1. Користувальницький інтерфейс текстового редактора Microsoft Office Word 2007

Для того щоб *запустити текстовий редактор Microsoft Office Word 2007*, слід виконати наступні дії:

1. на панелі завдань Windows натисніть на кнопку **Пуск** (традиційно вона розташована в лівому нижньому кутку екрану після автоматичного завантаження операційної системи Windows);
2. у діалоговому вікні, що з'явилося, виберіть пункт **Всі програми**;
3. у списку, що розкрився, виберіть пункт **Microsoft Office Word 2007**.

Результат запуску текстового редактора Microsoft Office Word 2007 представлений на рисунку 6.2.1.

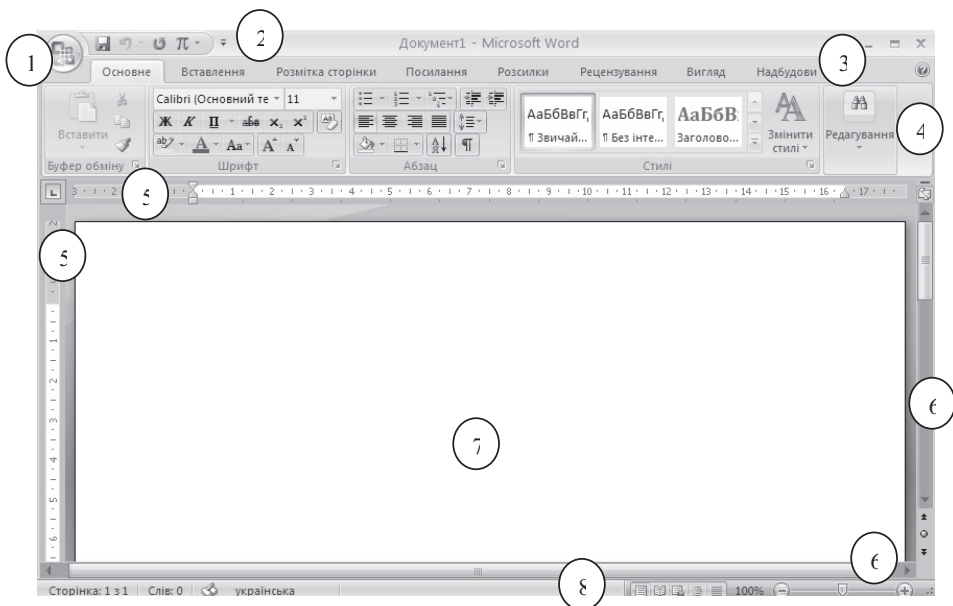



Рис. 6.2.1. Вікно редактора Microsoft Office Word 2007

Перерахуємо основні елементи вікна Microsoft Office Word 2007:

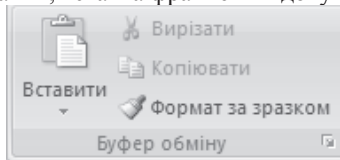
- 1 – кнопка Microsoft Office;
- 2 – панель швидкого доступу;
- 3 – вкладки стрічкового інтерфейсу;
- 4 – стрічка груп інструментів;
- 5 – горизонтальна і вертикальна лінійки;
- 6 – горизонтальна і вертикальна смуги прокрутки;
- 7 – робоча область документа;
- 8 – рядок стану.

Перерахуємо *основні елементи стрічкового інтерфейса редактора Microsoft Office Word 2007 і їх призначення*. Стрічковий інтерфейс редактора Microsoft Office Word 2007 організований у вигляді стрічки з розташованими

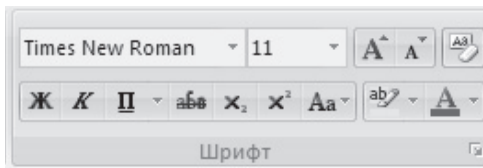
над нею вкладками. Кожна вкладка стрічки складається з групи або груп інструментів і має конкретне функціональне призначення. Зауважимо, що в правому нижньому куті майже кожної групи інструментів міститься кнопка , що дозволяє викликати допоміжне (контекстне) меню з розширеними можливостями відповідної групи.

Вкладка **Основне** призначена для редагування і форматування тексту і містить 5 груп інструментів:

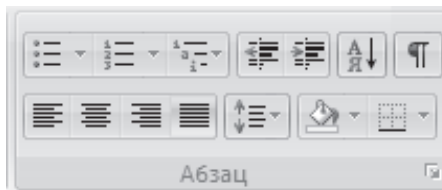
- група **Буфер обміну** містить інструменти роботи з буфером обміну, такі як копіювання, вирізування, вставка фрагментів документа;



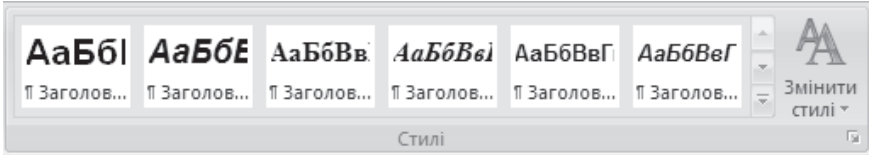
- група **Шрифт** містить інструменти управління зображенням тексту, установкою шрифту і його розміру;



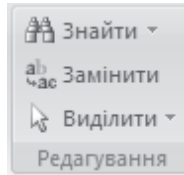
- група **Абзац** містить інструменти вирівнювання тексту (по правому краю, по центру, по лівому краю, по ширині), установка міжрядкового інтервалу і відступу, формування нумерованих списків;



- група **Стилі** містить набір стилів, кольорів і шрифтів;

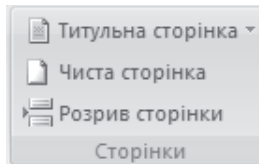


- група **Редагування** містить інструменти пошуку, заміни і виділення фрагментів тексту.

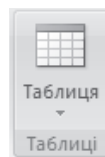


Вкладка **Вставлення** призначена для вставки в документ таблиць, малюнків, фігур, діаграм, об'єктів SmartArt, написів, об'єктів WordArt, символів, математичних формул, гіперпосилань, нижніх і верхніх колонтитулів, номерів сторінок і тому подібне і містить 7 груп інструментів:

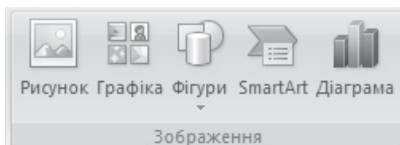
- група **Сторінки** містить інструменти вставки титульної сторінки, чистої сторінки, розривів сторінки;



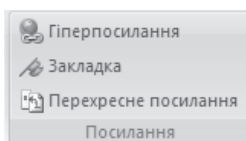
- група **Таблиці** містить інструмент Таблиця із списком способів створення таблиці, що розкривається;



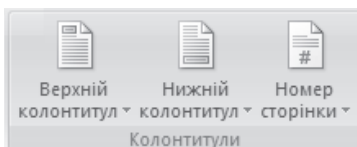
- група **Зображення** містить інструменти вставки в документ малюнка, графіки, фігури, об'єкту SmartArt, діаграми;



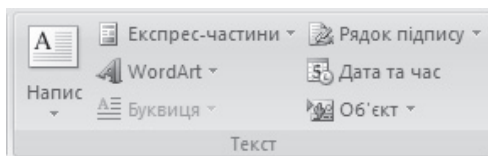
- група **Посилання** містить інструменти вставки гіперпосилання, закладки, перехресного засланя;



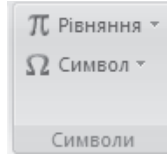
- група **Колонтитулі** містить інструменти вставки верхнього колонтитулу, нижнього колонтитулу, номера сторінки;



- група **Текст** містить інструменти вставки напису, об'єкту WordArt, дати і часу, вставки великої заголовної букви на початку абзацу (буквиця), рядка підпису, вбудованого об'єкту або тексту з файлу;



- група **Символи** містить інструменти вставки спеціальних символів і математичних формул.

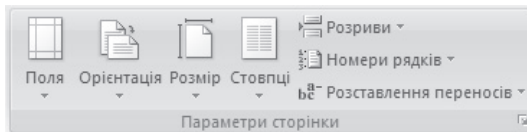


Вкладка **Розмітка сторінки** призначена для установки параметрів сторінки, розмірів відступів і інтервалів, завдання теми і способів впорядкування об'єктів і містить 5 груп інструментів:

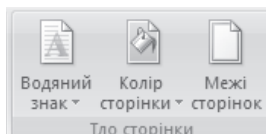
- група **Теми** містить інструменти завдання теми, що визначає вид всього документа, включаючи завдання кольору, шрифту і ефекту теми;



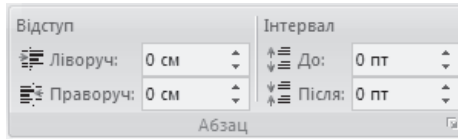
- група **Параметри сторінки** містить інструменти установки розмірів полів документа, його орієнтації, розміру паперу, розбиття тексту на стовпці, управління розривами, завдання номерів рядків на полях і розставлення перенесень;



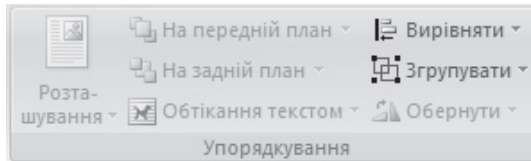
- група **Тло сторінки** містить інструменти установки фону документа, вставки прихованого тексту (водяного знаку) за вмістом сторінки, зміну кольору сторінки і завдання рамки сторінки;



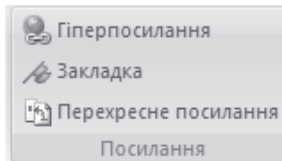
- група **Абзац** містить інструменти завдання розмірів відступів і інтервалів;



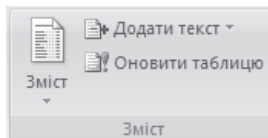
- група **Упорядкування** містить інструменти розміщення вибраного об'єкту на сторінці, угруповання сукупності графічних об'єктів, вирівнювання країв декількох об'єктів, а також можливості повороту або дзеркального віддзеркалення графічних об'єктів.



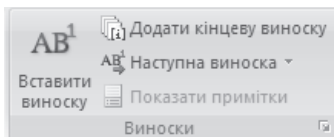
Вкладка **Посилання** призначена для створення виносок, бібліографії, списку ілюстрацій, автоматичного змісту і містить 6 груп інструментів:



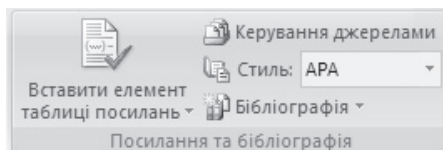
- група **Зміст** містить інструменти створення автоматичного змісту, вставки поточного абзацу як елемента змісту і оновлення змісту;



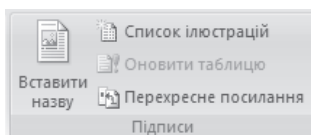
- група **Виноски** містить інструменти вставки виносок в документ, переходу між виносками, прокручування документа для відображення розташування виносок;



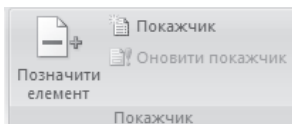
- група **Посилання та бібліографія** містить інструменти вставки елемента таблиці заслань, проглядання списку джерел, на які посилається документ, завдання стилю таблиці заслань, додавання бібліографії в документ;



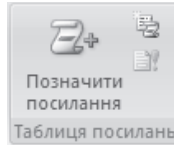
- група **Підписи** містить інструменти вставки назви малюнка або іншого об'єкту, створення списку ілюстрацій, оновлення списку ілюстрацій, після внесення змін до документа, створень перехресних заслань;



- група **Покажчик** містить інструменти додавання виділеного тексту до покажчика документа – списку ключових слів з переліком сторінок, на яких ці слова зустрічаються, вставки покажчика в документ і оновлення покажчика;

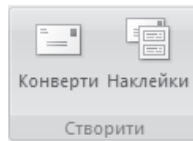


- група **Таблиця посилань** містить інструменти вставки виділеного тексту в таблицю заслань, вставки в документ таблиці заслань і оновлення таблиці заслань після внесення змін до документа.

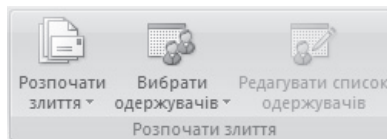


Вкладка **Розсилки** призначена для роботи з електронною поштою і містить 5 груп інструментів:

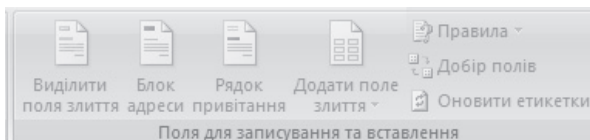
- група **Створити** містить інструменти для створення і друку конвертів і наклеювання;



- група **Розпочати злиття** містить інструменти для створення форми листа, який необхідно відправити по електронній пошті різним одержувачам, вибору списку одержувачів і редагування списку одержувачів;

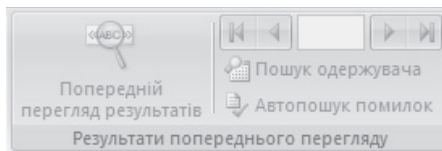


- група **Поля для записування та вставлення** містить інструменти вставки адреси в лист, рядка вітання, полів із списку одержувачів тощо;

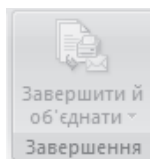


- група **Результати попереднього перегляду** містить інструменти заповнення полів злиття документа даними із списку одержувачів з

можливістю попереднього перегляду, пошуку одержувача в списку одержувачів тощо;

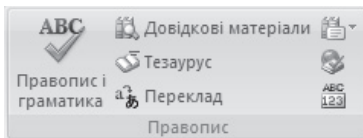


- група **Завершення** містить інструменти створення для кожної копії листа окремих документів з можливістю друку і відправки електронною поштою.

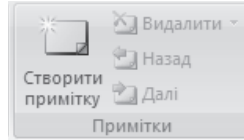


Вкладка **Рецензування** призначена для редагування тексту в режимі запам'ятовування виправлень, вставки приміток в текст документа і містить 6 груп інструментів:

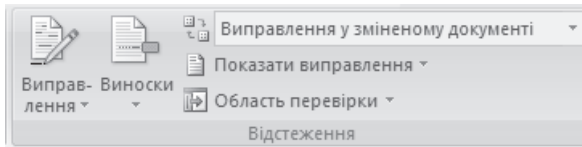
- група **Правопис** містить інструменти перевірки правопису і граматики, пошуку слів у вбудованих словниках і енциклопедіях, тезауруса (пошуку слів однакових за значенням з виділеним словом), вибору мови і визначення кількості рядків, абзаців, символів і слів в документі;



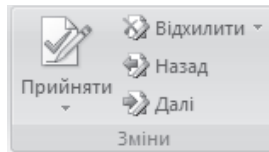
- група **Примітки** містить інструменти для вставки примітки до виділеного фрагмента тексту, видалення примітки, переміщення між примітками;



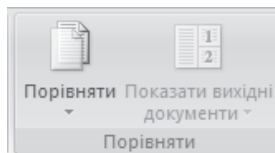
- група **Відстеження** містить інструменти для контролю внесених в документ змін, перегляду змін і завдання області перевірки виправлень, завдання способу відображення виправлень в документі;



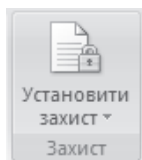
- група **Зміни** містить інструменти для відміни змін, переміщення між змінами;



- група **Порівняти** містить інструменти для порівняння і об'єднання декількох версій документа, вибір документів для відображення;

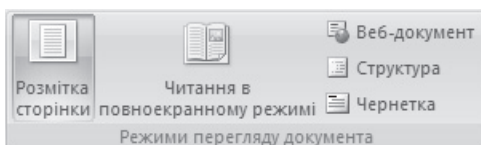


- група **Захист** містить інструменти для установки способу доступу до документа.

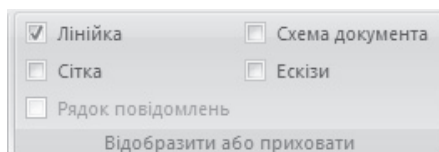


Вкладка **Вигляд** призначена для редагування тексту в режимі запам'ятовування виправлень, вставки приміток в текст документа і містить 5 груп інструментів:

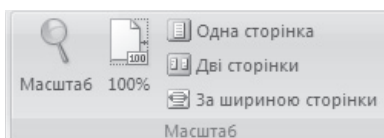
- група **Режим перегляду документа** містить інструменти для попереднього перегляду документа перед його виводом на друк, відображення документа в повноекранному режимі, відображення у вигляді веб-документа, проглядання структури, проглядання документа у вигляді чернетки;



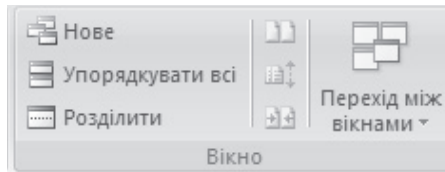
- група **Відобразити або приховати** містить інструменти для відображення лінійки, сітки, ескізів сторінок документа, схеми документа, рядка повідомлень;



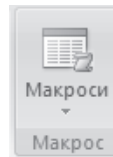
- група **Масштаб** містить інструменти для установки масштабу (у відсотках) відображення документа і способу відображення документа (одна або дві сторінки);



- група **Вікно** містить інструменти розміщення і впорядкування декількох вікон і переходу між ними;



- група **Макрос** містить інструменти для запису, запуску або видалення макросів.



6.2.2. Робота з текстом в Microsoft Office Word 2007

1. **Налаштування параметрів сторінки.** Перед введенням тексту слід настроїти параметри сторінки. Для цього слід перейти на вкладку *Розмітка* сторінки і вибрати кнопку *Поля*, що розкривається. На екрані з'явиться список варіантів полів (звичайне, вузьке, середнє, широке, дзеркальне) і пункт *Настроювані поля...*, вибір якого приведе до появи діалогового вікна **Параметри сторінки** (вкладка *Поля* активна за умовчанням), що дозволяє ввести з клавіатури будь-який допустимий розмір полів, вказати орієнтацію (книжкова або альбомна). Діалогове вікно **Параметри сторінки** представлено на рисунку 6.2.2. Змінити орієнтацію сторінки можна за допомогою кнопки *Орієнтація* на вкладці *Розмітка сторінки* в групі *Параметри сторінки*.

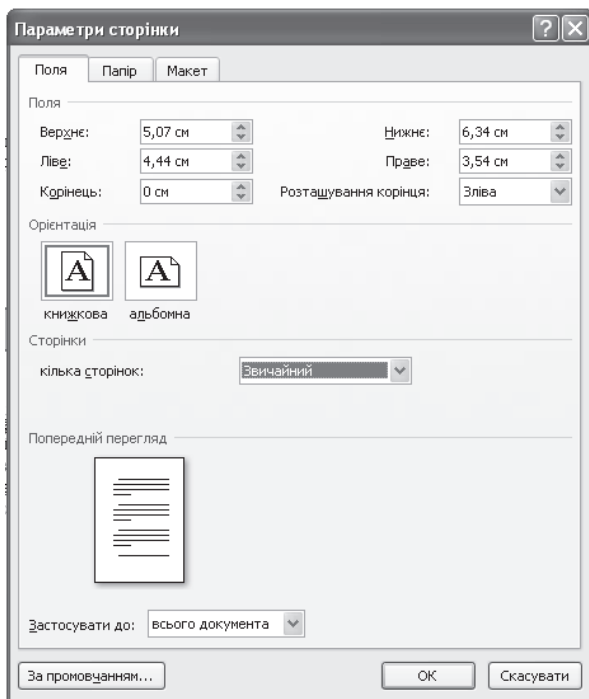


Рис.6.2.2. Діалогове вікно Параметри сторінки

У групі інструментів *Параметри сторінки* на вкладці *Розмітка сторінки* кнопка *Розмір* дозволяє встановити розмір сторінки шляхом вибору одного з варіантів: Letter (21,59 см x 27,94 см), Legal (21,59 см x 35, 56 см), Executive (18,41 см x 26,67 см), A5 (14,8 см x 21 см), B5 (18,2 см x 25,7 см), A4 (21 см x 29,7 см), конверт Monarch (9,84 см x 19,05 см), конв. №10 (10,48 см x 24,13 см), конверт DL (11 см x 22 см), конверт C5 (16,2 см x 22,9 см) або встановити необхідний розмір за допомогою вибору пункту *Інші розміри аркушів*, що приведе до появи діалогового вікна *Параметри сторінки* з активною вкладкою *Папір*.

Для можливості розміщення тексту в дві, три або більше колонок слід натиснути на кнопку *Стовпці*, що розкривається (вкладка *Розмітка сторінки*, група інструментів *Параметри сторінки*). Для установки кількості колонок, їх ширини і розміру проміжку між колонками слід вибрати пункт *Додаткові*

стовпці..., що приведе до появи діалогового вікна *Стовпці*, представленого на рисунку 6.2.3.

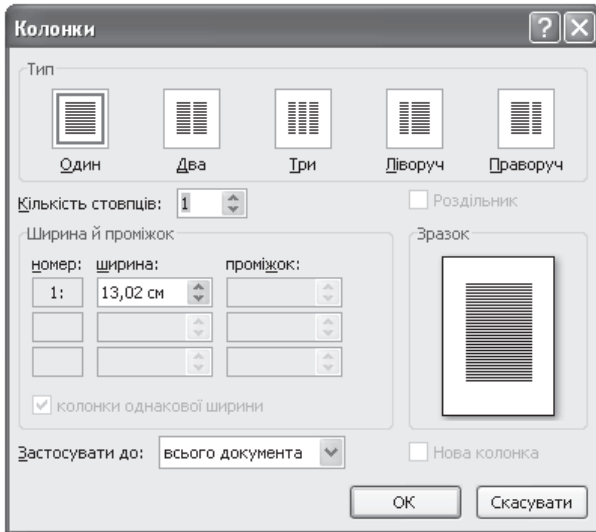
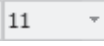
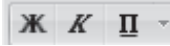









Рис.6.2.3. Діалогове вікно Колонки

При вставці текстів комп'ютерних програм (лістингів) виникає необхідність нумерації рядків. Пронумерувати рядки можна натисненням на кнопку *Номери рядків*, що розкривається на вкладці *Розмітка сторінки* в групі *Параметри сторінки*. За умовчанням нумерація рядків відключена. Користувач може вибрати область для нумерації рядків – на кожній сторінці, в кожному розділі, заборонити для поточного абзацу.

Налаштувати можливість розставляння переносів в словах можна за допомогою кнопки *Розставлення переносів*, що розкривається.

2. Установка параметрів шрифту. Перед вводом тексту рекомендується установити шрифт, розмір, накреслення, колір фону текста і колір шрифту, а також спосіб вирівнювання текста. Для установки більшої частини з перелічених параметрів слід переключатися на вкладку *Основне* і використовувати інструменти групи Шрифт. Для установки шрифту слід обрати список зміна шрифту *Times New Roman*, що розкривається і вибрати одинарним клацанням лівої кнопки миші потрібний шрифт. Аналогічно

встановлюється розмір  шрифту. Накреслення можна задати за допомогою однієї з слідуєчих кнопок (жирне накреслення, курсивне накреслення, підкреслення виділеного тексту) . Кнопка Підкреслення, що розкривається, дозволяє вибрати тип лінії підкреслення та її колір. Завдати колір шрифту можна за допомогою кнопки , що розкривається, а колір фона виділеного тексту – за допомогою кнопки , що розкривається. Ввести нижній індекс можна за допомогою натискання на кнопку , а верхній – за допомогою кнопки . Закреслити виділений текст можна одинарним клацанням по кнопці , а змінити регістр – за допомогою кнопки , що розкривається.

Вирівнювання текста по правому краю, по центру, по лівому краю, за шириною можна одинарним клацанням лівою кнопкою миші відповідно по кнопкам , що розташовані в групі інструментів Абзац на вкладці Основне.

3. Способи виділення тексту. Перш ніж змінити параметри тексту або виконати переміщення і копіювання, його необхідно виділити. Найпростіший спосіб виділення тексту – натискувати на початку фрагмента, що виділяється, ліву кнопку миші і, утримуючи її, «протягнути» до кінця фрагмента, що виділяється, після чого слід відпустити ліву кнопку миші. Для швидкого виділення конкретного слова в тексті досить виконати по ньому подвійне клацання лівою кнопкою миші, для виділення конкретного абзацу – виконати в будь-якому місці абзацу потрійне клацання лівою кнопкою миші. Для виділення тексту всього документа слід натискувати комбінацію клавіш Ctrl+A. Для виділення вертикального блоку тексту слід при натиснутій клавіші Alt виділити блок по вертикалі за допомогою натиснення і утримання лівої кнопки миші. Для виділення горизонтального блоку тексту слід встановити курсор в початок блоку, натиснути ліву кнопку миші і, утримуючи її, «протягнути» курсор уздовж фрагмента, що виділяється. Виділений фрагмент тексту виділяється блакитно-сірим кольором.

Після виділення тексту на екрані з'явиться майже прозора панель з інструментами форматування тексту. Для прояву цієї панелі на екрані слід підвести до неї покажчик миші. Панель з інструментами форматування представлена на рисунку 6.2.4.

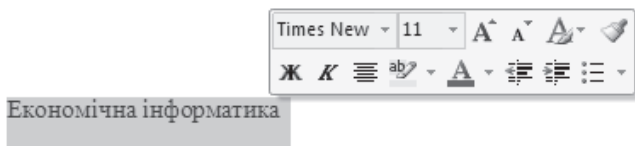


Рис.6.2.4. Панель з інструментами форматування тексту

Панель на рисунку 6.2.4 містить багато інструментів з групи інструментів Шрифт вкладки Основне. Використовуючи список Times New, що розкривається, можна встановити потрібний шрифт (Times New Roman, Arial и т.д.), список 11, що розкривається – встановити розмір. Одним клацанням лівої кнопки миші можна змінити накреслення тексту на жирний Ж, курсив К, встановити вирівнювання тексту по центру. Використовуючи кнопки, що розкриваються, можна встановити потрібний фон тексту ab (за умовчанням жовтий) або потрібний колір тексту A (за умовчанням червоний).

4. Копіювання і переміщення тексту. Копіювати, вирізувати або вставляти текст можна за допомогою групи інструментів Буфер обміну, розташованих на вкладці Основне. Для того, щоб скопіювати фрагмент тексту, його необхідно спочатку виділити, потім або натиснути праву кнопку миші і в допоміжному меню, що з'явилося, вибрати пункт Копіювати, або натиснути на кнопку Копіювати в групі інструментів буфер обміну, або натиснути комбінацію клавіш Ctrl+C, або натиснути комбінацію клавіш Ctrl+Insert.

Для того, щоб вирізувати фрагмент тексту, його слід заздалегідь виділити, а потім або натиснути праву кнопку миші і в допоміжному меню, що з'явилося, вибрати пункт Вирізати, або натиснути на кнопку Вирізати в групі інструментів Буфер обміну, або натиснути комбінацію клавіш Shift+Delete.

Для вставки скопійованого або вирізаного фрагмента тексту необхідно встановити курсор в місце вставки, а потім або натиснути праву

кнопку миші і в допоміжному меню, що з'явилося, вибрати пункт *Вставити*, або натиснути на кнопку *Вставити*, що розкривається, в групі інструментів *Буфер обміну*, або натиснути комбінацію клавіш Ctrl+V, або натиснути комбінацію клавіш Shift+Insert.

Для переміщення фрагмента тексту його необхідно виділити, а потім натиснути ліву кнопку миші на виділеному фрагменті тексту і, утримуючи її, «відбуksирувати» текст в потрібне місце документа.

6.2.3. Створення маркованих и нумерованих списків засобами Microsoft Office Word 2007

Під списком, як правило, розуміють перелік деяких елементів, об'єднаних єдиною тематикою. Якщо для позначення елементів списку використовується цифри, то список називається *нумерованим*. Якщо для позначення елементів списку використовуються значки-маркери, то список називається *маркованим*.

Для створення списку будь-якого з перелічених типів слід перейти на вкладку *Основне* в групу інструментів *Абзац*. Створення маркованого списку виконується за допомогою натиснення на кнопку *Маркери*, що розкривається. Далі у списку маркерів, що розкривається, – у бібліотеці маркерів слід вибрати потрібний маркер одинарним клацанням лівої кнопки миші по ньому. Бібліотека маркерів представлена на рисунку 6.2.5.

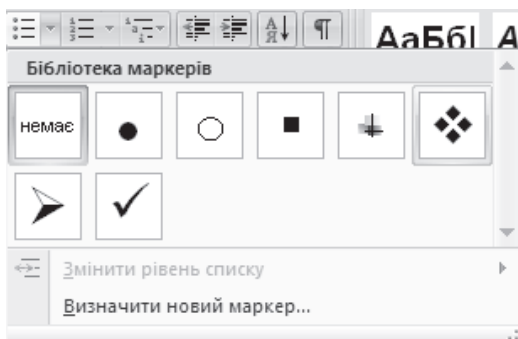


Рис. 6.2.5. Бібліотека маркерів

При необхідності користувач може ввести маркер будь-якого вигляду, скориставшись командою *Визначити новий маркер...* зі списку кнопки

Маркери, що розкривається. Діалогове вікно **Визначення нового маркера** представлено на рисунку 6.2.6. При натисненні на кнопку *Символ...* на екрані з'являється однойменне діалогове вікно, в якому одинарним клацанням лівої кнопки миші можна вибрати необхідний символ для маркера списку. При натисненні на кнопку *Малюнок...* на екрані з'являється діалогове вікно **Мальований маркер**, в якому можна вибрати один з готових малюнків для маркера, або розширити список малюнків, підключившись до мережі Інтернет і, встановивши прапорець напроти пункту *Включати вміст із сайту Office Online*, вставити малюнок маркера за допомогою кнопки *Імпорт...* . У вікні *Попередній перегляд* можна бачити як виглядатимуть вибрані маркери в текстовому документі.

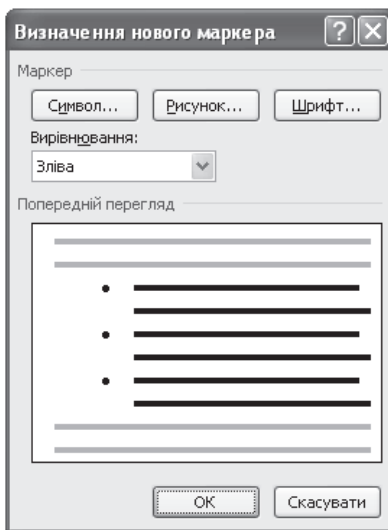



Рис.6.2.6. Діалогове вікно **Визначення нового маркера**

Створення нумерованного списку виконується за допомогою натиснення по кнопці *Нумерація* , що розкривається. В списку, що розкривається, представлена *Бібліотека нумерованих списків*, яку можна побачити на рисунку 6.2.7. При необхідності користувач може задати свій числовий формат для нумерації елементів списку. Для цього в списку кнопки *Нумерація*, що розкривається, слід вибрати команду *Визначити новий*

числовий формат... В результаті на екрані з'являється діалогове вікно **Визначення нового числового формату**, представлене на рисунку 6.2.8. В цьому діалоговому вікні можна установити стиль нумерації, шрифт, формат номерів і спосіб вирівнювання, а також побачити результат у вікні **Попередній перегляд**.

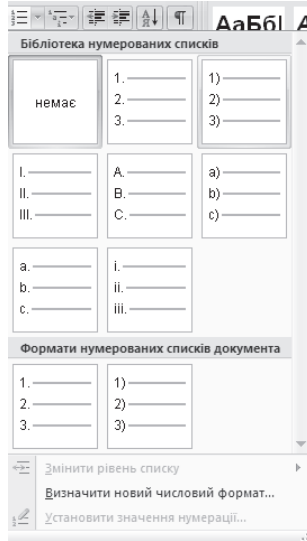


Рис.6.2.7. Бібліотека нумерованих списків

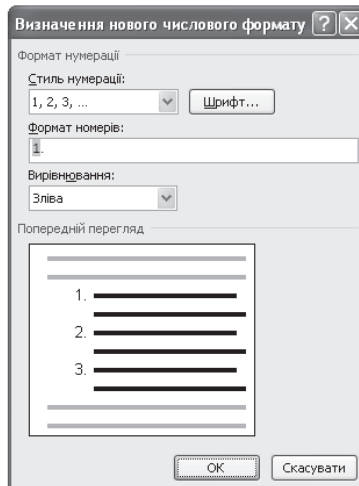



Рис.6.2.8. Ліалогове вікно **Визначення нового числового формату**

Створити багаторівневий список можна за допомогою однойменної кнопки , що розкривається. В результаті на екрані з'явиться список, представлений на рисунку 6.2.9.

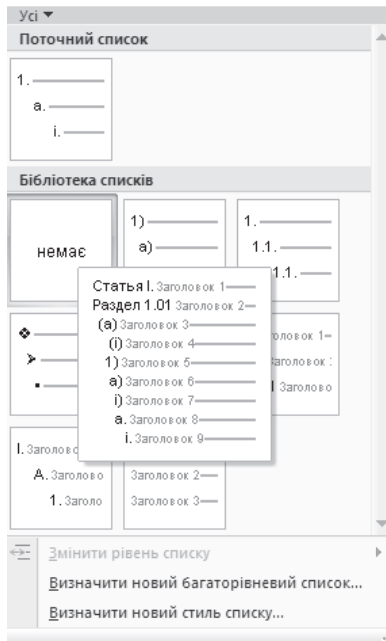


Рис.6.2.9. Список кнопки Багаторівневий список, що розкривається

Питання і завдання для самостійної роботи

Варіант 1. 1) Створити новий документ і встановити ширину полів: ліве поле – 2,2 см., праве – 1,5 см., верхнє – 1,5 см., нижнє – 1,5 см. Підготувати бланк учбової установи, в якій Ви навчаєтесь.

Варіант 2. 1) Створити новий документ і встановити ширину полів: ліве поле – 2,5 см., праве – 0,5 см., верхнє – 0,5 см., нижнє – 0,5 см. Підготуйте доповідну записку.

Варіант 3. 1) Створити новий документ і встановити ширину полів: ліве поле – 1,5 см., праве – 0,5 см., верхнє – 1,7 см., нижнє – 1,7 см. Підготуйте автобіографію.

Варіант 4. 1) Створити новий документ і встановити ширину полів: ліве поле – 2,8 см., праве – 1,5 см., верхнє – 2,3 см., нижнє – 2,2 см. Підготуйте заяву до суду.

Варіант 5. 1) Створити новий документ і встановити ширину полів: ліве поле – 1,9 см., праве – 0,25 см., верхнє – 1,2 см., нижнє – 1,2 см. Підготуйте договір комісії на здійснення операції з продажу майна.

Варіант 6. 1) Створити новий документ і встановити ширину полів: ліве поле – 1,27 см., праве – 1,7 см., верхнє – 1,23 см., нижнє – 1,25 см. Підготуйте договір комісії на реалізацію товарів і послуг.

Варіант 7. 1) Створити новий документ і встановити ширину полів: ліве поле – 3,2 см., праве – 0,3 см., верхнє – 2 см., нижнє – 1,5 см. Підготуйте договір комісії на покупку цінних паперів.

Варіант 8. 1) Створити новий документ і встановити ширину полів: ліве поле – 1,13 см., праве – 1,2 см., верхнє – 0,7 см., нижнє – 0,7 см. Підготуйте договір комісії з експорту.

Варіант 9. 1) Створити новий документ і встановити ширину полів: ліве поле – 2,3 см., праве – 2,3 см., верхнє – 0,9 см., нижнє – 0,5 см. Підготуйте заявку на участь в тендері.

Варіант 10. 1) Створити новий документ і встановити ширину полів: ліве поле – 2,52 см., праве – 0,52 см., верхнє – 1,25 см., нижнє – 1,25 см. Підготуйте договір постачання товару.

6.2.4. Створення графічних об'єктів засобами Microsoft Office Word 2007

У Microsoft Office Word 2007 є потужні інструменти як для вбудовування в текстовий документ готових зображень, так і для створення власних малюнків. При будь-якому способі створення і вбудовування в текст малюнків користувачеві зручно скористатися інструментами групи *Зображення* на вкладці *Вставлення*.

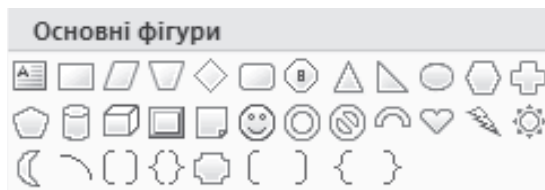
6.2.4.1. Використання інструмента Фігури для створення малюнка засобами Word 2007

Кнопка *Фігури*, що розкривається зі списком графічних примітивів, з яких створюються складніші малюнки, розташована в групі інструментів *Зображення* на вкладці *Вставлення* і містить наступні категорії:

- категорія *Лінії* містить різні види прямих і ламаних ліній із стрілками і без них



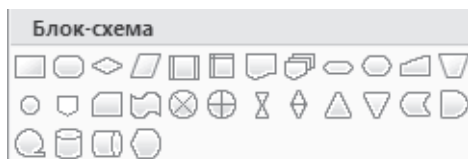
- категорія *Основні фігури* містить графічний об'єкт *Напис*, а також прямокутник (квадрат), паралелограм, трапецію, ромб, овал, багатогранник, куб, циліндр, різні види дужок і так далі



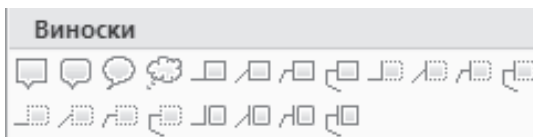
- категорія *Фігурні стрілки* містить стрілки різної форми і напрямку



- категорія *Блок-схема* містить різні елементи, використовувані при графічному способі представлення алгоритмів



- категорія *Виноски* містить виноски різної форми і напрямку



- категорія *Зірки та стрічки* містить зірки і стрічки різної форми



Щоб намалювати будь-яку з цих фігур, досить клацнути лівою кнопкою миші по кнопці із зображенням потрібної фігури, перемістити покажчик миші у вільну область листа і, утримуючи натиснутою ліву кнопку миші, розтягнути фігуру до потрібного розміру і форми. Після того, як фігура намальована до неї можна застосувати наступні операції (заздалегідь фігури треба виділити, клацнувши по ній лівою кнопкою миші):

- для масштабування фігури (зміни її розмірів) слід підвести покажчик миші або до синього квадрата для зміни розмірів по одній з осей – горизонтальній або вертикальній (при цьому покажчик миші є горизонтальною або вертикальною подвійною стрілкою), або до синього кружочка для зміни розміру по двох осях (при цьому покажчик миші є діагональною подвійною стрілкою) і, утримуючи натиснутою ліву кнопку миші, розтягнути або стискувати зображення. Приклад масштабування фігури наведений на рисунку 6.2.10.

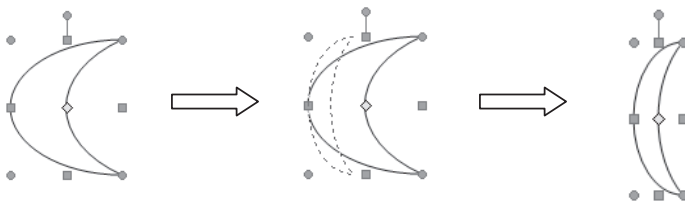


Рис.6.2.10. Приклад масштабування фігури Місяць

- для зміни зігнутої лінії слід підвести покажчик миші до жовтого ромба, при цьому покажчик миші перетвориться в білу трикутну стрілку і, утримуючи натиснутою ліву кнопку миші, змінити зігнутість лінії. Приклад зміни параметра фігури – зігнутість лінії – приведений на рисунку 6.2.11.

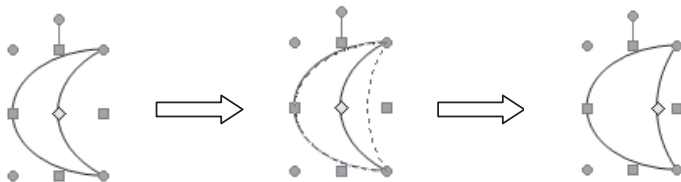


Рис.6.2.11. Приклад змінення зігнутої лінії фігури Місяць

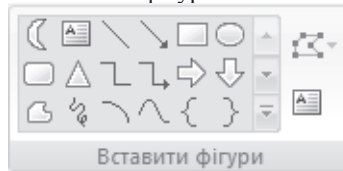
- для зміни кута повороту фігури слід підвести покажчик миші до зеленого кружечка, розташованого над верхньою часткою фігури, при цьому покажчик миші прийме вид петлеподібної чорної стрілки і, утримуючи натиснутою ліву кнопку миші, виконати поворот фігури на потрібний кут. Приклад виконання повороту фігури наведений на рисунку 6.2.12.



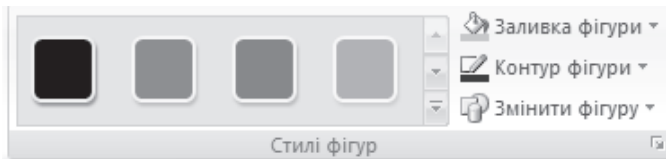
Рис.6.2.12. Приклад змінення кута повороту фігури

При виділенні фігури серед основних вкладок стрічкового інтерфейсу з'являється допоміжна вкладка *Формат (Засоби малювання)*, інструменти якої дозволяють встановити додаткові властивості фігури. Вкладка *Формат* містить 6 груп інструментів.

1. Група *Вставити фігури* призначена для вставки нової фігури в документ. Розкрити повний список фігур можна натисанням на кнопку



2. Група *Стили фігур* дозволяє установити спосіб заливання і тип контура фігури, змінити саму фігуру або скористатися готовим стилем зображення фігури. Розкрити повний список загальних візуальних стилів фігур можна натисненням на кнопку . Кнопки *Заливка фігури*, *Контур фігури*, *Змінити фігуру* містять списки, що розкриваються і дозволяють вибрати користувачеві колір заливки, тип і товщину лінії-контура фігури.



Приклад використання готового стилю наведений на рисунку 6.2.13.

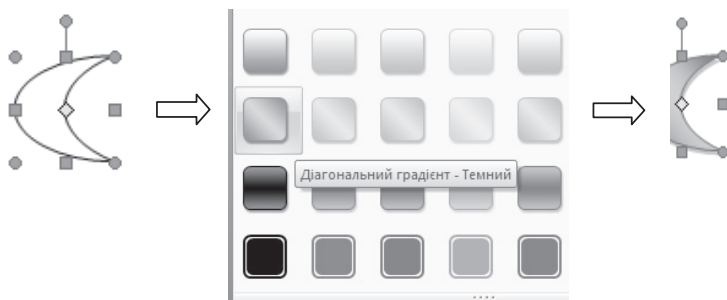


Рис.6.2.13. Приклад використання стиля Діагональний градієнт - Темний

Приклад зміни товщини і типа контура наведено на рисунку 6.2.14. Натисніть на кнопку *Контур фігури* і в списку, що розкривається, виберіть пункт Штрихи і ще в одному списку, що розкривається, виберіть пункт Круглі крапки.

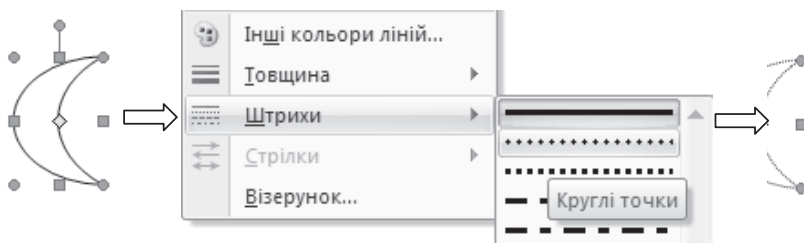


Рис.6.2.14. Приклад змінення накреслення контура з вибором типу штриха – Круглі точки

У списку кнопки *Контур фігури*, що розкривається виберіть пункт Товщина і ще в одному списку, що розкривається, слід вибрати підпункт 2,25 пт.

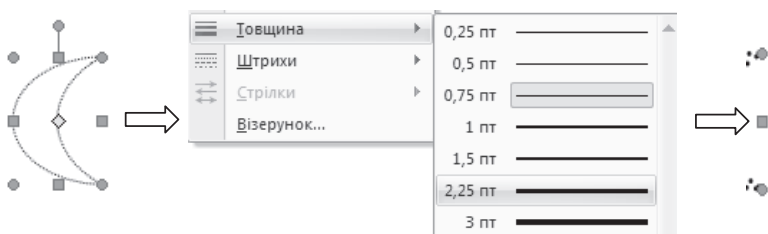

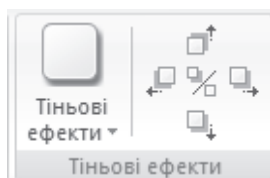


Рис.6.2.14. Приклад змінення товщині лінії з установкою товщини рівної 2,25 пт.

3. Група *Тіньові ефекти* дозволяє вибрати тип тіньового ефекту і установити його колір за допомогою кнопки *Тіньові ефекти*, що

розкривається, а за допомогою однієї з чотирьох кнопок  вибрати напрям переміщення тіні.



Приклад додавання тіньового ефекту наведений на рисунку 6.2.15.

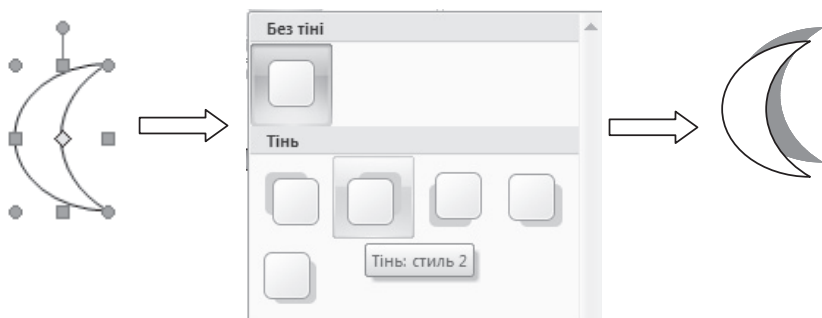
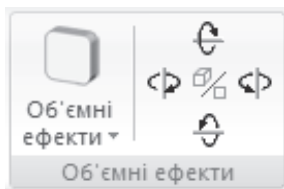


Рис.6.2.15. Приклад установки тіньового ефекту – Тінь: стиль 2

4. Група *Об'ємні ефекти* дозволяє зробити фігуру об'ємною (можливість доступна лише для деяких фігур), установити колір об'ємної фігури, глибину, напрям, кут освітлення, поверхню (матову, пластик, метал або каркас) за допомогою кнопки Об'ємні ефекти, що розкривається, а за

допомогою однієї з чотирьох кнопок  вибрати кут повороту фігури



Приклад використання об'ємного ефекту представлений на рисунку 6.2.16.

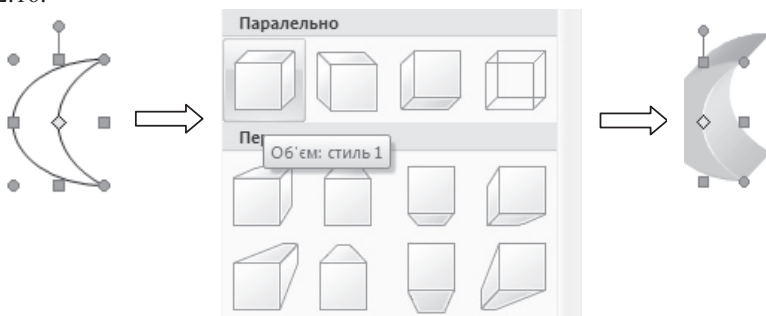
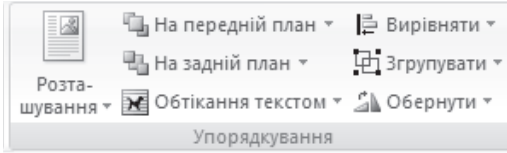
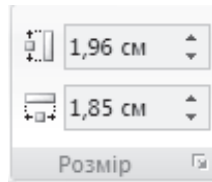


Рис.6.2.16. Приклад установки об'ємного ефекту – Об'єм: стиль 1

5. Група *Упорядкування* дозволяє розташувати фігуру на сторінці з установкою режиму обтікання тексту, згрупувати або розгрупувати сукупність фігур, вирівняти фігури по одному краю, обернути фігуру або дзеркально відображувати її по одній з осей.



6. Група *Розмір* дозволяє задати ширину і висоту фігури або з клавіатури або за допомогою перемикачів.



6.2.4.2. Використання інструменту Рисунок для вставлення рисунка (зображення) в документ Word 2007 з файлу

Вставити зображення, представлене у вигляді графічного файлу, можна за допомогою кнопки *Рисунок*, розміщеної в групі інструментів *Зображення* на вкладці *Вставлення*. В результаті на екрані з'явиться діалогове вікно *Вставлення рисунка*, в якому можна вказати шлях до графічного файлу, що містить зображення (малюнок). Після вибору графічного файлу слід натиснути на кнопку *ОК*. Рисунок з'явиться в тому місці документа, в якому заздалегідь був встановлений курсор. Діалогове вікно *Вставлення рисунка* представлено на рисунку 6.2.17.

Приклад вставленого з файлу зображення наведений на рисунку 6.2.18.



Рис.6.2.18. Приклад вставлення рисунка (зображення) із файлу

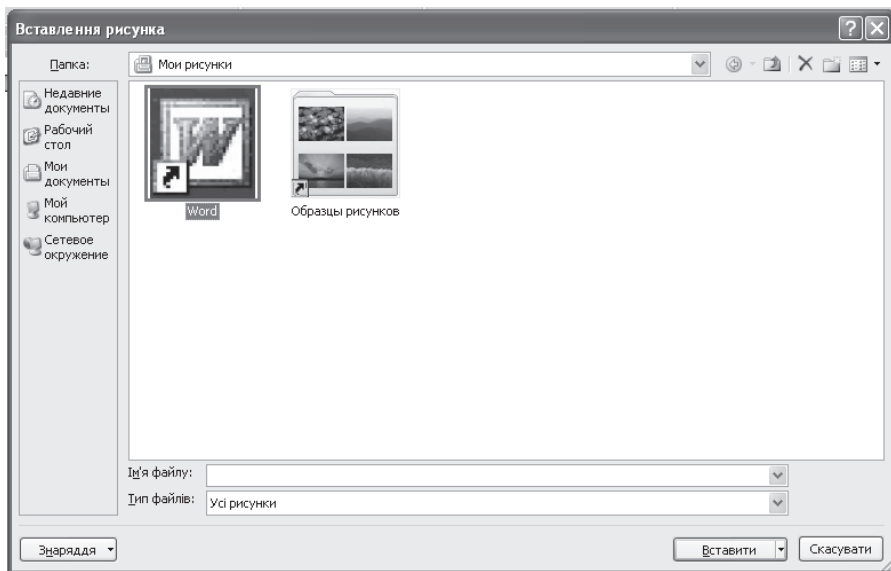
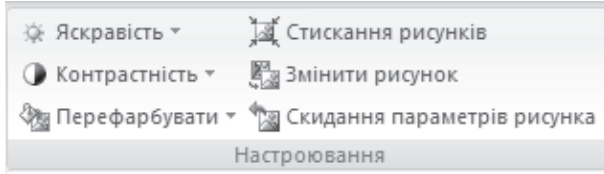



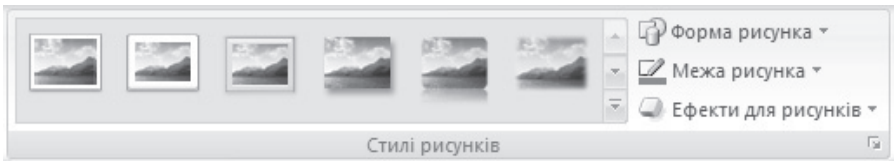
Рис.6.2.17. Діалогове вікно Вставлення рисунка

Для виділення рисунка слід клацнути по ньому лівою кнопкою миші. В результаті серед основних вкладок з'явиться допоміжна вкладка *Формат (Знаряддя для зображення)*, інструменти якої дозволяють встановити додаткові властивості рисунку (зображення). Вкладка *Формат* містить 4 групи інструментів.

1. Група *Настроювання* дозволяє за допомогою кнопки *Яскравість*, що розкривається, встановити яскравість (у відсотках); за допомогою кнопки *Контрастність*, що розкривається, встановити контрастність; за допомогою кнопки *Перефарбувати* перефарбувати рисунок, вибравши режим кольорів, темний або світлий варіант, встановити прозорий колір; за допомогою кнопки *Стискання рисунків* стискувати рисунок; за допомогою кнопки *Змінити рисунок* замінити вибраний рисунок новим; за допомогою кнопки *Скидання параметрів рисунка* відновити параметри рисунка за умовчанням.



2. Група *Стилі рисунків* дозволяє задати стиль рисунка (розкрити список стилей рисунка можна натисканням на кнопку ) , змінити форму і межу рисунка, задати ефект рисунка.



На рисунку 6.2.19 наведений приклад застосування до виділеного рисунка (зображення) одного із стилів.

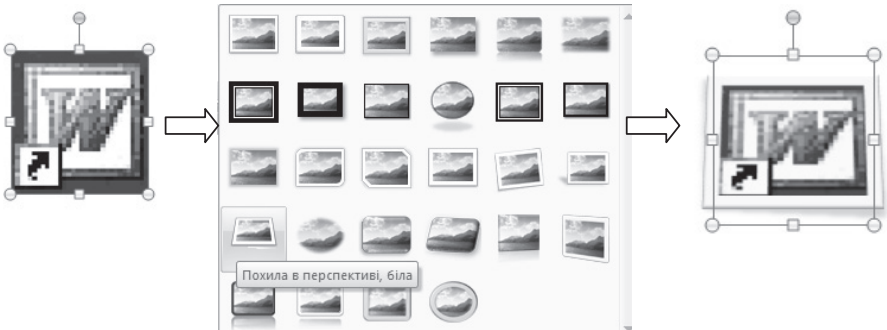


Рис.6.2.19. Приклад установки стиля рисунка – Похила в перспективі, біла

На рисунку 6.2.20 наведений приклад зміни форми малюнка за допомогою натиснення на кнопку *Форма рисунка*, що розкривається, і вибору відповідної форми.

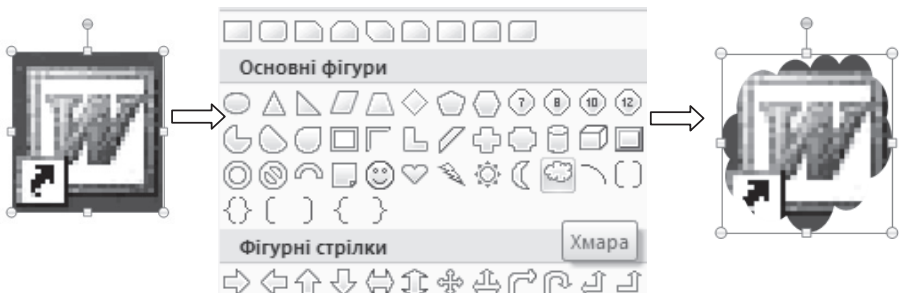
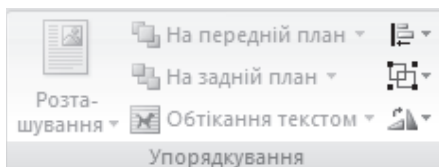
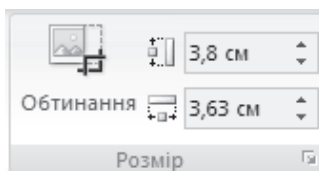


Рис.6.2.20. Приклад зміни форми рисунка

3. Група *Упорядкування* дозволяє розташувати рисунок (зображення) на сторінці з установкою режиму обтікання тексту, згрупувати або розгрупувати сукупність рисунків, вирівняти рисунки по одному краю, обернути рисунок або дзеркально відображувати його по одній з осей.



4. Група *Розмір* дозволяє задати ширину і висоту рисунка або з клавіатури або за допомогою перемикачів, а також виконати обрізання рисунка.



6.2.4.3. Використання інструмента SmartArt для вставлення схем в документ Word 2007

Вставити схему (рисунок) SmartArt можна за допомогою кнопки *SmartArt*, розміщеної в групі інструментів *Зображення* на вкладці *Вставка*. В результаті на екрані з'явиться діалогове вікно **Вибір рисунка SmartArt**, в якому необхідно вибрати потрібну схему в одній з категорій, перерахованих в лівій частині діалогового вікна (Усі, Список, Процес, Цикл, Структура, Зв'язок, Матриця, Піраміда). Діалогове вікно **Вибір рисунка SmartArt** представлено на рисунку 6.2.21.

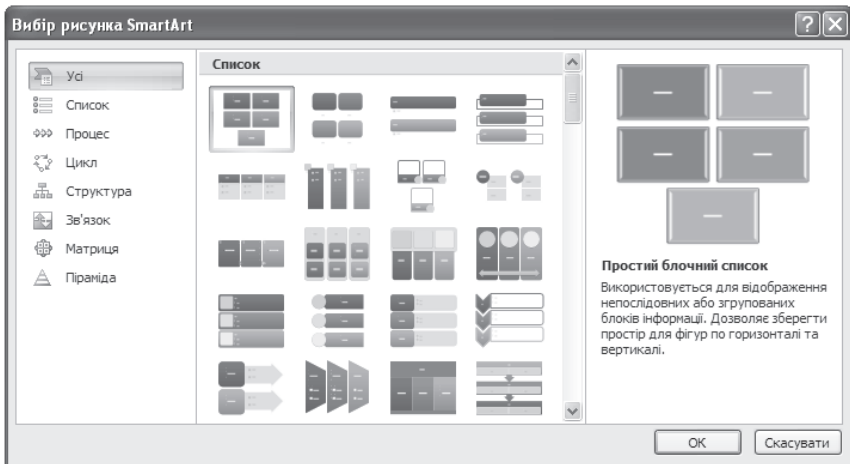


Рис.6.2.21. Діалогове вікно Вибір рисунка SmartArt

Наприклад, в розділі *Список* виберемо *Сегментований процес* і натиснемо на кнопку *ОК*. В результаті на екрані з'явиться об'єкт SmartArt, представлений на рисунку 6.2.22.

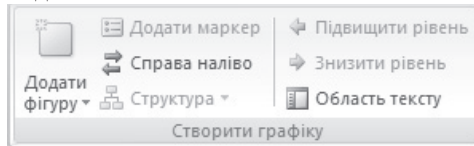
Поява будь-якого об'єкту SmartArt супроводиться активізацією допоміжної вкладки *Знаряддя для рисунків SmartArt* разом з тематичними вкладками *Конструктор* і *Формат*.



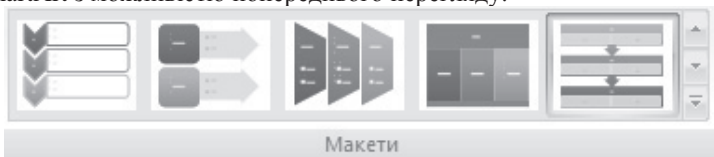
Рис.6.2.22. Об'єкт SmartArt Сегментований процес

Вкладка *Конструктор* містить 4 групи інструментів.

1. Група *Створити графіку* призначена для додавання фігур до об'єкту SmartArt, пониження (зниження) виділеного маркера або фігури, додавання текстового маркера до зображення, відображення або утаєння області тексту і тому подібне



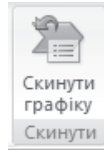
2. Група інструментів *Макети* призначена для легкої зміни макету об'єкту SmartArt з можливістю попереднього перегляду.



3. Група інструментів *Стилі SmartArt* призначена для зміни кольору або стилю графічного зображення SmartArt.



4. Група інструментів *Скинути* призначена для відміни всіх змін, внесених до формату зображення SmartArt.



6.2.4.4. Використання інструмента Діаграма для вставлення діаграм різних типів в документ Word 2007

Вставити діаграму в документ Word 2007 можна за допомогою кнопки *Діаграма*, розміщеної в групі інструментів *Зображення* на вкладці *Вставлення*. В результаті на екрані з'явиться діалогове вікно *Вставлення діаграми*, призначене для вибору типу потрібної діаграми і представлене на рисунку 6.2.23.

Опишемо коротко типи пропонуваніх діаграм.

1. **Гістограма** – прямокутна, об'ємна, конічна, з накопиченням.
2. **Графік** – діаграма по точках, із з'єднанням точок.
3. **Круг** – для представлення частин або частин одного цілого.
4. **Лінійчата** – гістограма з іншим розташуванням осей.
5. **З областями** – графіки із зафарбованою під ними областю.
6. **Точкова** – відображення інформації по точках із з'єднанням їх чи ні.
7. **Інші типи** – біржова, поверхня, кільце, бульбашкова, пелюсткова.

Після вибору типу діаграми на екрані з'явиться лист Excel, в якому можна змінювати дані (рисунок 6.2.24).

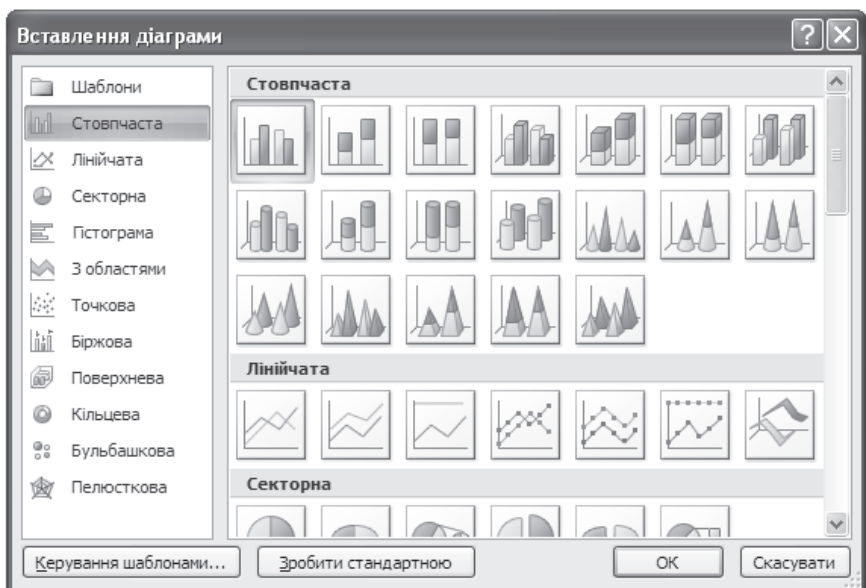


Рис.6.2.23. Діалогове вікно Вставка діаграми

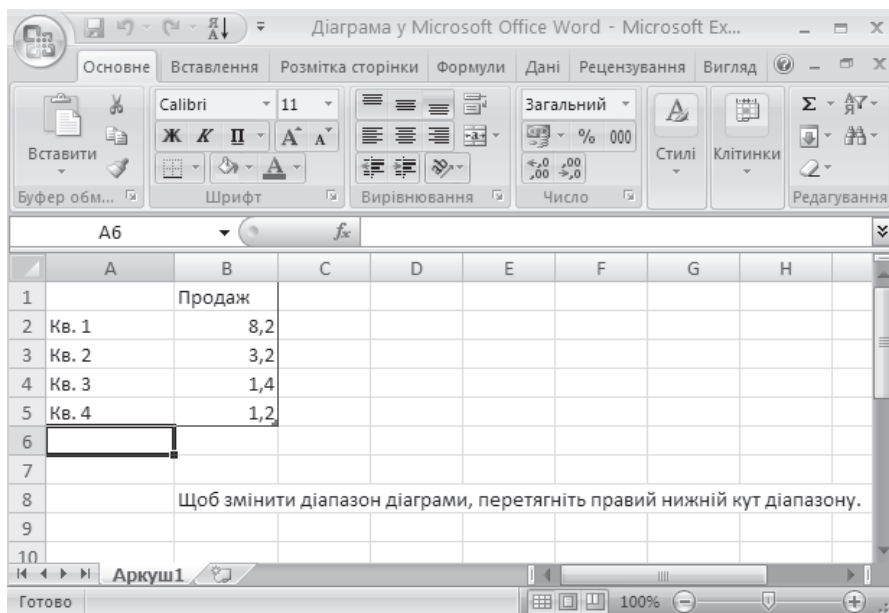


Рис.6.2.24. Лист Excel зі зразком даних

Після введення даних в документі MS Office Word 2007 з'явиться діаграма заданого типу, що відображує введені дані (рисунок 6.2.25). Активізація такої діаграми призведе до появи допоміжної вкладки *Знаряддя для діаграм* разом з тематичними вкладками *Конструктор*, *Макет*, *Формат*. Вкладка *Конструктор* складається з 4 груп інструментів: *Тип*, *Дані*, *Макети діаграм*, *Стилі діаграм*. При побудові діаграми на рисунку 6.2.25 вибраний *Макет1*.



Рис. 6.2.25. Приклад побудови секторної діаграми з використанням *Макета1*

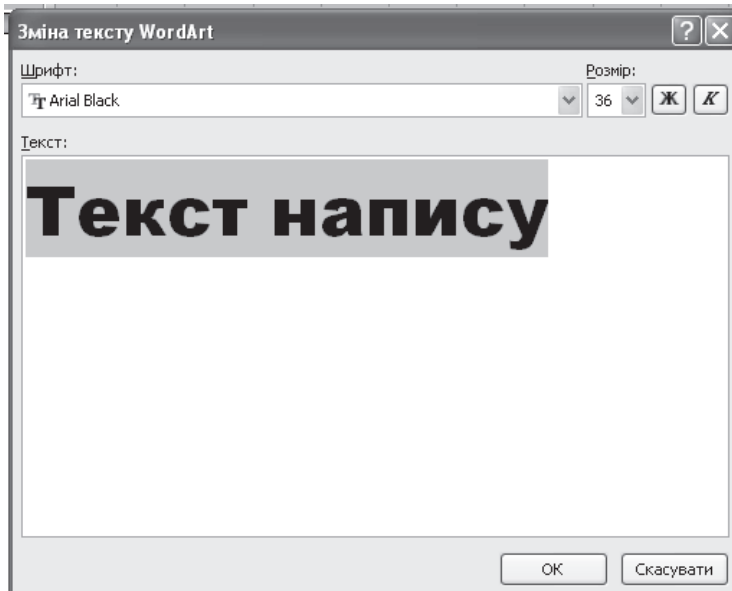
6.2.4.5. Використання інструмента *WordArt* для вставлення написів *WordArt* в документ Word 2007

Вставити напис в стилі *WordArt* в документ Word 2007 можна за допомогою кнопки *WordArt*, що розкривається, розміщеної в групі

інструментів *Текст* на вкладці *Вставка*. У списку (рисунок 6.2.26), що розкривається, клацанням лівої кнопки миші слід вибрати вподобаний напис в стилі WordArt. В результаті на екрані з'явиться діалогове вікно *Зміна тексту WordArt*, в якому можна задати шрифт, його розмір, зображення тексту (рисунок 6.2.27).



Рис.6.2.26. Стили напису WordArt з прикладом вибору стилю WordArt 9



Як приклад введемо наступні дані в діалоговому вікні *Зміна тексту WordArt*. Встановимо шрифт Arial, зображення Напівжирний, розмір 12 і введемо, натискуючи після кожного слова клавішу *ENTER*, слова в наступному порядку: економічна дисципліна інформатика. Після завершення всіх вказаних дій слід натиснути клавішу *OK*. Результат виконання перерахованих дій представлений на рисунку 6.2.28.

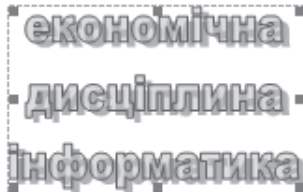
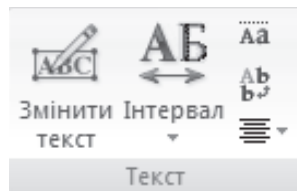


Рис.6.2.28. Приклад використання стилю WordArt 9 і введення даних в діалоговому вікні *Зміна тексту WordArt*

Якщо виділити напис в стилі WordArt, то серед основних вкладок з'являється контекстна (допоміжна) вкладка *Формат (Знаряддя для WordArt)*. Інструменти цієї вкладки дозволяють змінити параметри напису в стилі WordArt і містить 6 груп інструментів.

1. Група *Текст* призначена для зміни тексту, інтервалу між літерами тексту, завдання вирівнювання тексту і вертикального розташування тексту.



2. Група *Стили WordArt* дозволяє змінити стиль напису WordArt, заливку, контур напису і фігуру.



Змінимо заливку напису в стилі WordArt, а також колір контура. Для цього слід натиснути на кнопку *Заливка фігури*, що розкривається, і в списку, що розкрився, вибрати чорний колір, потім натиснути на кнопку *Контур фігури*, що розкривається, і в палітрі кольорів, що розкрилася, вибрати чорний колір.

Приклад зміни форми фігури з отриманим результатом представлений на рисунку 6.2.29. Для зміни форми фігури слід натиснути на кнопку *Змінити фігуру*, що розкривається, і в списку, що розкрився, вибрати траєкторію руху або викривлення.

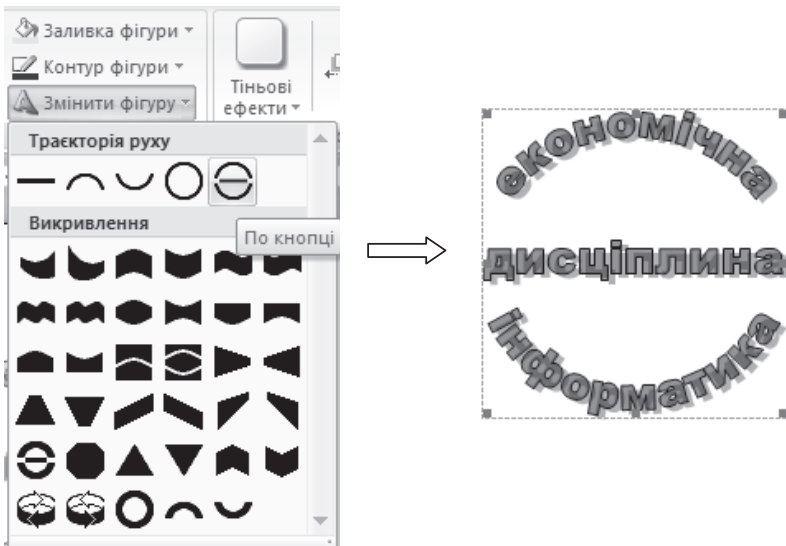


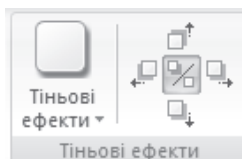
Рис.6.2.29. Приклад змінення форми напису в стилі Word Art (форма По кнопці)

3. Група *Тіньові ефекти* дозволяє вибрати тип тіньового ефекту і встановити його колір за допомогою кнопки *Тіньові ефекти*, що

розкривається, а за допомогою однієї з чотирьох кнопок



вибрати напрям переміщення тіні.



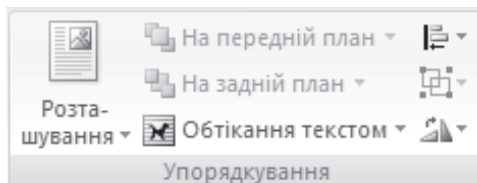
4. Група *Об'ємні ефекти* дозволяє зробити напис в стилі WordArt об'ємною, встановити колір об'ємного напису, глибину, напрям, кут освітлення, поверхню (матову, пластик, метал або каркас) за допомогою кнопки *Об'ємні ефекти*, що розкривається, а за допомогою однієї з чотирьох



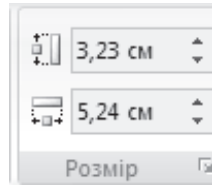
вибрати кут поворота напису.



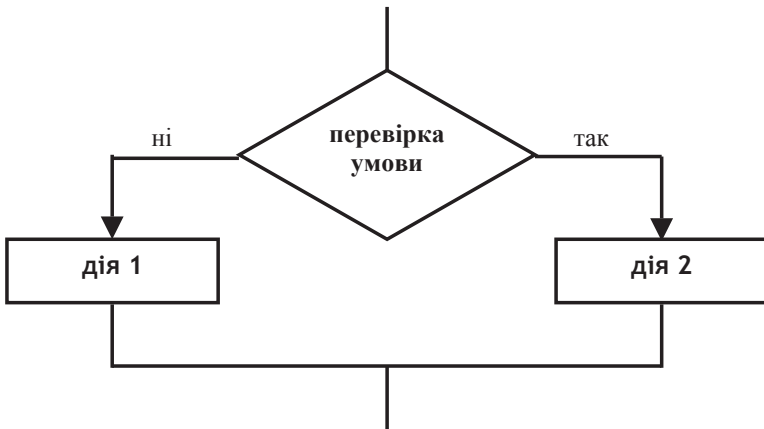
5. Група *Упорядкування* дозволяє розташувати напис в стилі WordArt на сторінці з установкою режиму обтікання тексту, згрупувати або розгрупувати сукупність написів, вирівняти написи по одному краю, обернути напис або дзеркально відображувати його по одній з осей.



6. Група **Розмір** дозволяє задати ширину і висоту напису в стилі WordArt або з клавіатури або за допомогою перемикачів.



Приклад. Створити фрагмент блок-схеми наступного вигляду:



Виконання завдання. Почнемо створення фрагмента блок-схеми з побудови ромба. Для цього слід скористатися кнопкою *Фігури*, що розкривається, на вкладці *Вставлення* в групі інструментів *Зображення* і вибрати ромб як показано на рисунку 6.2.30.

В результаті таких дій покажчик миші перетвориться в чорний тонкий хрест +. Слід у вільній області документа (там, де необхідно розмістити ромб) натиснути ліву кнопку миші і, утримуючи її, створити ромб необхідного розміру. Для зміни товщини лінії межі і установки потрібної ширини і довжини слід натиснути праву кнопку на зображенні ромба, що приведе до появи допоміжного меню, в якому слід вибрати пункт *Формат автофігури*. (рисунок 6.2.31).

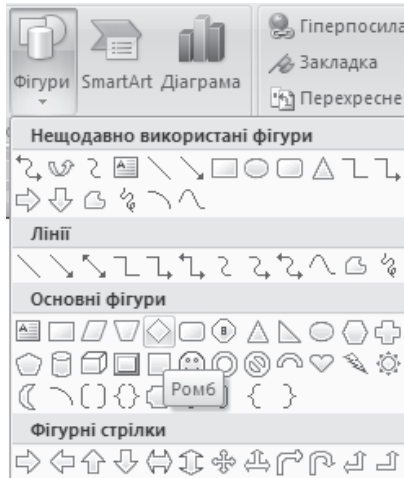


Рис.6.2.30. Список фігур, що розкривається

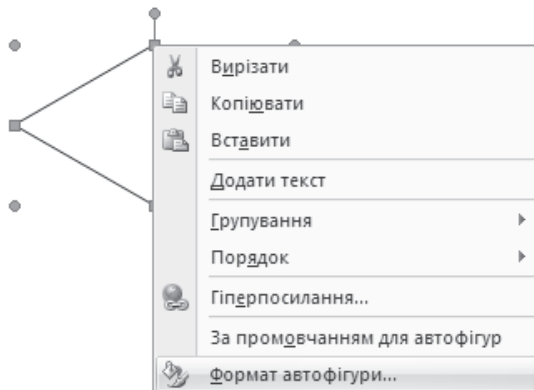


Рис.6.2.31. Допоміжне меню Формат автофігури

В результаті на екрані з'явиться однойменне діалогове вікно, на вкладці *Кольори та лінії* якого слід встановити товщину лінії 1,5 пт. Активна вкладка *Кольори та лінії* діалогового вікна **Формат автофігури** представлена на рисунку 6.2.32.

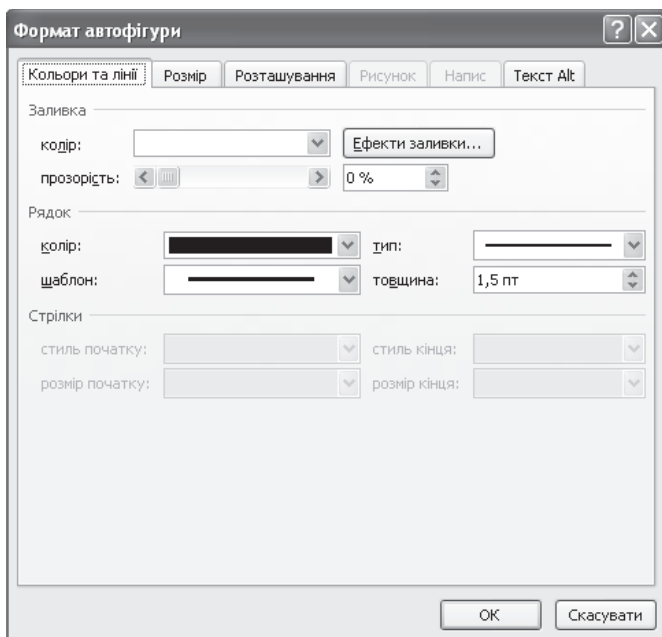


Рис.6.2.32. Допоміжне меню Формат автофігури з активною вкладкою Кольори та лінії

Для установки розмірів ромба слід перемкнутися на вкладку *Розмір*, прибрати прапорець *Зберегти пропорції* і встановити висоту 2,54 см, а ширину – 4,45 см. Після виконання всіх дій слід натиснути на кнопку *ОК*. Активна вкладка *Розмір* діалогового вікна **Формат автофігури** представлена на рисунку 6.2.23.

Для введення тексту усередині ромба слід натиснути на зображенні ромба правою кнопкою миші і в допоміжному меню, що з'явилося, вибрати пункт *Додати текст*. Після чого усередині ромба заблимає курсор і можна буде ввести текст – перевірка умови, заздалегідь встановивши шрифт Times New Roman, розмір 12 пт, креслення – напівжирний. Вибір пункту *Додати текст* приведений на рисунку 6.2.34.

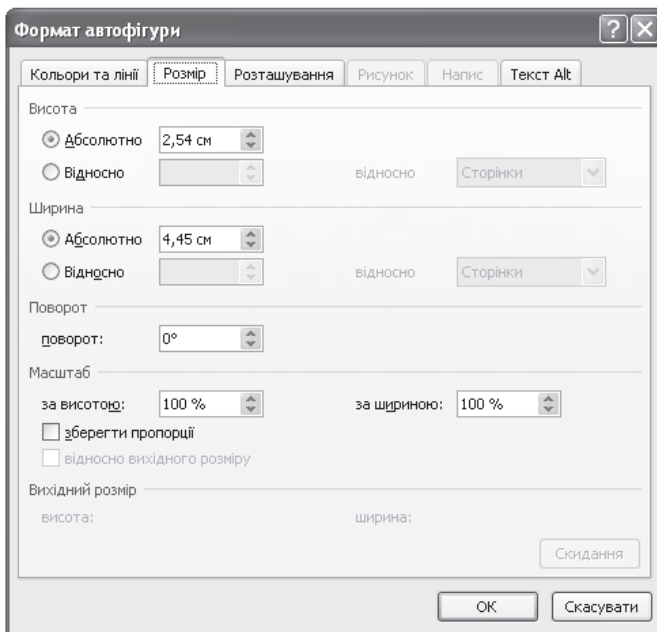


Рис.6.2.33. Допоміжне меню Формат автофігури з активною вкладкою Розмір

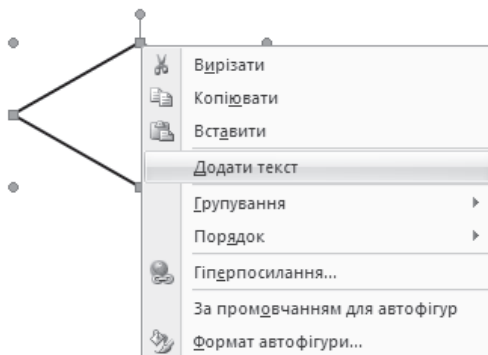






Рис.6.2.34. Пункт Додати текст допоміжного меню

Всі інші фігури слід створювати аналогічно, використовуючи для створення відрізків – кнопку *Лінія* , для створення стрілок – кнопку *Стрілка* , для створення прямокутників – кнопку *Прямокутник* , які можна вибрати в списку, що розкривається, натисканням на кнопку *Фігури*. В результаті отримаємо наступний фрагмент блок-схеми.

Для створення написів – «так» і «ні» – над відрізками можна підвести курсор до потрібного місця і ввести текст, а можна використати кнопку

Напис  із списку *Фігури*, що розкривається на вкладці *Вставлення* в групі інструментів *Зображення*. Розглянемо цю можливість докладніше. Для створення напису слід натиснути на кнопку *Напис*, підвести курсор до потрібного відрізка і натиснути ліву кнопку миші, а потім, утримуючи її, створити прямокутну область, усередині якої з'явиться курсор, після чого можна буде ввести текст. Результат дії представлено на рисунку 6.2.35.

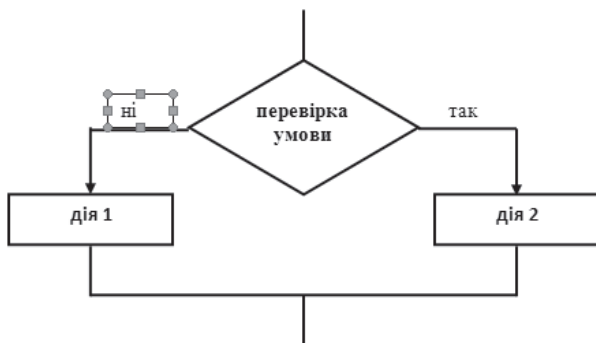


Рис.6.2.35. Процес створення напису

Для усунення чорного кордону навколо напису слід на межі прямокутної області, що обрамувала напис, натиснути праву кнопку миші і в

допоміжному меню, що з'явилось, вибрати пункт *Формат напису...*, після чого на екрані з'явиться однойменне діалогове вікно.



У діалоговому вікні *Формат напису* слід змінити колір лінії з чорного на білий як показано на рисунку 6.2.36. Для того, щоб область напису не перекривала відрізок необхідно на межі, що обрамляє напис, натиснути праву кнопку меню і в допоміжному меню вибрати пункт *Порядок* і його підпункт *На задній план*, як показано на рисунку 6.2.37.

Після введення всіх написів рекомендується згрупувати всі графічні об'єкти. Для цього треба виділити всі графічні об'єкти, натиснути ліву кнопку миші у вільній області екрану і, утримуючи її, «захопити» всі графічні об'єкти. Після всіх перерахованих дій слід натиснути праву кнопку миші на виділених об'єктах і вибрати в допоміжному меню пункт *Угрупувати* і підпункт *Групувати*, як показано на рисунку 6.2.38.

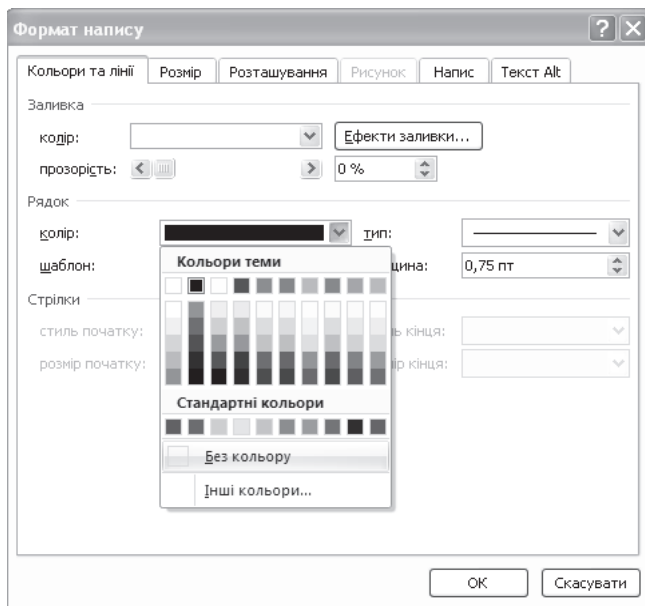


Рис.6.2.36. Діалогове вікно Формат напису з активною вкладкою Кольори та лінії

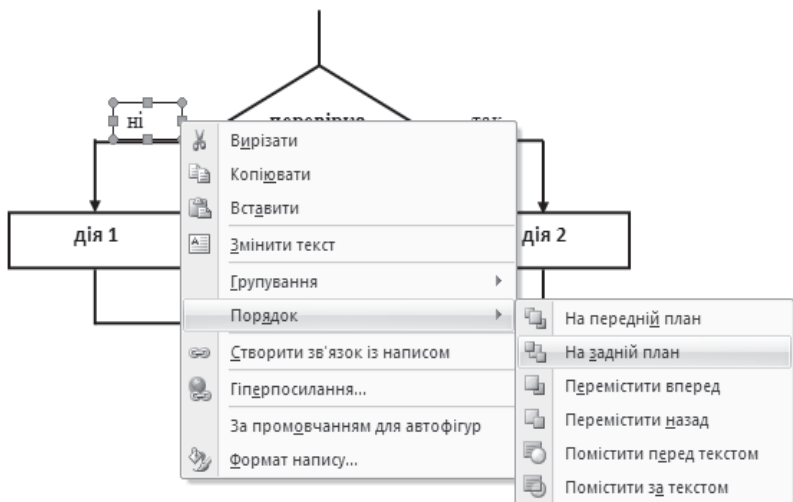


Рис.6.2.37. Змінення порядку розміщення об'єктів

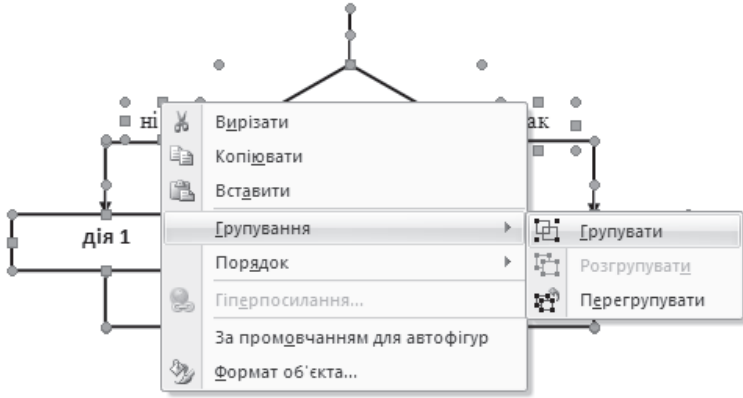
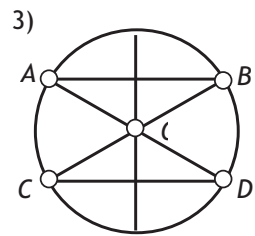
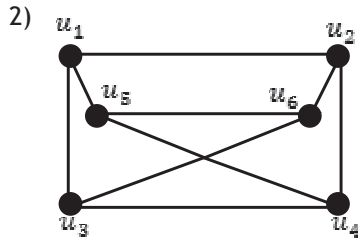
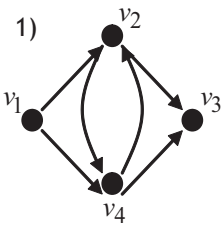


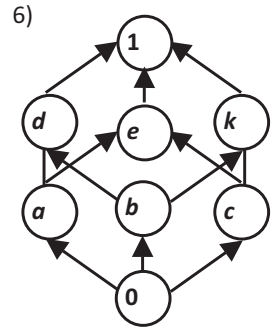
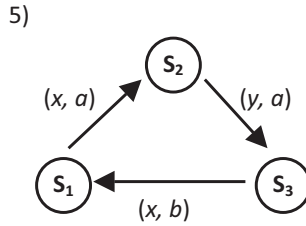
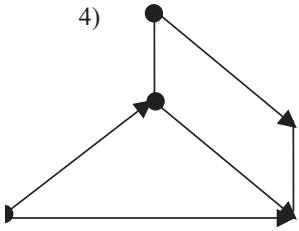
Рис.6.2.38. Виконання групування графічних об'єктів

Питання і завдання для самостійної роботи

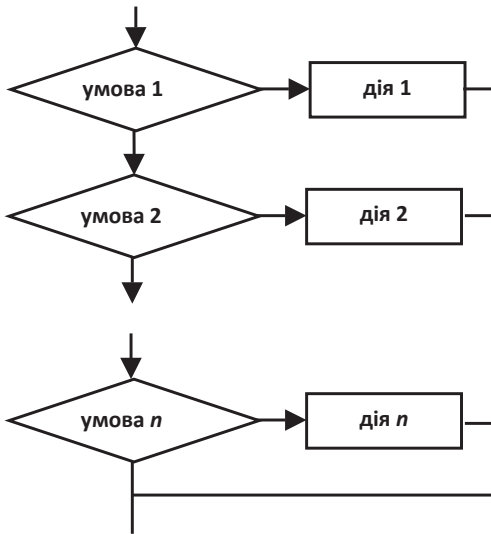
Використовуючи інструменти MS Office Word 2007, створити наступні рисунки (зображення)

Інструмент Фігури





7)



6.2.5. Введення математичних формул у документ Microsoft Office Word 2007

При написанні курсових і дипломних робіт або наукових статей студентам економічних і математичних спеціальностей доводиться вводити текст, що містить велику кількість математичних формул. Для спрощення

введення формул в Microsoft Word вбудований редактор формул (Equation Editor), а в Microsoft Office Word 2007 вбудований ще один зручний інструмент – конструктор формул.

Для того, щоб скористатися звичним для попередніх версій редактора Word способом введення формул, слід перейти на вкладку *Вставлення* і в групі інструментів *Текст* натиснути на кнопку *Об'єкт*. В результаті на екрані з'явиться діалогове вікно, в якому слід вибрати об'єкт *Microsoft Equation 3.0* (якщо цей модуль встановлений при установці програм пакету Office) і натиснути на кнопку *ОК* (рисунок 6.2.39).

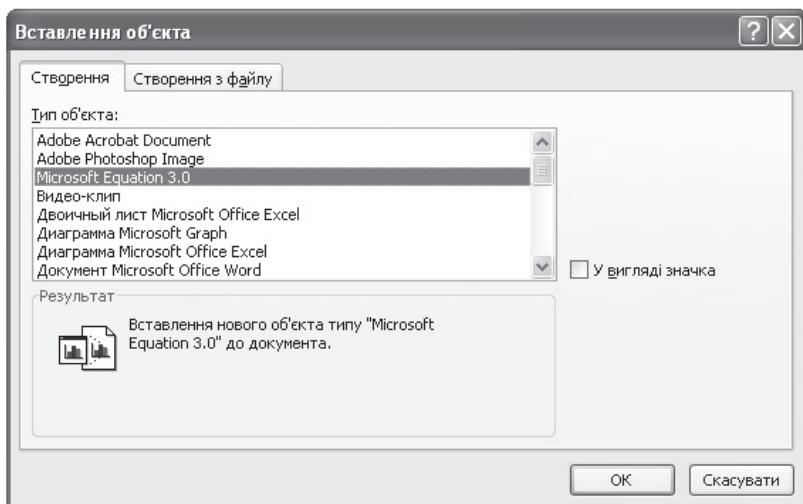


Рис. 6.2.39. Вставка об'єкта Microsoft Equation 3.0

На екрані з'явиться вікно введення формул в Microsoft Equation 3.0, представлене на рисунку 6.2.40.

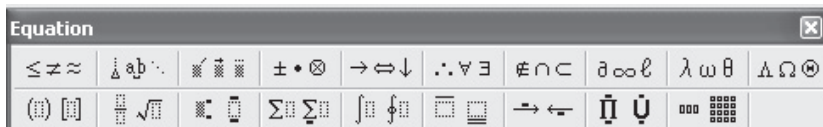














Рис 6.2.40. Вікно введення формул в Microsoft Equation 3.0


В робочій області екрана з'явиться спеціальне поле  з пунктирним прямокутником в його лівій частині, в якій потрібно вводити формулу. Відзначимо, що з допомогою панелі інструментів Equation (Формула) слід вводити ті символи, які неможливо ввести з клавіатури.


Опишемо призначення всіх кнопок панелі інструментів Equation (Формула). При натисненні на будь-яку кнопку верхньої стрічки розкривається палітра символів, відповідних назві кнопки. При натисненні на будь-яку кнопку нижньої стрічки виводиться палітра шаблонів, відповідна цій кнопці.

Кнопка *Символи відношень*  призначена для введення наступних символів відношень. При натисканні на цю кнопку з'являється панель з кнопками наступних символів відношень. Для введення будь-якого з цих символів слід натиснути на кнопку, що відповідає символу.



Вид кнопки та її призначення	Результат форматування
 знак нерівності «не більше» або «менше або рівно»	\leq
 знак нерівності «не менше» або «більше або рівно»	\geq
 відношення часткового порядку	\cap
 відношення часткового порядку	\cup
 група є нормальною підгрупою іншої групи	\triangleleft
 група є нормальною підгрупою іншої групи	\triangleright
 знак «не рівно»	\neq
 знак тотожної рівності	\equiv

 знак наближеної рівності	\approx
 пропорційність	\propto

Кнопка *Пробіли і три крапки*  призначена для введення наступних символів відносин. При натисканні на цю кнопку з'являється панель із кнопками наступних видів пробілів і три крапки. У формулі, створюваній за допомогою редактора формул, неможливо ввести пробіл натисканням клавіші пробілу. Для цього треба використовувати кнопки з панелі інструментів, що розкривається, при натисканні на кнопку *Пробіли і три крапки*. Для того, щоб продемонструвати розміри пробілів будемо вводити символи a і b із клавіатури, після введення літери a будемо натискати кнопку пробілу, установлюючи пробіл певного розміру, а потім слід ввести літеру b .

Кнопка *Надрядкові знаки*  призначена для введення представлених у таблиці надрядкових знаків. При цьому спочатку треба ввести літеру із клавіатури, наприклад літеру a , а потім натиснути на потрібну кнопку зі списку, що розкрився, який проявляється після натискання на кнопку *Надрядкові знаки*.

Наприклад, для того щоб ввести символ \tilde{a} треба виконати наступні дії:

- 1) відкрити редактор формул;
- 2) ввести літеру a в пунктирному прямокутнику, розміщеному в полі введення формул ;
- 3) вибрати кнопку *Надрядкові знаки* на панелі інструментів *Формула*;
- 4) в списку кнопок, що розкривається, треба натиснути на кнопку , що призведе до появи надрядкового знаку \sim («тільда») над літерою a




Розглянемо надрядкові знаки, які можна поставити над літерою a .

Вид кнопки та її призначення	Результат форматування
	a
	ϕ
	$a' \quad a'' \quad a'''$
	$\dot{a} \quad \ddot{a} \quad \overset{\cdot\cdot}{a}$
	\bar{a}
	\hat{a}
	\tilde{a}
	$\vec{a} \quad \overleftarrow{a} \quad \overrightarrow{a}$
	$\bar{\vec{a}} \quad \overleftarrow{\bar{a}}$
	$\widehat{\vec{a}} \quad \overrightarrow{\hat{a}}$
	$\`a$


Кнопка *Оператори* призначена для введення наступних математичних операторів. Для введення будь-якого з цих операторів необхідно натиснути на відповідну оператору кнопку розміщену на панелі, що розкривається.





Вид кнопки і її призначення	Результат форматування
	$\pm \quad \mp$
	$\times \quad \div$
	$*$
	$\circ \quad \bullet$
	$\otimes \quad \oplus$



Кнопка *Стрілки*  призначена для введення різних видів стрілок. Для введення будь-якої з цих стрілок необхідно натиснути на відповідну стрілку кнопку, розміщену на панелі, що розкривається.

Вид кнопки та її призначення	Результат форматування
  	\rightarrow \leftarrow \leftrightarrow
  	\uparrow \downarrow \updownarrow
  	\Rightarrow \Leftarrow \Leftrightarrow
  	\Uparrow \Downarrow \Updownarrow
 	\mapsto \lrcorner


Кнопка *Логічні операції*  призначена для введення квантора загальності і квантора існування, таких логічних операцій як кон'юнкція, диз'юнкція та інших. Для введення будь-якого символу логічної операції необхідно натиснути на кнопку, що відповідає цій операції.


Вид кнопки та її призначення	Результат форматування
	\therefore \because
	\exists
 квантор загальності	\forall
 квантор існування	\exists
 заперечення (інверсія)	\neg
 кон'юнкція або логічне складання	\wedge


Ψ диз'юнкція або логічне складання	\vee
--	--------


Кнопка *Символи теорії множин* $\notin \cap \subset$ призначена для введення основних операцій теорії множин таких, як об'єднання, перетинання, знак пустої множини тощо. Для введення будь-якого символу з теорії множин необхідно натиснути на кнопку, що відповідає цьому символу.


Вид кнопки та її призначення	Результат форматування
\in приналежність	\in
\cup об'єднання	\cup
\bigcup об'єднання	\bigcup
\subset міститься	\subset
\subseteq міститься або збігається	\subseteq
$\not\subset$ не міститься	$\not\subset$
\notin не належить	\notin
\cap Перетинання	\cap
\bigcap Перетинання	\bigcap
\supset Включає	\supset
\supseteq включає або рівно	\supseteq
\emptyset порожня множина	\emptyset


Кнопка *Різні символи*  призначена для введення символів, таких як знак нескінченності і т.і. Для введення кожного з цих символів необхідно натиснути на кнопку відповідного символу.


Кнопка *Грецькі букви (рядкові)*  призначена для введення всіх рядкових (менших) грецьких букв. Для введення кожної з букв грецького алфавіту необхідно натиснути на кнопку відповідного символу.


Кнопка *Грецькі букви (прописні)*  призначена для введення всіх рядкових (більших) грецьких букв. Для введення кожної з букв грецького алфавіту необхідно натиснути на кнопку відповідного символу.


Другий рядок панелі інструментів *Формула* призначена для введення шаблонів. *Шаблон* – це набір форматуваних символів. Кнопка *Шаблони дужок*  призначена для введення різних видів дужок. Для введення будь-якого виду дужок слід натиснути на відповідну до дужок кнопку й у пунктирному прямокутнику, що з'явився, ввести потрібне математичне вираження.

Кнопка *Шаблони дробів і радикалів*  призначена для введення різних видів дужок. Для введення будь-якого виду дужок слід натиснути на відповідну до дужок кнопку й у пунктирному прямокутнику, що з'явився, ввести потрібне математичне вираження.


Кнопка *Шаблони верхніх і нижніх індексів*  призначена для введення різних видів дужок. Для введення будь-якого виду дужок слід натиснути на відповідну до дужок кнопку та у пунктирному прямокутнику, що з'явився, ввести потрібне математичне вираження.


Кнопка *Шаблони сум*  призначена для введення різних видів сум. Для введення будь-якого виду суми слід натиснути на відповідну до суми кнопку та у пунктирному прямокутнику, що з'явився, ввести потрібне математичне вираження.


Кнопка *Шаблони інтегралів*  призначена для введення різних видів інтегралів. Для введення будь-якого виду інтеграла слід натиснути на відповідну до інтегралу кнопку та у пунктирному прямокутнику, що з'явився, ввести потрібне математичне вираження.

Кнопка *Шаблони надкреслення і підкреслення*  призначена для введення різних видів підкреслення. Для введення будь-якого виду підкреслення слід натиснути на відповідну до типу підкреслення кнопку та у

пунктирному прямокутнику, що з'явився, увести потрібне математичне вираження.

Кнопка *Шаблони стрілок з текстом*  призначена для введення різних видів стрілок з написами над або під стрілкою. Для введення будь-якого виду стрілки з написом слід натиснути на відповідну до стрілки кнопку та у пунктирному прямокутнику, що з'явився, увести потрібне математичне вираження.

Кнопка *Шаблони добутків і символів теорії множин*  призначена для введення різних видів добутків і символів об'єднання і перетинання. Для введення будь-якого виду добутку, об'єднання або перетинання слід натиснути на відповідну до цієї операції кнопку та у пунктирному прямокутнику, що з'явився, увести потрібне математичне вираження.

Кнопка *Шаблони матриць*  призначена для введення різних видів матриць і систем рівнянь або нерівностей. Для введення будь-якого виду матриці або системи слід натиснути на відповідну до матриці або системі кнопку й у пунктирному прямокутнику, що з'явився, увести потрібне математичне вираження.

Вибір шаблону матриці приводить до появи діалогового вікна *Матриця*, в якому можна задати вид вирівнювання (по лівому краю, по правому краю, по центру), число стовпців і рядків (рисунок 6.2.41).

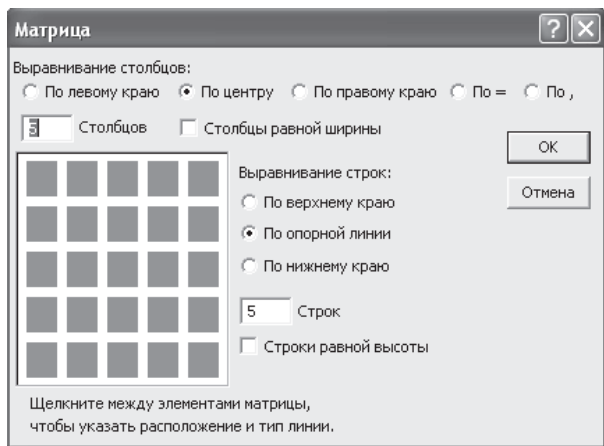



Рис. 6.2.41 Діалогове вікно Матриця

В Microsoft Office Word 2007 вбудований більш зручний ніж Microsoft Equation 3.0, інструмент для введення формул, запустити який можна за допомогою перемикавання на вкладку *Вставка* й натискання кнопки *Рівняння*  у групі *Символи*. У результаті в тому місці тексту, де встановлений курсор з'явиться місце для введення формули, представлене на рисунку 6.2.42.

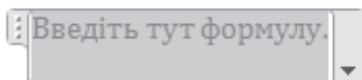
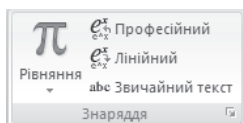


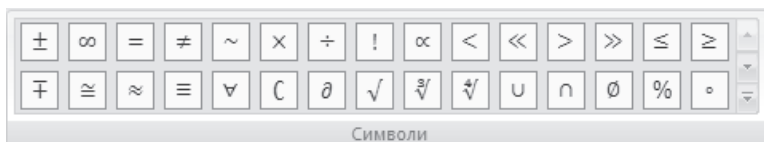
Рис. 6.2.42. Місце для введення

При цьому автоматично з'явиться допоміжна вкладка **Конструктор**, робота з інструментами якої аналогічна роботі з редактором формул Microsoft Equation 3.0. Перерахуємо групи інструментів Конструктора:

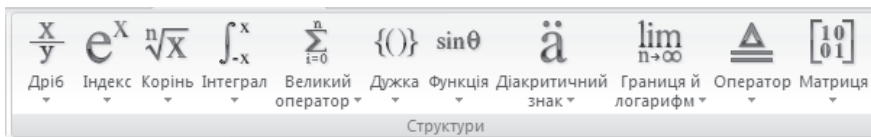
- група *Знаряддя* містить інструменти для вставки стандартних математичних формул (біном Ньютона, формули для обчислення коріння квадратного рівняння, площі круга, розкладання в ряди Тейлора і Фур'є, тригонометричні формули і тому подібне);



- група *Символи* містить інструменти для введення різних символів (основні математичні знаряддя, літероподібні символи, оператори, стрілки, відношення із запереченням, букви, геометрія);




- група *Структура* містить інструменти для введення дробів, індексів, коріння, інтегралів, дужок, функцій, діакритичних знаків, границь і логарифмів, операторів і матриць.






Розглянемо детальніше кожен з груп.

Група *Знаряддя* містить три кнопки для управління видом символів у формулі:

- кнопка *Професійний*  призначена для перетворення формул до готового для друку виду;

- кнопка *Лінійний*  призначена для перетворення формули до виду, зручного для редагування;

- кнопка *Звичайний текст*  призначена для перетворення використаних у формулі символів у звичайний текст.

Група *Символи* містить багату палітру символів, розкриття якої проводиться за допомогою натискання кнопки Додатково . Палітра символів, що розкривається, представлена на рисунку 6.2.43. Поруч із назвою палітри розташована кнопка  для розкриття списку палітр. Натиснувши на цю кнопку, слід установити прапорець напроти потрібної палітри, і вона з'явиться на екрані. Уведення символу здійснюється одинарним клацанням лівої кнопки миші по зображенню потрібного символу. На рисунках 6.2.44-6.2.51 показані доступні палітри.

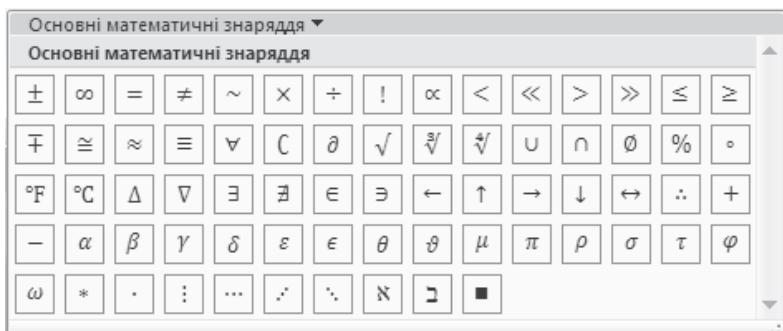


Рис. 6.2.43. Розкриття палітри Основні математичні знаряддя

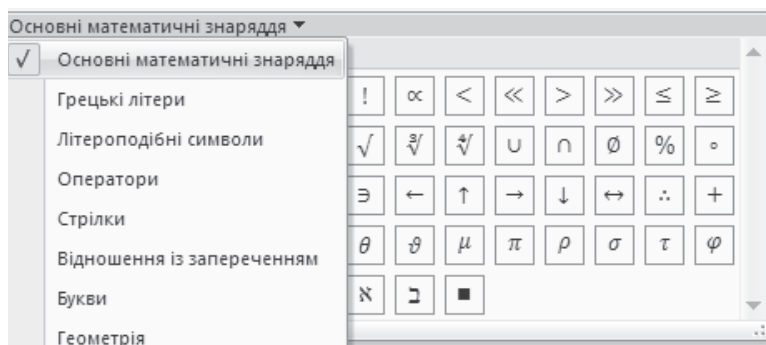


Рис. 6.2.44. Розкриття списку палітр



Рис. 6.2.45. Палітра символів Грецькі літери



Рис. 6.2.46. Палітра символів Літероподібні символи



Рис. 6.2.47. Палітра символів Стрілки


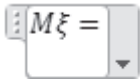

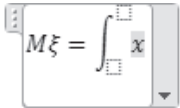


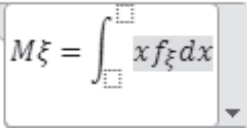


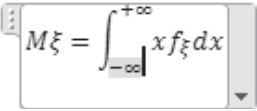
Рис. 6.2.48. Палітра символів Оператори

Завдання №.1. Засобами Microsoft Office Word 2007 ввести формулу $M\xi = \int_{-\infty}^{+\infty} x f_{\xi}(x) dx$.

Виконання. Розглянемо покрокове введення формули. При введенні формули будемо використовувати групи інструментів *Конструктора*.

Натиснемо кнопку *Рівняння* .

Група, інструмент	Дія	Результат
Група <i>Символи</i> , підгрупа <i>Грецькі літери</i>	Введемо з клавіатури букву M , а потім в підгрупі Грецькі літери натиснемо кнопку  . Після чого введемо з клавіатури знак рівності	
Група <i>Структури</i> , кнопка <i>Інтеграл</i> , що розкривається	Виберемо шаблон інтегралу  і введемо підінтегральне вираження натисканням правої кнопки миші усередині пунктирного прямокутника, що відповідає підінтегральному вираженню. Спочатку введемо з клавіатури x	
Група <i>Структури</i> , кнопка <i>Індекс</i> , що розкривається	Виберемо шаблон індексу –  . Нижній індекс  . Введемо в великому пунктирному прямокутнику букву f , а в пунктирному прямокутнику, що відповідає нижньому індексу введемо ξ так, як було описано вище. Потім натиснемо клавішу зсув на одну позицію вправо, щоб «вийти» з нижнього індексу і введемо з клавіатури dx .	
Група <i>Символи</i> , підгрупа <i>Основні математичні</i>	Натиснемо праву кнопку миші в пунктирному прямокутнику, що відповідає верхній межі інтегрування. Введемо з	

зряддя	клавіатури знак +. Перейдемо до підгрупи Основні математичні зряддя і натиснемо на кнопку ∞ . Аналогічно слід ввести нижню межу інтегрування.	
--------	--	---

Питання і завдання для самостійної роботи

За допомогою вбудованих в Microsoft Office Word 2007 інструментів для введення математичних формул введіть наступні формули:

$$1) D\xi = \int_{+\infty}^{+\infty} (x - M\xi)^2 f\xi dx$$

$$2) \sum_{i=0}^{n-1} x_i p_i = \sum_{i=0}^{n-1} x_i f\xi(x_i) \Delta x_i$$

$$3) f\xi(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(x-m)^2}{2\sigma^2}}$$

$$4) \lim_{n \rightarrow \infty} P\left(\frac{\xi_1 + \xi_2 + \dots + \xi_n - nM}{\sigma - \sqrt{n}} < x\right) = \Phi(x)$$

$$5) \lim_{n \rightarrow \infty} P\left(\left|\hat{F}_n(x) - F(x)\right| < \varepsilon\right) = 1$$

$$6) -\frac{\pi}{4} < \int_0^{+\infty} \frac{\cos ax}{x^2 + 4} dx < \frac{\pi}{4}$$

$$7) 0 < \int_2^{+\infty} e^{-x^2} dx < \frac{1}{4e^4}$$

$$8) \int_{-\infty}^{+\infty} x^n e^{-(x^2 + 1/x^2)} dx$$

$$9) \int_0^{+\infty} \frac{x dx}{1+x^2 \sin^2 x}$$

$$10) \lim_{x \rightarrow +0} \int_x^1 \frac{\cos t}{t^2} dt$$

$$11) \int_1^{+\infty} \frac{\ln^2 t}{t} \cos 2t dt$$

$$12) \lim_{x \rightarrow +0} \frac{\cot x}{\left(\frac{1}{2} - \lambda\right) x^{-\lambda - 1/2}}$$

$$13) \zeta(\lambda) = \int_1^{+\infty} \frac{\lambda [x]}{x^{\lambda+1}} dx = \lim_{n \rightarrow \infty} \left(1 + \frac{1}{2^\lambda} + \dots + \frac{1}{n^\lambda}\right)$$

$$14) \int_0^1 \frac{\cos^{-1} x}{\ln x} dx = -\sum_{k=0}^{\infty} \frac{(2k-1)!! \ln(2k+2)}{2^k k} \frac{1}{2k+1}$$

$$15) \int_0^{\infty} \cos bx \tan^{-1} \frac{a}{x^2} dx = -\frac{\pi}{b} e^{-b\sqrt{a/2}} \sin\left(b\sqrt{\frac{a}{2}}\right)$$

$$16) \sum_{k=1}^n (-1)^{k-1} \prod_{m=0}^{k-1} \sin \frac{2m+1}{2k} \pi = \frac{2}{3} + \frac{(-1)^n}{3 \cdot 2^{n-1}}$$

$$17) \sum_{k=1}^{\infty} a^{\sqrt{k}} x^k = \frac{x \ln a}{2\sqrt{\pi}} \int_0^{\infty} \frac{1}{(x - e^{-t})^{3/2}} e^{-(\ln^2 a)/(4t)} dt$$

$$18) \int_0^{\pi} (1 - 2a \cos x + a^2)^n dx = \pi \sum_{k=0}^n \binom{n}{k}^2 a^{2k}$$

$$19) \sum_{k=1}^t \sum_{j=k}^{\infty} F(j) = \sum_{k=1}^{\infty} a(k) \sum_{j=1}^{\min(t,k)} j \leq \frac{1}{2} g! \max_{1 \leq k \leq t} \frac{k(k+1)}{G(k)}$$

$$20) \lim_{x \rightarrow \pi} \frac{\sqrt{1 - \tan x} - \sqrt{1 + \tan x}}{\sin 2x}$$

$$21) A(x)B(x) = \left(\sum_{n=0}^{\infty} a_n x^n \right) \cdot \left(\sum_{n=0}^{\infty} b_n x^n \right) = \sum_{n=0}^{\infty} c_n x^n \text{ де } c_n = \sum_{k=0}^n a_k b_{n-k}$$

$$22) \begin{pmatrix} 2 & 0,3 & 0,5 \\ 0,1 & 3 & 0,4 \\ 0,1 & 0,1 & 0,8 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 2,9 \\ -4,7 \\ 14,3 \end{pmatrix}$$

$$23) \frac{P_*(\alpha)}{P_*(0)} = \begin{cases} \frac{c^\alpha \hat{P}_*^\alpha}{a!}, \alpha = \overline{0, c-1}, \\ \frac{c^c \hat{P}_*^\alpha}{c!}, \alpha \geq c. \end{cases}$$

$$24) F = -2x_1 + 3x_2 \rightarrow \max \begin{cases} 2x_1 + 3x_2 \geq 12, \\ -x_1 + x_2 \leq 7, \\ 2x_1 + x_2 \leq 10, \\ x_2 \geq 2, \end{cases} \quad x_1 \geq 0.$$

$$25) [ab] = \begin{bmatrix} \vec{i} & \vec{j} & \vec{k} \\ a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{bmatrix} = \hat{i}(a_2 b_3 - a_3 b_2) - \hat{j}(a_1 b_3 - a_3 b_1) + \hat{k}(a_1 b_2 - a_2 b_1)$$

6.2.6. Створення таблиць різного ступеня складності в документах Microsoft Office Word 2007

Одним із способів наочного представлення інформації є її розміщення у вигляді таблиці. Економісти, фінансисти та інші офісні працівники використовують складні таблиці, у вічках яких можна вводити формули і виконувати автоматичні обчислення. Складні таблиці такого роду можна створювати в програмі Microsoft Office Excel 2007 офісного пакету Microsoft Office 2007, функціональні можливості якої ми вивчимо в наступному розділі цього підручника.

Для створення таблиці слід перейти на вкладку *Вставлення* і натиснути на кнопку *Таблиця*, що розкривається, групи інструментів *Таблиці*. Список, що розкрився, визначає всілякі способи створення таблиць, представлений на рисунку 6.2.52.

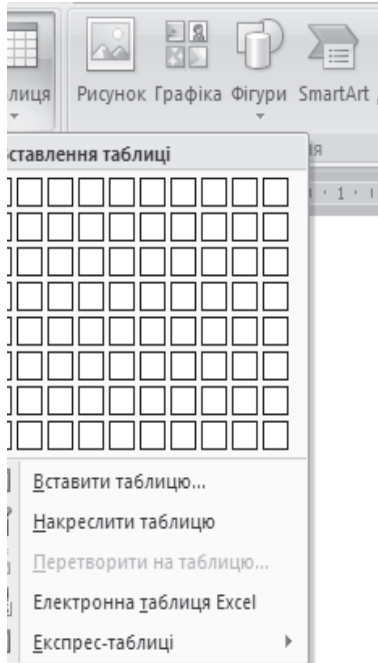


Рис.6.2.52. Список способів створення таблиць в MS Office Word 2007

1. Створення таблиці за допомогою пункту *Вставка таблиці*. Цей спосіб зручно використовувати, якщо таблиця, яку необхідно створити, містить не більше 8 рядків і не більше 10 стовпців. Для цього досить переміщати покажчик миші, вибираючи потрібне число рядків і стовпців. На рисунку 6.2.53 показана побудова таблиці, що складається з 5 рядків і 4 стовпців. Після того, як у верхньому рядку з'явиться напис Таблиця 4x5 слід клацнути лівою кнопкою миші по нижньому правому вічку виділеної області. В процесі вибору потрібного числа рядків і стовпців таблиця буде промальовуватися в тому місці документа, де був встановлений курсор.

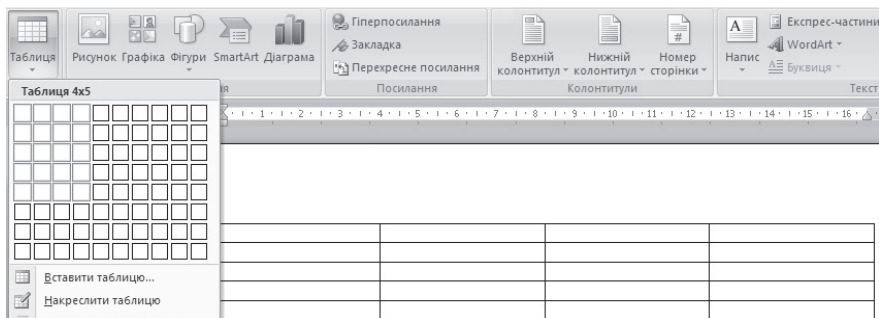


Рис.6.2.53. Приклад створення таблиць за допомогою пункту Вставлення таблиці із списку кнопки Таблиці, що розкривається

2. Створення таблиці за допомогою пункту *Вставити таблицю...* Цей спосіб можна використовувати, якщо число рядків створюваної таблиці більше 8 або число стовпців більше 10. Після вибору пункту *Вставити таблицю* на екрані з'являється діалогове вікно **Вставлення таблиці**, в якому слід вказати потрібне число рядків і стовпців створюваної таблиці, а також при необхідності вибрати спосіб автопідбору ширини стовпців (постійна, за вмістом, за шириною вікна). Після завдання всіх налаштувань слід натиснути на кнопку *ОК*, підтвердивши свій вибір. Після чого в тому місці, де був встановлений курсор, з'явиться таблиця. На рисунку 6.2.54 представлено діалогове вікно **Вставлення таблиці**.

3. Створення таблиці за допомогою пункту *Накреслити таблицю*. Після вибору пункту *Накреслити таблицю* покажчик миші прийме вигляд олівця. У місці, де необхідно вставити таблицю, слід натиснути ліву кнопку миші і, утримуючи її, накреслити межі майбутньої таблиці. Відпустіть ліву кнопку миші і переведіть покажчик миші до того місця таблиці, в якому необхідно створити рядок (провести горизонтальну лінію). Утримуючи ліву кнопку миші, накресліть стільки горизонтальних і вертикальних ліній, скільки необхідно. Наприклад, для створення 5 рядків необхідно накреслити 4 горизонтальних лінії, а для створення 4 стовпців – 3 горизонтальних лінії. Створення таблиці цим способом подібно до того, як креслять таблицю на листі паперу за допомогою олівця і лінійки.

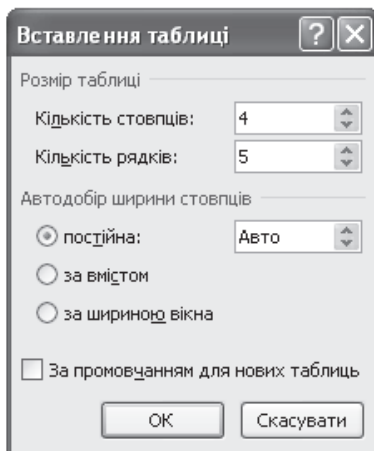


Рис.5.2.54. Приклад створення таблиці за допомогою пункту Вставити таблицю... із списку кнопки Таблиці, що розкривається

4. Створення таблиці за допомогою пункту *Перетворити на таблицю*.

5. Створення таблиці за допомогою пункту *Електронна таблиця Excel*. При виборі цього пункту немає необхідності вказувати число рядків і стовпців. У документ буде вбудована довільна таблиця Excel, ширину рядків і стовпців якої можна змінювати, коли курсор миші набуває вигляду подвійної чорної стрілки. Відбувається це в той момент, коли курсор миші підведений до межі двох рядків або стовпців. Утримуючи ліву кнопку миші, можна встановити потрібну ширину стовпця або рядка. Після цього ліву кнопку миші слід відпустити. При роботі з такою таблицею у вікні Word з'явиться стрічка інструментів Microsoft Office Excel 2007, за допомогою якої можна не лише вводити дані у клітинки таблиці, але і проводити обчислення за допомогою функцій Excel. Приклад вставки такої таблиці наведений на рисунку 6.2.55.

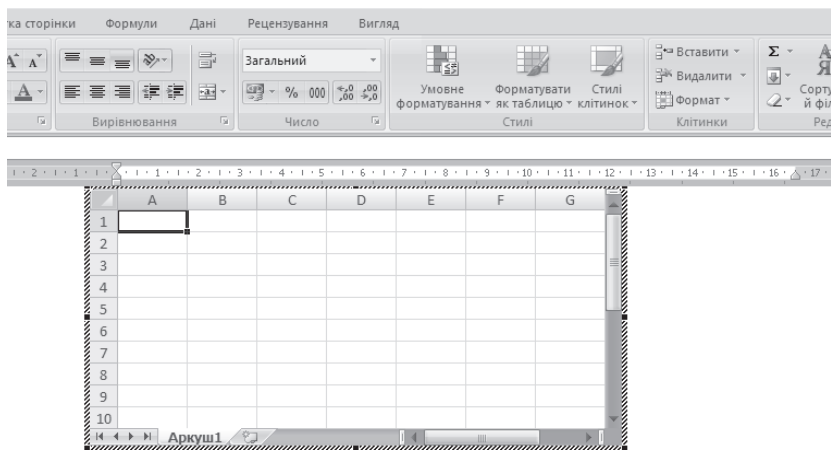


Рис.6.2.55. Приклад створення таблиці за допомогою пункта Електронна таблиця Excel

6. Створення таблиці за допомогою пункту *Експрес-таблиці*. Після вибору цього пункту на екрані з'являється список вбудованих шаблонів таблиць, що розкривається. Якщо Вам підходить один з шаблонів, просто клацніть по ньому лівою кнопкою миші і в документ буде вбудована потрібна таблиця. На рисунку 6.2.56 представлений список шаблонів таблиць, що розкривається, з виділенням одного з шаблонів – 3 підзаголовками 1.

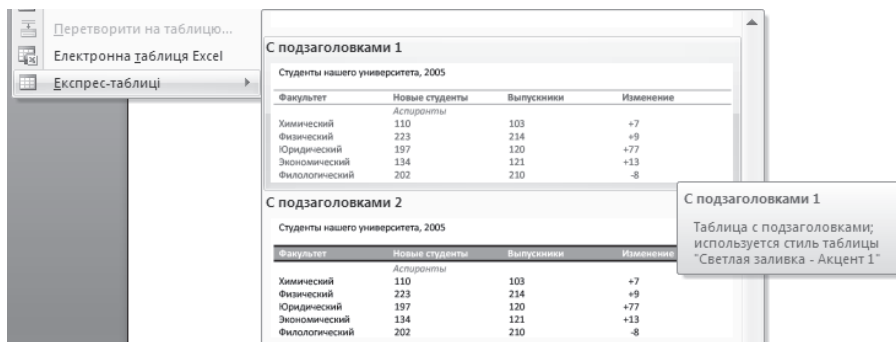


Рис.6.2.56. Приклад створення таблиці за допомогою пункту Експрес-таблиці із списку кнопки Таблиці, що розкривається

Після вставки шаблону дані таблиці слід замінити на реальні дані.

Введення даних в таблицю. Дані в таблицю, створену будь-яким з перерахованих способів, вводять з клавіатури. Заздалегідь треба зробити активною клітинку, призначену для введення, клацнувши по ній лівою кнопкою миші. Переходити від клітинки до клітинки можна за допомогою клавіші Tab або просто, виконуючи клацання лівою кнопкою миші по потрібній клітинці.

Додавання або видалення рядків, стовпців, елементів таблиці. Для видалення рядка, стовпця або клітинки слід їх виділити, після чого натиснути праву кнопку миші і в допоміжному меню, що розкрилося, вибрати пункт *Видалити клітинки*. На екрані з'явиться діалогове вікно **Видалення клітинок**, в якому треба встановити видалити весь рядок (для стовпця - видалити весь стовпець, для клітинки – зі зсувом вліво або зі зсувом угору) (рисунок 6.2.57).

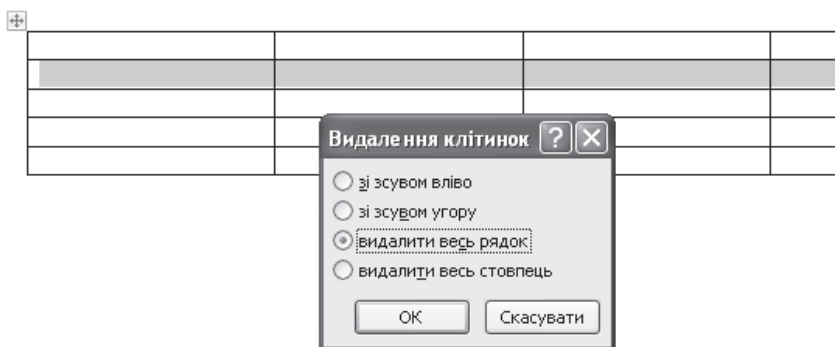


Рис.6.2.57. Діалогове вікно Видалення клітинок

Для додавання рядка, стовпця або клітинки (клітинок) слід їх виділити, після чого натиснути праву кнопку миші і в допоміжному меню, що розкрилося, вибрати пункт *Вставити* з трикутною стрілкою поруч. У списку, що розкрився, – *Додати стовпці зліва, Додати стовпці справа, Додати рядки вище, Додати рядки нижче, Вставити клітинки...* - вибрати потрібний варіант.

Об'єднання і розбиття елементів таблиці. Для об'єднання клітинок заздалегідь потрібні клітинки необхідно виділити, після чого натиснути на

праву кнопку миші і в допоміжному меню, що розкрилося, вибрати пункт *Об'єднати клітинки*.

Для розбиття клітинок заздалегідь потрібну клітинку необхідно виділити, після чого натиснути праву кнопку миші і в діалоговому вікні, що розкрилося, вибрати пункт *Розділити клітинки...* . Після чого на екрані з'явиться діалогове вікно **Поділ клітинок**, в якому слід вказати потрібне число рядків і стовпців. На рисунку 6.2.58 представлено діалогове вікно **Поділ клітинок**.

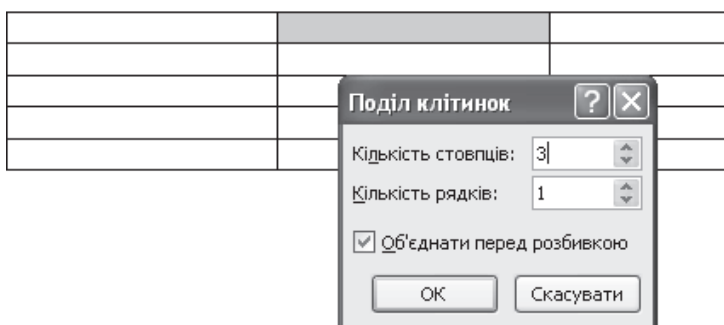
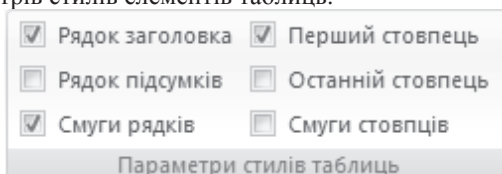


Рис.6.2.58. Діалогове вікно Поділ клітинок

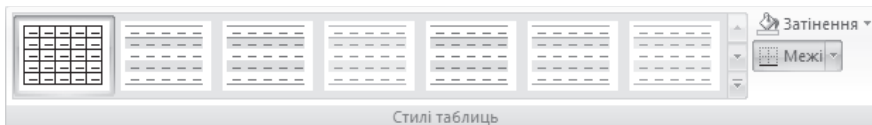
Оформлення таблиці за допомогою інструментів допоміжних вкладок Конструктора (Табличні знаряддя) і Макету. Клацніть лівою кнопкою миші по будь-якому елементу таблиці, створеної на підставі одного з описаних вище способів (за винятком таблиці Excel).

Вкладка *Конструктор (Табличні знаряддя)* містить наступні групи інструментів.

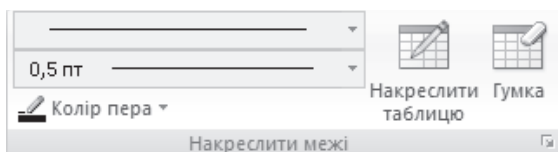
1. Група *Параметри стилів таблиць* призначена для установки параметрів стилів елементів таблиць.



2. Група *Стилі таблиць* призначена для зміни виду таблиці за допомогою вбудованих стилів, а також для зміни кольору виділених клітинок за допомогою кнопки *Затінення* і для установки виду межі за допомогою кнопки *Межі*.

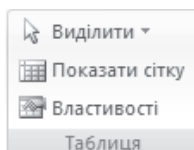


3. Група *Накреслити межі* призначена для зміни таблиці «уручну» за допомогою інструменту «олівець» (кнопка *Накреслити таблицю*) можна додати рядки або стовпці таблиці, а за допомогою інструменту гумка (кнопка *Гумка*) – стерти деякі межі таблиці. При натисненні кнопки *Гумка* покажчик миші набуває вид гумки.

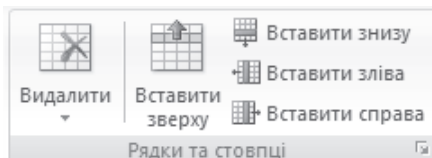


Вкладка *Макет* містить наступні групи інструментів.

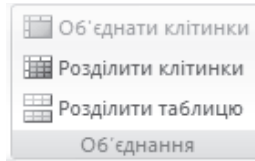
1. Група *Таблиця* призначена для установки властивостей таблиці



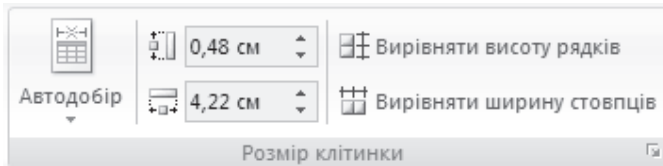
2. Група *Рядки та стовпці* призначена для вставки та видалення рядків і стовпців



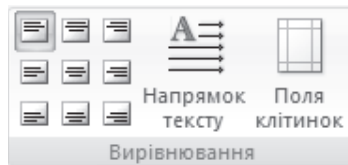
3. Група *Об'єднання* призначена для об'єднання і розділення клітинок таблиць



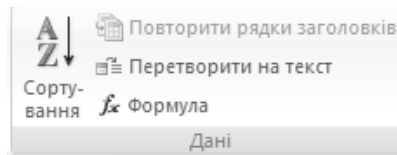
4. Група *Розмір клітинки* призначена для установки розмірів клітинок, автопідбору розмірів елементів таблиці



5. Група *Вирівнювання* призначена для установки типу вирівнювання тексту в клітинках таблиці, зміни напрямку тексту та завдання полів клітинок



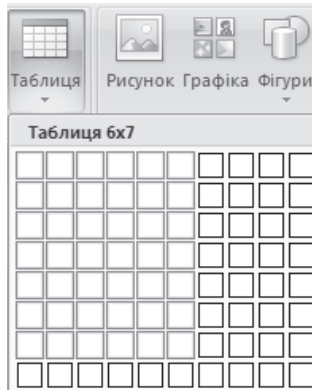
6. Група *Дані* призначена для сортування даних в таблиці і введення формул в клітинки таблиці



Приклад. Створити таблицю міжгалузевого балансу виробництва і розподілу продукції наступного вигляду

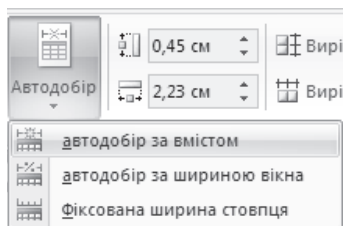
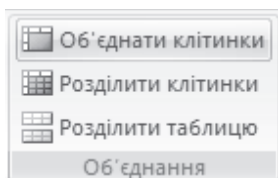
Виробляючі галузі	Споживаючі галузі				
	1	2	3	Кінцева продукція	Валова продукція
1	232,6	51,0	291,8	200,0	775,3
2	155,1	255,0	0,0	100,0	510,1
3	232,6	51,0	145,9	300,0	729,6
Умовно чиста продукція	155,0	153,1	291,9	600,0	
Валова продукція	775,3	510,1	729,6		2015,0

Створимо таблицю, що містить 6 стовпців і 7 рядків, наприклад, за допомогою Вставлення таблиці.



На екрані з'явиться наступний шаблон таблиці.

Для створення заголовків стовпців таблиці виділимо перші два рядки першого стовпця і об'єднаємо їх. Для цього перейдемо на допоміжну вкладку *Макет* в групу інструментів *Об'єднання* і натиснемо на кнопку *Об'єднати клітинки*. Потім в групі інструментів *Розмір клітинки* слід натиснути на кнопку *Автодобір*, що розкривається, і в списку, що розкрився, вибрати пункт *Автодобір за вмістом*. У об'єднаній клітинці введемо заголовок першого стовпця – *Виробляючі галузі*, задалегідь встановивши зображення курсивом і вирівнювання по центру. Аналогічно слід вчинити, виділивши, починаючи з другого, всі стовпці першого рядка (при виконанні цієї операції немає необхідності встановлювати автодобір за вмістом).



<i>Виробляючі галузі</i>	<i>Споживаючі галузі</i>				

Далі всі елементи таблиці слід заповнити за наведеним у завданні зразком.

Приклад. Упорядкуйте за убутанням розрахункової ймовірності варіанта

Варіант прогнозу	Розрахункова ймовірність варіанта %	Прогноз прибутку за проектами, тис. грн.	
		А	В
1	35	500	800
2	40	570	870
3	25	620	750

Після створення й заповнення зазначеної таблиці користувачеві необхідно виділити всі рядки таблиці (крім заголовків стовпців), що містять числові дані. Після цього перейти на допоміжну вкладку *Макет* і в групі інструментів *Дані* натиснути на кнопку *Сортування*. У результаті на екрані

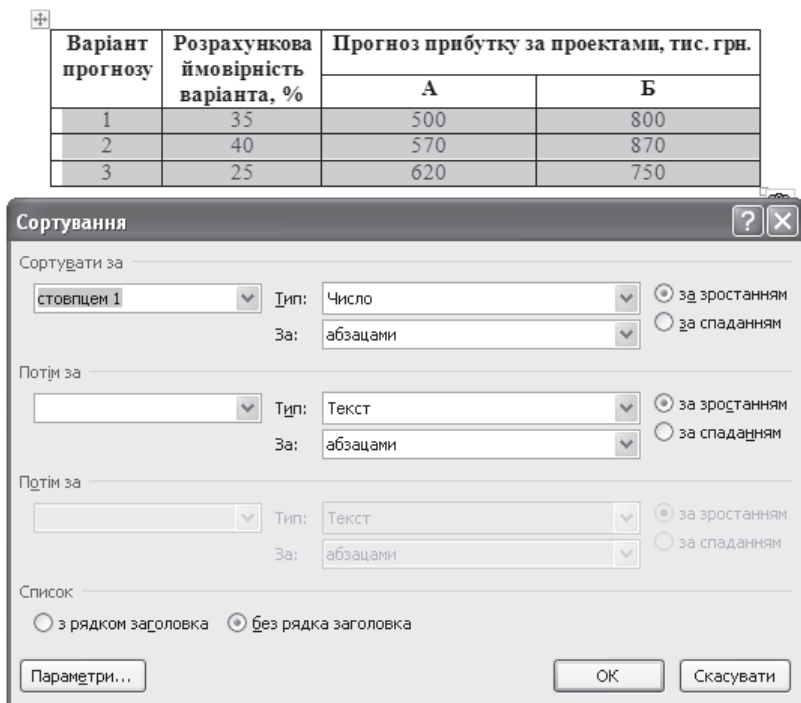


Рис.6.2.59. Діалогове вікно Сортування

У діалоговому вікні *Сортування* в списку *Сортувати*, що розкривається, необхідно встановити за стовпцем2, а потім активізувати за убуванням. Результат сортування наведений на рисунку 6.2.60.

Варіант прогнозу	Розрахункова ймовірність варіанта, %	Прогноз прибутку за проектами, тис. грн.	
		А	В
2	40	570	870
1	35	500	800
3	25	620	750

Рис.6.2.60. Результат сортування таблиці за убуванням розрахункової ймовірності варіанта

Питання і завдання для самостійної роботи

Створіть бланки наступних документів.

1. Заповніть журнал обліку бланків

Надходження					Видача							
дата надходження	номер супровідного листа	звідки надійшли	кількість примірників	серія і номери бланків	дата	дата і номер документа	кому видано		кількість примірників	серія і номери бланків	розписка про отримання	примітка (відмітка про знищення)
							найменування підрозділу	прізвище та ініціали отримувача				

2. Заповніть журнал обліку документів та видань з грифом «Для службового використання»

Дата надходження та індекс документа	Дата та індекс документа	Звідки надійшов або куди надісланий	Вид документа та короткий зміст	Кількість сторінок		Кількість та номери примірників	Резолюція або кому надісланий на виконання	Відмітка про взяття на контроль та термін	Дата і підпис		Індекс (номер) справи, до якої підшито доку-мент	Відмітка про знищення	Примітка
				документа	додавка				отримання	повернення			

3. Заповніть картку обліку документів та видань за грифом «Для службового використання»

(Лицьовий бік)

1. Контроль	2. 1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31	3. Гриф
4. Кореспондент		5. Адресат
6. Дата надходження та індекс		7. Дата та індекс документа
8. Вид документа та його короткий зміст		9. Кількість примірників та їх номери
		10. Кількість сторінок
11. Резолюція або кому надісланий на виконання		
12. Відмітка про виконання документа		13. Номер справи за номенклатурою

(Зворотний бік)

14. Відмітка про отримання		15. Відмітка про повернення
16. Перевірка виконання		
17. Інші відмітки		
18. Фонд №	19. Опис №	20. Справа №

4. Заповніть інвентаризаційний опис

ІНВЕНТАРИЗАЦІЙНИЙ ОПИС _____ на _____ 200__ р.

товарно-матеріальних цінностей, що знаходяться _____
склад, комора

на відповідальному зберіганні _____

посада, прізвище, ім'я, по батькові

інвентаризація проведена на підставі наказу (розпорядження)

від _____ 200__ р. № _____

№ п/п	номенклатурний номер	Найменування матеріалу	Розмір (тип)	Сорт або марка	Одиниця виміру	Фактична наявність	Ціна	Примітка

5. Заповніть заяву на переказ готівки

Заява на переказ готівки № _____				
Дата здійснення операції _____				
Дата валютування _____				
Назва валюти	№ рахунка		Сума	Еквівалент у гривнях
Дебет				
Кредит				
Загальна сума (цифрами)				
Платник _____				
Код платника** _____				
Банк платника* _____				
Отримувач _____				
Код отримувача* _____				
Банк отримувача _____				
Код банку отримувача* _____				
Загальна сума _____			(словами)	
Призначення платежу _____				
Пред'явлений документ** _____ (паспорт або документ, що його замінює)				
серія _____		№ _____		виданий _____ (номер документа) (найменування установи, яка видала документ)
(дата видачі документа)				
(дата народження)		(місце проживання особи)		
Додаткові реквізити _____				
Підпис платника _____			Підписи банку _____	

6. Заповніть видатковий касовий ордер

Номер документа	Дата складання		Кореспондуючий рахунок	Код аналітичного обліку	Сума	Код цільового призначення	

Видати _____
прізвище, ім'я, по батькові

Підстава _____

_____ грн. _____ коп.
прописом

Керівник

Головний бухгалтер

7. Заповніть інформацію про майновий стан та фінансову-господарську діяльність емітента

ІНФОРМАЦІЯ

про майновий стан та фінансово-господарську діяльність емітента

Інформація про основні засоби емітента (за залишковою вартістю)

Найменування основних засобів	Власні основні засоби (тис. грн.)		Орендовані основні засоби (тис. грн.)		Основні засоби Всього (тис. грн.)	
	на початок періоду	на кінець періоду	на початок періоду	на кінець періоду	на початок періоду	на кінець періоду
1. Виробничого призначення:						
— будівлі та споруди						
— машини та обладнання						
— транспортні засоби						
— інші						
2. Невиробничого призначення:						
— будівлі та споруди						
— машини та обладнання						

— транспортні засоби						
— інші						
Усього						
Пояснення:						

8. Заповніть розрахунок суми страхових внесків на загальнообов'язкове державне пенсійне страхування

Розрахунок суми страхових внесків на загальнообов'язкове державне пенсійне страхування за 20__ рік

Подається платником до органів Пенсійного фонду України щорічно до 1 квітня року, наступного за звітним роком

Повідомляю, що я,

_____ ,
(прізвище, ім'я, по батькові)

номер реєстрації платника _____ ,

ідентифікаційний номер _____ ,

проживаю за адресою _____ ,

телефон _____ ,

кількість членів сім'ї, які беруть участь у провадженні підприємницької діяльності _____ ,

одержав дохід (прибуток) від підприємницької діяльності, що підлягає оподаткуванню податком на доходи фізичних осіб:

(грн.)

Назва місяця	Сума чистого доходу (прибутку), заявлена у податковій декларації		Сума доходу, на яку нараховуються внески з урахуванням максимальної величини	Нараховано внесків (гр. 4 x 32 %)	Сплачено авансових сум
	фізичної особи – суб'єкта підприємницької діяльності	членів сім'ї, які беруть участь у провадженні підприємницької діяльності*			
1	2	3	4	5	6
Січень					
Лютий					
Березень					
Квітень					
Травень					
Червень					

Липень					
Серпень					
Вересень					
Жовтень					
Листопад					
Грудень					
Усього					

Правильність зазначених відомостей стверджую

(підпис, дата)

Бібліографічний список

1. Гольшева А.В., Ерофеев А.А. Word 2007 «без воды». Все, что нужно для уверенной работы. – СПб.: Наука и Техника, 2008. – 192 с.
2. Культин Н.Б. Word 2007. Самое необходимое. – СПб.: БХВ-Петербург, 2007. – 176 с.
3. Вонг Уоллес Office 2003 для «чайников»: пер. с англ. – М.: Издательский дом «Вильямс», 2004. – 336 с.

6.3. ТАБЛИЧНИЙ ПРОЦЕСОР Microsoft Office Excel 2007

Студентам, працівникам офісів, науковцям доводиться вирішувати завдання з опрацювання даних та їх подання в наочному вигляді. Для вирішення завдань подібного роду наявні дані зручно представляти у вигляді таблиць, що складаються з рядків і стовпців. Кожен стовпець заповнюється даними одного типа, наприклад, в один стовпець можна записати список товарів, а в іншій - ціни для кожного з них. Вид таблиці має і відомість на зарплату, деякі стовпці розраховуються по формулах, в яких беруть участь дані інших стовпців. Наприклад, при обчисленні суми виплати необхідно із заробленої суми відняти податок, і це зробити для даних всього стовпця. Також часто потрібно обчислити суму чисел всього стовпця. Подібні завдання зручно вирішувати за допомогою програм, званих *електронними таблицями*.

➤ *Електронна таблиця* – програма для побудови і розрахунку даних, представлених у вигляді рядків і стовпців, а також ілюстрації результатів у вигляді різних графіків.

Існує декілька поширених програм - електронних таблиць для персональних комп'ютерів. Одна з них, що працює в системі Windows, називається Excel. Зручність представлення даних, різноманітність обчислень дозволяють користувачеві провести їх якісну оцінку і вирішити завдання по управлінню підприємством.

Опишемо призначений для користувача інтерфейс табличного редактора Microsoft Office Excel 2007, представлений на рисунку 6.3.1. У Microsoft Office Excel 2007 на відміну від раніших версій реалізований новий стрічковий інтерфейс.

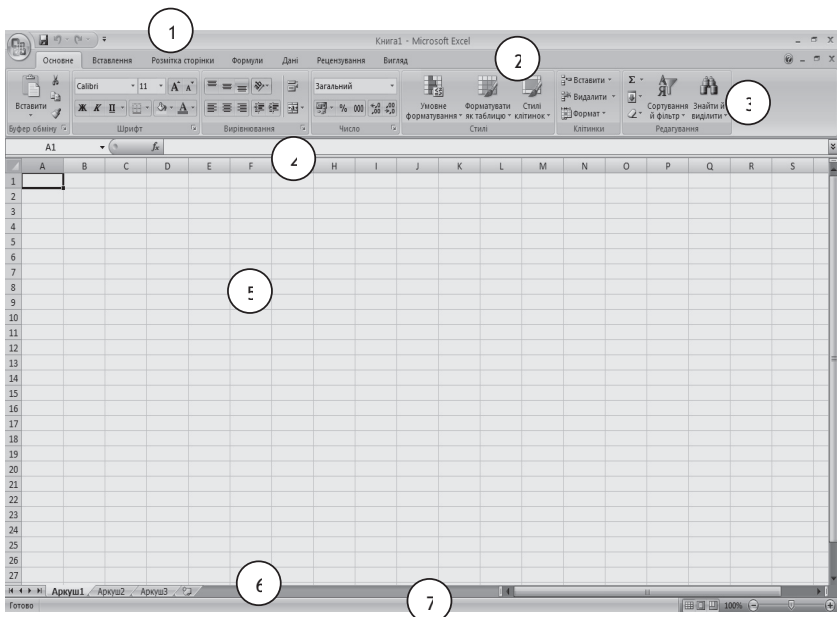


Рис 6.3.1. Вікно табличного редактора Microsoft Office Excel 2007

Перерахуємо основні елементи стрічкового інтерфейсу Microsoft Office Excel 2007:

- 1 – Кнопка Microsoft Office і панель швидкого доступу
- 2 – Вкладки стрічки інструментів
- 3 – Стрічка інструментів
- 4 – Рядок формул
- 5 – Робоча область у вигляді розміченої таблиці, що складається з сукупності клітинок (поточний лист)
- 6 – Ярлики листів
- 7 – Рядок стану

У лівому верхньому куті вікна розташована зручна кнопка *Office*



, при натисканні на яку користувач одержує доступ до таких основних

команд роботи з документом як створення нового документа, відкриття існуючого документа, збереження документа, його друк, підготовка документа (настроювання його властивостей, установка права доступу до документа та ін.), відправлення по електронній пошті, закриття документа.

Праворуч від кнопки Office розташована Панель швидкого доступу. За

замовчуванням вона містить кнопки швидкого доступу *Зберегти* ,

Скасувати додавання фрагмента  (Скасувати введення) (скасування

попередньої команди) і *Повторити додавання фрагмента*  (Повторити

введення) (повторення попередньої команди), кнопка *Сортування від Я до А*

 і кнопка *Настроювання панелі швидкого доступу* .

Кнопка *Настроювання панелі швидкого доступу* дозволяє додати й інші кнопки для найбільш часто використовуваних команд. Для цього напроти команд потрібно встановити галочку так, як це зроблене, наприклад, для команди *Зберегти* (рисунок 6.3.2).

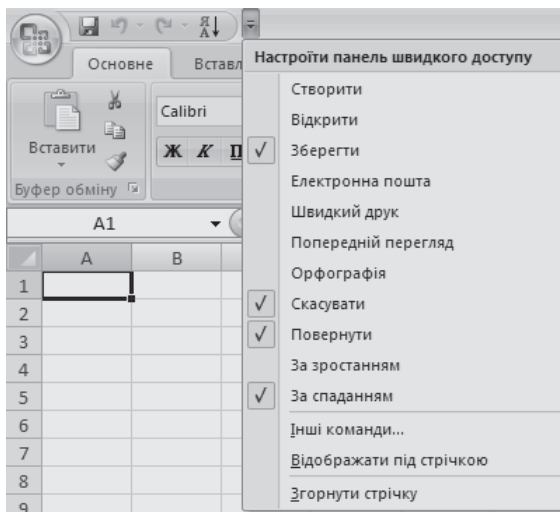


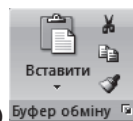
Рис.6.3.2. Активізація кнопки *Настроїти панелі швидкого доступу*

Нижче панелі швидкого доступу розташовані вкладки головного меню. Кожна вкладка містить стрічку з переліком основних команд, виконати які можна, натиснувши на відповідну кнопку або вибрати потрібну команду, викликавши допоміжне діалогове вікно за допомогою натискання на кнопку



. Якщо навести покажчик миші на будь-яку кнопку стрічки, то з'являється спливаюча підказка, що пояснює призначення команди, відповідної до цієї кнопки. Перелічимо пункти (вкладки) головного меню й стисло охарактеризуємо команди цих пунктів.

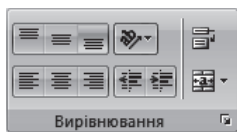
1. Вкладка **Основне** містить наступні основні команди:



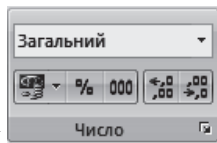
- копіювання, вставлення (команди роботи з буфером обміну)



- установки формату клітинок (вибір стиля, розміра шрифта, типа креслення),

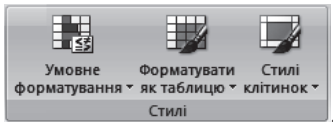


- установка типа вирівнювання



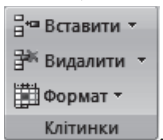
- установка числового формату

- установка стилю



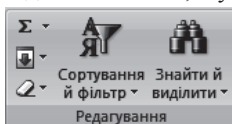
- вставлення, виділення, зміна ширини або висоти відповідного стовпця і

рядка



- команди пошуку і виділення необхідного тексту, упорядкування

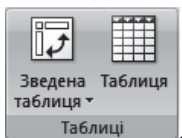
(сортування) даних, очистки клітинок та ін.



2. Вкладка *Вставка* містить наступні основні команди:

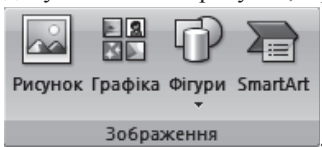
- команда оформлення виділеного діапазону клітинок у вигляді таблиці -

створення таблиці

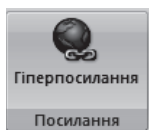
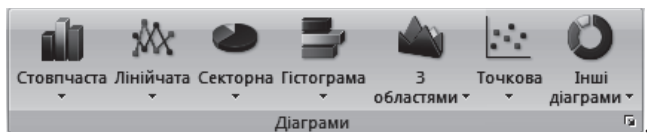


- вставлення в документ Excel рисунка, графіка, фігури або графічного

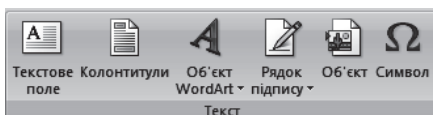
об'єкта SmartArt



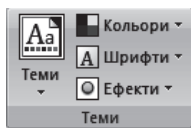
- вставлення в документ Excel діаграми



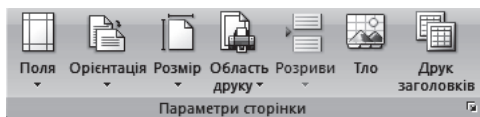
- створення гіперпосилання
- вставлення текстового поля, об'єкту WordArt, символу і тому подібне



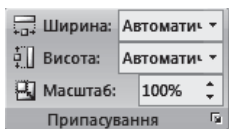
3. Вкладка *Розмітка сторінки* містить наступні основні команди:



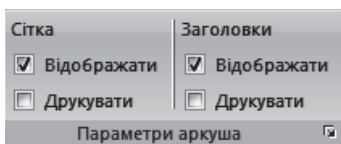
- зміна загального вигляду документа (теми)



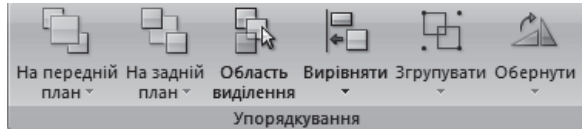
- установка параметрів сторінки



- припасування



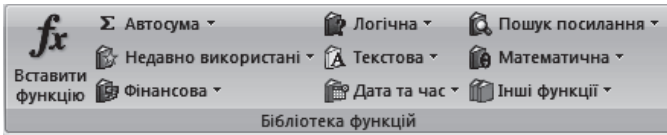
- установка параметрів аркуша



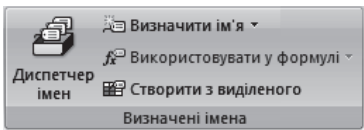
- упорядкування

4. Вкладка *Формули* містить наступні основні команди:

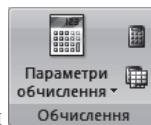
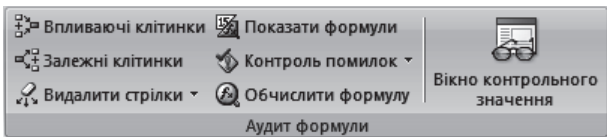
- бібліотека функцій (вставка функції і вибір категорії)



- призначення імен окремим клітинкам і групам клітинок



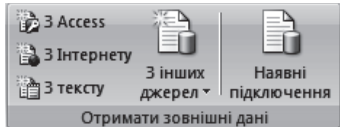
- виконання аудиту формул (перевірка помилок у формулах, налаштування формули з можливістю окремого обчислення кожного її компонента, відображення стрілок, які вказують залежність між вічками)



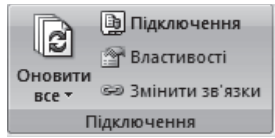
- вибір режиму обчислення формули

5. Вкладка *Дані* містить наступні основні команди:

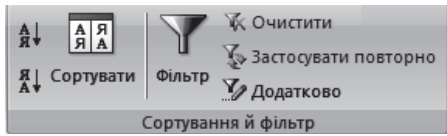
- імпорт даних



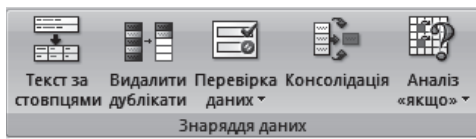
- оновлення всіх відомостей в книзі на основі джерела даних



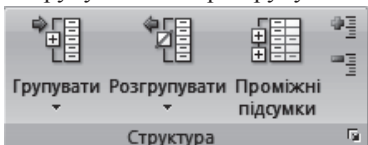
- сортування даних по декількох критеріях і фільтрація



- перевірка даних (запобігання введенню некоректних даних у клітинку, видалення рядків, що повторюються, об'єднання значень з деякого діапазону в новий діапазон, розподіл вмісту клітинки по декількох стовпцях, перевірка різних значень формул)

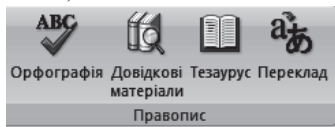


- групування і розгрупування діапазона значень, проміжний підсумок



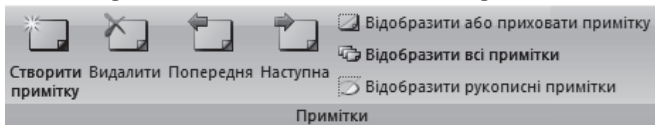
6. Вкладка *Рецензування* містить наступні основні команди:

- перевірка орфографії, пошук в словниках, тезаурус (пропозиція слів, однакових за значенням (синонімів) з виділеним словом) і переклад

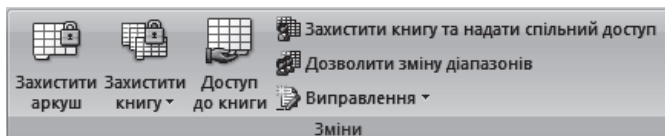


виділеного тексту на іншій мові

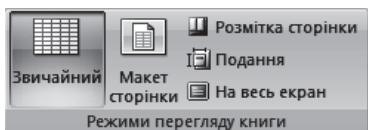
- створення, видалення і переміщення між примітками



- організація захисту книги або окремого аркуша і доступу до книги

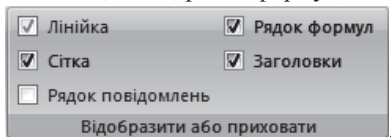


7. Вкладка *Вигляд* містить наступні основні команди:



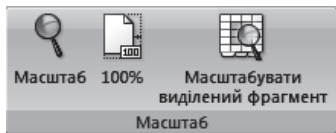
- установка режиму перегляду книги

- відображення або приховання лінійки, сітки, рядка формул, заголовків строк

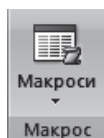
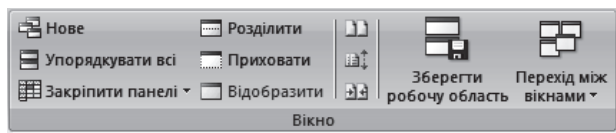


- і стовпців, рядки повідомлень

- установка масштабу



- способи відображення вікна



- запис або настроювання параметрів макросів


6.3.1. Робоча книга і дії з аркушами книги

Файл, що створюється в MS Excel, називається *робочою книгою*. Робоча книга складається з декількох аркушів (за умовчанням три), кількість яких обмежується об'ємом доступної оперативної пам'яті. Імена аркушів за умовчанням – Аркуш1, Аркуш2, Аркуш3 – розташовані на ярликах в нижній частині вікна над статусним рядком.



Розглянемо, як додавати, перейменовувати, переміщати і видаляти аркуши з робочої книги.

Додавання аркушу. Додавання аркушу до робочої книги можна виконати декількома способами: 1) натиснути комбінацію клавіш <SHIFT>+<F11> (новий аркуш буде розміщений перед активним аркушем);

2) клацнути по ярликові аркушу – *Вставити аркуш* виду  лівою кнопкою миші (новий аркуш буде розміщений після всіх існуючих аркушів); 3) натиснути правою кнопкою миші на ярлику будь-якого аркушу й вибрати пункт *Вставити...* (рисунок 6.3.3) у контекстному меню, що з'явилось, потім вибрати в діалоговому вікні *Вставка*, що розкрилося, піктограму Аркуш і

натиснути кнопку *OK* (новий аркуш буде розміщений перед активним аркушем).

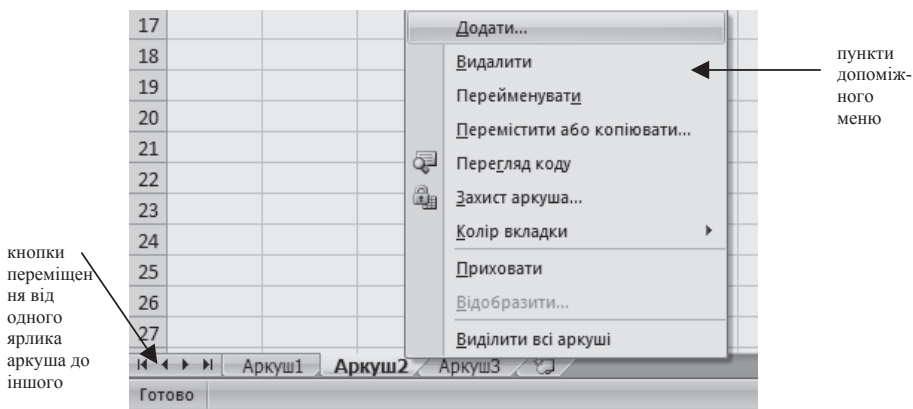


Рис.6.3.3. Допоміжне меню, що виникає при натисканні правої кнопки миші

Переходи між аркушами. Зліва від ярликів аркушів розташовані кнопки переходу від одного ярлика до іншого:



Переход до першого ярлика аркуша;



Переход до ярлика попереднього аркуша;



Переход до ярлика наступного аркуша;



Переход до останнього ярлика аркуша.

Виділення аркуша. Для того, щоб виділити або зробити активним який-небудь аркуш, слід клацнути по ньому лівою кнопкою миші. Для виділення групи аркушів слід натиснути клавішу *Ctrl* і, утримуючи її, клацати лівою кнопкою миші по потрібних аркушах.

Перейменування аркуша. Для того, щоб перейменувати аркуш, слід або виконати подвійне клацання лівою кнопкою миші по потрібному аркушу і ввести назву аркуша, або викликати допоміжне меню, клацнувши правою кнопкою миші по потрібному аркушу, і вибрати в цьому меню пункт *Перейменувати*, після чого ввести назву аркуша.

Переміщення (копіювання) аркуша. Для того, щоб перемістити аркуш, потрібно зробити його активним, після чого натиснути ліву кнопку миші і, втримуючи її, перетягнути аркуш у потрібне місце, або вибрати пункт допоміжного меню *Перемістити (Скопіювати...)*. У результаті на екрані з'явиться однойменне діалогове вікно (рисунок 6.3.4), у якому можна вибрати робочу книгу, у яку треба перемістити (скопіювати) аркуш, а також позицію його розміщення й натиснути на кнопку *ОК*. Для копіювання аркуша перед натисканням кнопки *ОК* слід поставити галочку для пункту *Створити копію*.

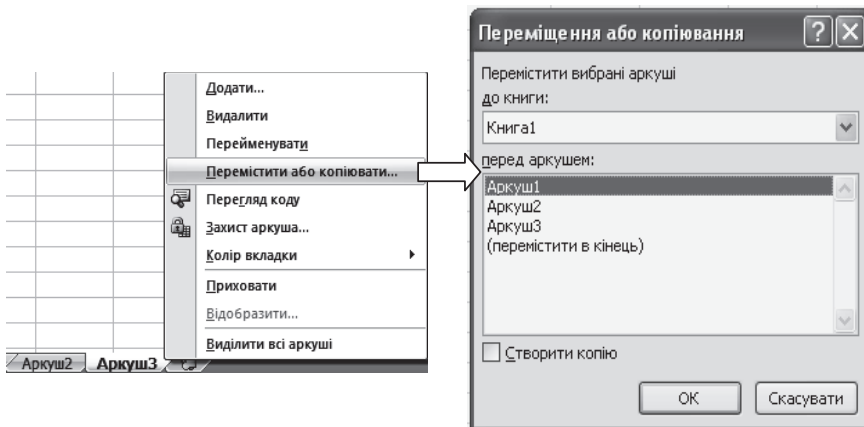


Рис.6.3.4. Переміщення (копіювання) аркуша

Зміна кольору вкладки аркуша. Для того, щоб змінити колір вкладки, необхідно зробити активним потрібний аркуш і викликати допоміжне меню, в якому вибрати пункт *Колір вкладки* і в списку, що розкрився, вибрати потрібний колір клацанням лівої кнопки миші (рисунок 6.3.5).

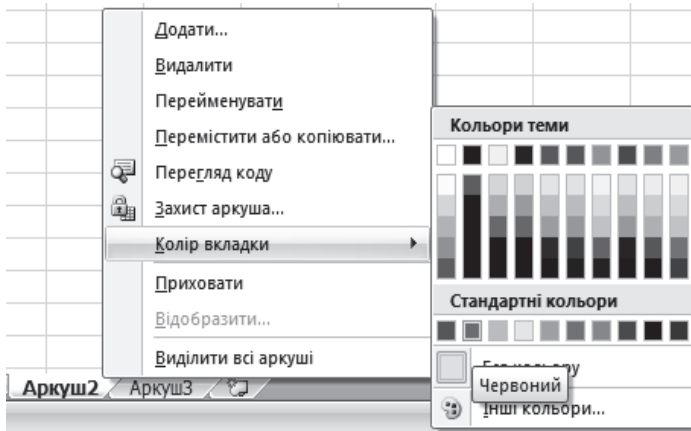


Рис.6.3.5. Змінення кольору вкладки ярлика аркуша

Видалення аркуша. Для того, щоб видалити аркуш, слід зробити його активним, після чого викликати допоміжне меню клацанням правої кнопки миші і вибрати пункт *Видалити*.

Будь-який аркуш є електронною таблицею, що містить в MS Excel 2007 1048576 рядків і 16384 стовпців. Рядки позначаються по порядку числами (1,2,3.,1048576), стовпці - латинськими буквами і їх поєднаннями (A, B, C., Z, AA, AB.,AZ, BA.,BZ, CA.,CZ.,XFD). Електронна таблиця є сукупністю клітинок, кожна з яких має адресу. Зазвичай адреса клітинки включає ім'я стовпця і номер рядка. Наприклад, клітинка, розташована в лівому верхньому кутку електронної таблиці на перетині першого стовпця і першого рядка, має адресу A1, а клітинка, розташована в правому нижньому кутку на перетині останнього стовпця і рядка, – адресу XFD1048576.

6.3.2. Операції над клітинками електронної таблиці

Активізація клітинки і виділення груп клітинок. Одна з клітинок зазвичай завжди активна – вона поміщена в чорний прямокутник (рамку).



Найпростіший спосіб зробити активною деяку клітинку – виконати клацання лівою кнопкою миші по потрібній клітинці або скористатися клавішами управління курсором і підвести чорну прямокутну рамку до потрібної клітинки. Для виділення рядка слід клацнути лівою кнопкою миші по номеру потрібного рядка, а для виділення стовпця – по букві потрібного стовпця. Для виділення групи суміжних клітинок слід натиснути ліву кнопку миші і, утримуючи її, протягнути уздовж всієї групи суміжних клітинок або, натиснувши клавішу Shift і утримуючи її, скористатися потрібною клавішею управління курсором і виділити групу суміжних клітинок. Для виділення декількох груп клітинок слід виділити першу групу клітинок описаним вище способом, а потім натиснути клавішу Ctrl і, утримуючи її, виділити другу групу клітинок і так далі. Говорять, що клітинки утворюють діапазон, якщо вони можуть бути поміщені в прямокутник. Для завдання адреси діапазону клітинок використовується двокрапка, яка розділяє адресу лівої верхньої клітинки і правої нижньої клітинки прямокутника. Наприклад, адреса приведеного на рисунку 6.3.6 діапазону записується як B2:D3.

	A	B	C	D
1				
2				
3				
4				

Рис.6.3.6. Приклад виділення і запису адреси діапазона клітинок

Введення даних у клітинку і установка формату даних. Для того, щоб у клітинці з'явився курсор і можна було вводити дані, слід або виконати подвійне клацання лівою кнопкою миші по активній клітинці або натиснути клавішу *F2*. Якщо Ви допустили помилку при введенні даних і хочете відмінити введення, то натискуйте на клавішу *Esc*. Після закінчення введення правильних даних слід натискувати клавішу *ENTER*, або клацнути лівою кнопкою миші по будь-якій іншій клітинці, або скористатися клавішею управління курсором для переходу в іншу клітинку. Після цього активною стане наступна клітинка.

Щоб розширити стовпець для розміщення в ньому всього тексту, слід перемістити покажчик миші на праву межу заголовка стовпця і, коли з'явиться подвійна чорна стрілка, виконати подвійне клацання лівою кнопкою миші.

Перед введенням даних у клітинку необхідно визначитися з форматом (типом) даних. Це може бути числовий, грошовий, фінансовий, процентний, дробовий, експонентний формат або формат час, дата тощо. Для установки формату слід активізувати потрібну клітинку або виділити діапазон клітинок, потім перейти на вкладку *Основне* в розділ *Число*, потім вибрати потрібний формат або зі списку , що розкривається, або вибрати потрібну кнопку, наприклад, для завдання формату фінансовий – кнопку , а при використанні списку, що розкривається, ще й установити грошову одиницю. Установити формат можна й інакше. Наприклад, виділивши потрібні клітинки, викликати допоміжне меню, натиснувши праву кнопку миші, а потім здійснити вибір команди *Формат клітинок...*. На екрані з'явиться однойменне діалогове вікно (наведено на рисунку 6.3.7), у якому слід зробити активною вкладку *Число* (вона активна за замовчуванням). Розглянемо, як установити потрібний формат клітинки. За замовчуванням клітинка має загальний формат, який використовується для відображення як текстових, так і числових значень будь-якого типу. Для подання чисел традиційно використовується числовий формат. Числові дані – це послідовність цифр від 0 до 9 з використанням спеціальних символів операцій (+, -, *, /), точок і круглих дужок. При виборі цього числового формату можна встановити число підтримуваних знаків після точки (десяткових знаків).

Для відображення грошових і фінансових величин використовується грошовий і фінансовий формати, для яких можна встановити грошову одиницю, причому фінансовий формат використовується для вирівнювання грошових величин по роздільнику цілої і дробової частини. На рисунку наведений приклад установки грошового формату і вибір грошової одиниці – гривні. Після завершення всіх дій слід натиснути кнопку *OK*.

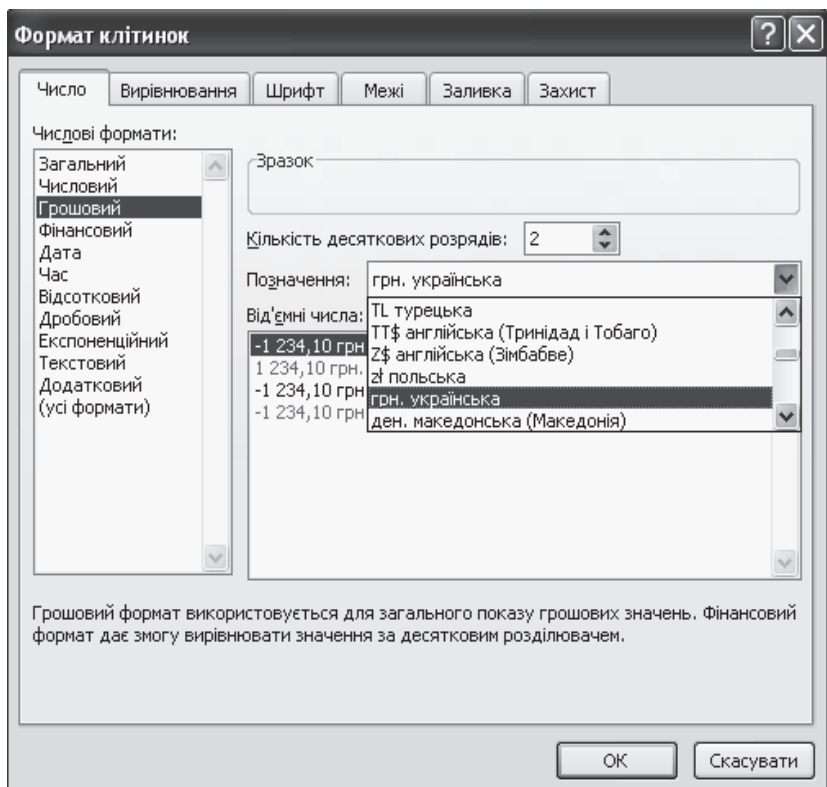








Рис. 6.3.7. Приклад установки грошового формату і вибору грошової одиниці – гривні

Редагування вмісту клітинки. Для того, щоб відредагувати дані у клітинці, необхідно виконати подвійне клацання лівою кнопкою миші по потрібній клітинці або натиснути клавішу *F2*, заздалегідь зробивши потрібну клітинку активною. Після появи текстового курсора у клітинці потрібно за допомогою клавіш управління курсором підвести його до потрібного символу. Якщо символ слід видалити, то розташуйте курсор праворуч від символу, що видаляється, і натискайте клавішу *Backspace* або розташуйте

курсор зліва від символу, що видаляється, і натискайте клавішу *Delete*. Після завершення редагування натискайте клавішу *ENTER*.

Вирівнювання вмісту клітинки. Для вирівнювання вмісту клітинки слід зробити клітинку активною, після чого вибрати вкладку *Основне* й у розділі

Вирівнювання вибрати тип вирівнювання: по лівому краю , по центру , по правому краю , по верхньому краю , посередині , по нижньому краю .

Видалення вмісту клітинки. Для того, щоб повністю видалити вміст клітинки, необхідно зробити цю клітинку активною і натиснути клавішу *Delete*.

Швидке копіювання даних у клітинку. Зробіть активним клітинку, вміст якої необхідно скопіювати. Активну клітинку буде поміщено в чорний прямокутник – табличний курсор, в нижньому правому кутку якого розташований маленький чорний квадрат, званий *маркером автозаповнення*. При підведенні покажчика миші, який має вид білого хресту, до маркера автозаповнення білий хрест перетвориться в чорний тонкий хрест. У цей момент слід натискувати ліву кнопку миші і, утримуючи її, протягнути уздовж тих клітинок, в які необхідно скопіювати дані. Приклад копіювання вмісту клітинки A1 – текст «вартість товару» – в діапазон клітинок від A2 до A10 приведений на рисунку 6.3.8. При швидкому копіюванні виникає спливаюча підказка, яка вказує, що саме копіюється.



Рис.6.3.8. Приклад виконання швидкого копіювання

Для того, щоб не створювати при копіюванні серію назв місяців, чисел або інших даних, при швидкому копіюванні слід утримувати натиснутою клавішу *Ctrl*.


Наведемо приклад швидкого копіювання й створення серії шляхом швидкого копіювання. Уведемо в клітинку A1 значення 10, в A2 – 15, в A3 – 20. Виділимо діапазон клітинок від A1 до A3. Підведемо покажчик миші до маркера автозаповнення й коли покажчик миші прийме вид чорного тонкого хреста, натиснемо ліву кнопку миші й, утримуючи, її заповнимо клітинку A4. Як видно, у клітинку A4 буде скопійоване значення 25, тобто до значення клітинки A3 буде додано 5 (див. рисунок 6.3.9 ліворуч). Функція автозаповнення «інтерпретує» вміст клітинок діапазону й створює необхідну послідовність. Так, у клітинках A1, A2, A3 задано три початкові значення послідовності, а інші значення завдяки функції автозаповнення будуть обчислені автоматично. Якщо при виконанні автозаповнення утримувати натиснутою клавішу *Ctrl*, то буде копіюватися серія, й у клітинку A4 буде скопійовано значення 10, в A5 – 15, в A6 – 20 (див. рисунок 6.3.9 праворуч). Такі послідовності являють собою арифметичні прогресії.

1	10
2	15
3	20
4	25
5	
6	

1	10
2	15
3	20
4	10
5	
6	

Рис. 6.3.9. Швидке копіювання при використанні клавіші Ctrl (праворуч) і без її використання (ліворуч)

Для створення геометричної прогресії слід ввести в деякій клітинці початкове значення прогресії, виділити клітинки, призначені для заповнення

елементами геометричної прогресії й натиснути кнопку *Заповнити* , що розкривається у групі інструментів *Редагування* на вкладці *Основне*. У списку, що розкривається, слід вибрати пункт *Прогресія*. На екрані з'явиться однойменне діалогове вікно, представлене на рисунку 6.3.10.

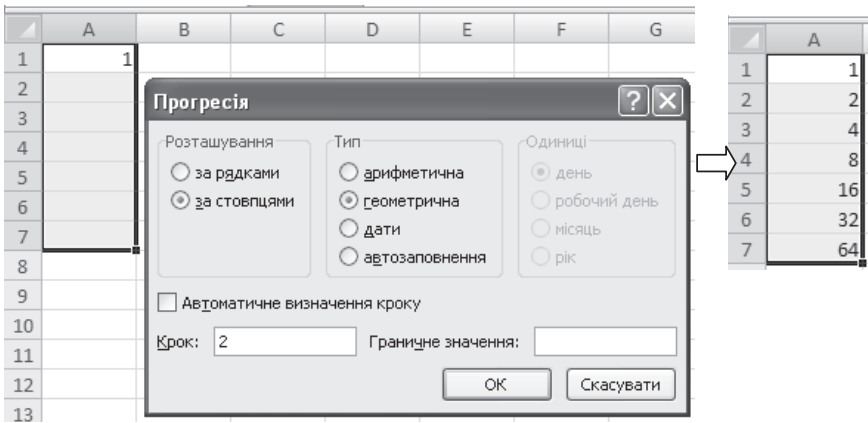


Рис.6.3.10. Діалогове вікно Прогресія

У діалоговому вікні *Прогресія* користувач може задати тип прогресії (арифметична, геометрична, дати, автозаповнення), крок, граничне значення, розміщення елементів прогресії (за рядками, за стовпцями), а для дат – встановити одиницю виміру. Результат заповнення виділених клітинок

елементами геометричної прогресії з початковим значенням 1 і кроком 2 представлений на рисунку 6.3.10 (праворуч).

Введення формули у клітинку. Після введення числових даних в таблицю виникає природне завдання їх обробки або аналітично – по певних законах, визначуваних формулами, або графічно – побудова графіків, діаграм і тому подібне. *Формула в MS Excel 2007* може складатися із значень, адрес клітинок, функцій і математичних операторів (+ (складання), - (віднімання) * (множення), / (ділення), ^ (піднесення до ступеня)). Вкажемо пріоритет при виконанні математичних операцій: 1) піднесення до ступеня і операції в дужках; 2) множення і ділення; 3) складання і віднімання.

Запис будь-якої формули в MS Excel починається із знаку рівності =. Після закінчення введення формули слід натиснути клавішу *ENTER*, і якщо формула задана вірно, у клітинці буде отриманий результат обчислення за введеною формулою.

Існують наступні види посилань на клітинку в MS Excel: відносне посилання, абсолютне посилання і комбінація цих двох видів посилань. Відносне посилання змінюється при копіюванні формули в іншу клітинку. Відносне посилання складається з адреси стовпця (відповідною стовпцю букви) і адреси рядка (відповідного рядку числа). Абсолютне посилання не змінюється при копіюванні формули в іншу клітинку. При записі абсолютного посилання використовується знак \$, який ставиться перед буквою і перед числом. Наведемо приклади використання відносних, абсолютних посилань і комбінації цих двох видів посилань.

Приклад 6.3.1. Введемо у клітинку A1 значення 5, а у клітинку A2 формулу вигляду =A1+1. Після натиснення клавіші *ENTER* у клітинці A2 буде обчислено значення 6 і активною стане клітинка A3. Зробимо активною клітинку A2 і виконаємо копіювання формули =A1+1 у клітинку з A3 по A10, використовуючи маркер автозаповнення. При цьому у клітинці A3 з'явиться значення 7, а в рядку формул формула вигляду =A2+1. Копіювання було виконане у вертикальному напрямі, при цьому в копійованій формулі адреса стовпця не змінюється, але змінюється адреса рядка (див. рисунок 6.3.11).

	A	B	C	D	E	F
1	5					
2	6					
3	7					
4	8					
5	9					
6	10					
7	11					
8	12					
9	13					
10	14					

The image shows a spreadsheet with a formula bar at the top displaying $f_x = A2+1$. The active cell is A3, which contains the value 7. A vertical arrow on the right side of the spreadsheet is labeled "напрямок автозаповнення" (fill direction). A callout box points to the active cell A3 with the text "ім'я активної клітинки" (name of the active cell). Another callout box points to the formula bar with the text "скопійована в клітинку A3 формула" (copied formula to cell A3).

Рис.6.3.11. Приклад копіювання формули, що містить відносне посилання, у вертикальному напрямі

Введемо у клітинку A1 початкове значення 5, а у клітинку B1 формулу вигляду $=A1+1$ і виконаємо копіювання цієї формули у клітинку з C1 по J1, тобто виконаємо копіювання в горизонтальному напрямі з використанням маркера автозаповнення. У клітинку C1 буде скопійована формула $=B1+1$ і після натиснення клавіші *ENTER* обчислено значення 6, тобто при копіюванні формули в горизонтальному напрямі міняється адреса стовпця (буква), а адреса рядка (число, відповідне рядку) залишається незмінною (рис.6.3.11).

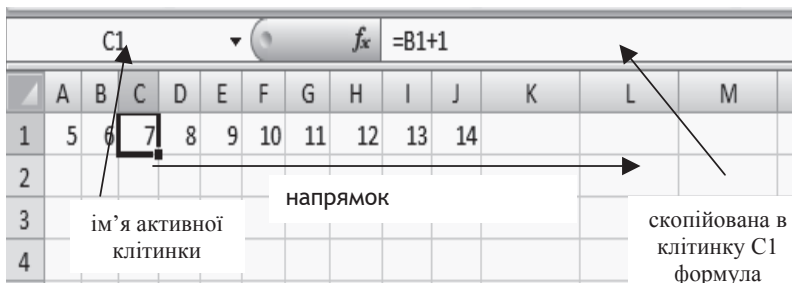



Рис.6.3.11. Приклад копіювання формули, що містить відносне посилання, в горизонтальному напрямі

6.3.3. Оформлення таблиць в MS Office Excel 2007

Розглянемо прийоми оформлення даних у вигляді таблиці.

1. Об'єднання клітинок. Спочатку треба виділити діапазон клітинок, які необхідно об'єднати. Об'єднати суміжні горизонтальні або вертикальні клітинки можна декількома способами, наприклад, перейти на вкладку *Основне*, вибрати *Вирівнювання*, а в ньому одну з команд списку *Об'єднати*

та розташувати в центрі  (команди представлено на рисунку 6.3.12) або виділити суміжні клітинки, потім викликати допоміжне меню, у якому слід вибрати команду *Формат клітинок...*. У діалоговому однойменному вікні, що з'явилось, слід перейти на вкладку *Вирівнювання*, потім у розділі *Відображення* поставити галочку напроти пункту *Об'єднання клітинок* і натиснути на кнопку *ОК* (див. рисунок 6.3.13).

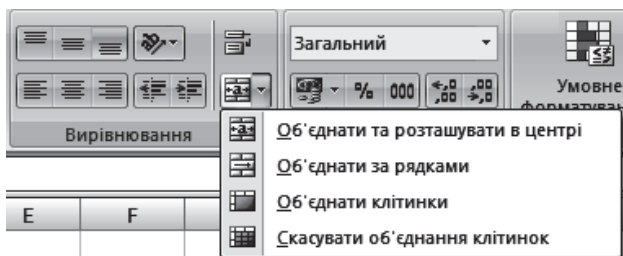


Рис.6.3.12. Команди списку Об'єднати та розташувати в центрі, що розкривається

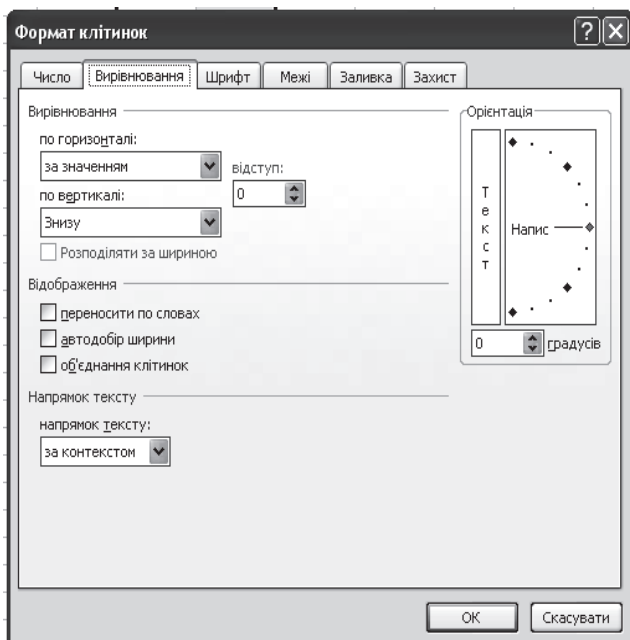


Рис. 6.3.13. Діалогове вікно Формат клітинок з активною вкладкою Вирівнювання

2. Установка точного значення ширини стовпця або висоти рядка. Для завдання точного значення ширини стовпця слід виділити весь стовпець, натиснувши ліву кнопку миші по букві, відповідній стовпцю. Потім

по виділеному стовпцю слід клацнути правою кнопкою миші і в допоміжному меню, що з'явилося, вибрати пункт *Ширина стовпця*. На екрані з'явиться однойменне діалогове вікно, в полі введення якого можна ввести значення, що встановлює ширину стовпця, і натиснути на кнопку *ОК* (рисунок 6.3.14 (ліворуч)). Аналогічно можна встановити висоту рядка. Для завдання точного значення висоти рядка слід виділити весь рядок, натиснувши ліву кнопку миші по числу, відповідному рядку. Потім по виділеному рядку слід клацнути правою кнопкою миші і в допоміжному меню, що з'явилося, вибрати пункт *Висота рядка*. На екрані з'явиться однойменне діалогове вікно, в полі введення якого можна ввести значення, що встановлює висоту рядка, і натиснути на кнопку *ОК* (рисунок 6.3.14 (праворуч)).

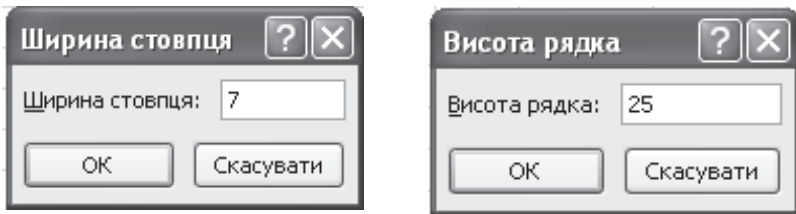


Рис. 6.3.14. Приклад установки точного значення ширини стовпця і висоти рядка

3. Зміна орієнтації тексту в клітинці. Для того щоб повернути текст у клітинці, наприклад, на 90 градусів, слід зробити активною клітинку, що містить текст, перейти на вкладку *Основне* в розділ *Вирівнювання* й у списку, що розкрився

Орієнтація  вибрати потрібну команду (див. рисунок 6.3.15).

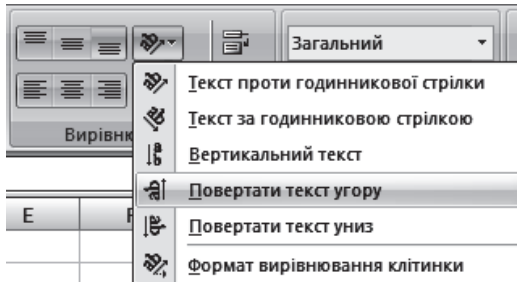


Рис.6.3.15. Команди списку Орієнтація, що розкривається

Якщо текст потрібно обернути на 52 градуси, то можна викликати діалогове вікно **Формат клітинок...**, або вибрати пункт **Формат вирівнювання клітинки**, або викликати допоміжне меню, натиснувши праву кнопку миші по активній клітинці і вибравши пункт **Формат клітинок**. У однойменному діалоговому вікні **Формат клітинок...** слід перемкнутися на вкладку **Вирівнювання** і в розділі **Орієнтація** встановити потрібне число градусів, як показано на рисунку 6.3.16, і натиснути на кнопку **OK**.

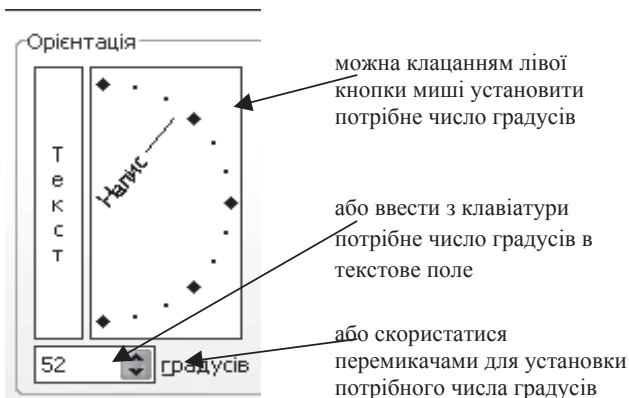

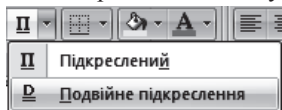


Рис.6.3.16. Розділ Орієнтація активної вкладки Вирівнювання діалогового вікна Формат клітинок...

4. Заливка клітинки, зміна шрифту, його кольору, накреслення й розміру. Усі ці дії можна виконати, перейшовши на вкладку *Основне* в розділ

Шрифт і вибравши відповідну команду (попередньо необхідно зробити активною потрібну клітинку): зробити жирним накреслення  потрібного

тексту, зробити курсивним накреслення  потрібного тексту,



підкреслити або двічі підкреслити потрібний текст

виконати заливку клітинки потрібним кольором, який можна вибрати в списку, що розкривається, або вибрати команду *Інші кольори...* для вибору іншого кольору, що приведе до відкриття діалогового вікна *Кольори* з двома вкладками *Звичайні* і *Спектр*. Перемикаючись між цими вкладками лівим клацанням миші, можна вибрати потрібний колір і натиснути на кнопку *ОК* (рисунок 6.3.17).

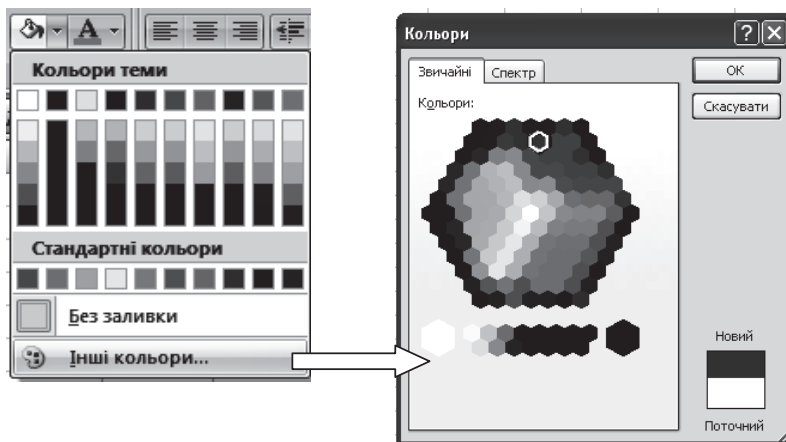



Рис.6.3.17. Приклад вибору кольора для заливання клітинки

Аналогічно можна змінити колір шрифту, використовуючи команду або розкривши список  для вибора кольору. Потрібний шрифт і його розмір можна вибрати з відповідних списків, що розкриваються, наведених на рисунку 6.3.18.

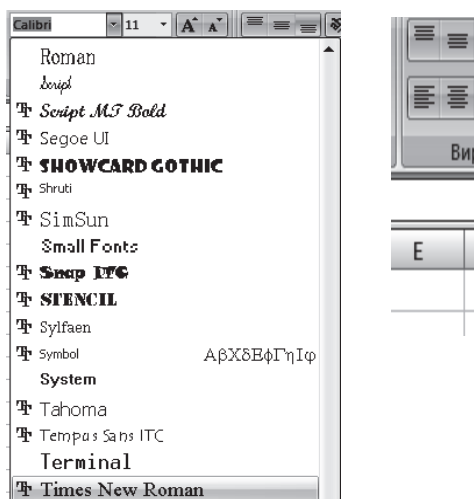


Рис.6.3.18. Приклад установки типа шрифту і його розміру

Всі перераховані вище команди можна виконати, натиснувши на активну клітинку правою кнопкою миші і викликавши контекстне меню, в якому треба вибрати команду *Формат клітинок...*, що приведе до появи однойменного діалогового вікна. У цьому діалоговому вікні слід перемкнутися на вкладку *Шрифт* або вкладку *Заливка* і виконати всі необхідні команди, після чого натиснути на кнопку *ОК* (див. рисунок 6.3.19).

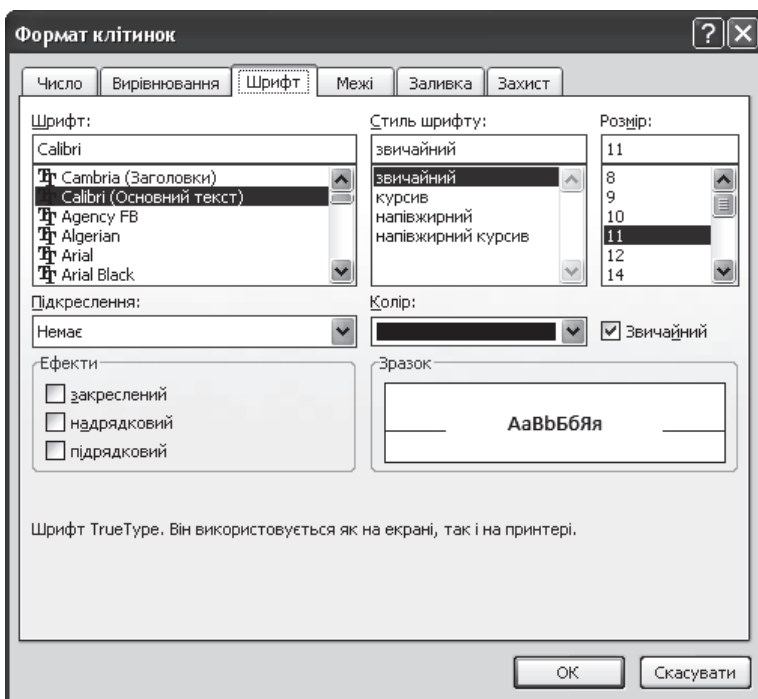



Рис.6.3.19. Діалогове вікно Формат клітинок з активною вкладкою Шрифт

5. Обрамлення клітинок (таблиці). Для обрамлення однієї клітинки або декількох клітинок (таблиці) слід виділити потрібні клітинки, потім перейти на вкладку *Основне* в розділ *Шрифт* і в списку Нижня межа , що розкрився, вибрати потрібне обрамлення. Варіанти обрамлення наведені на рисунку 6.3.20.

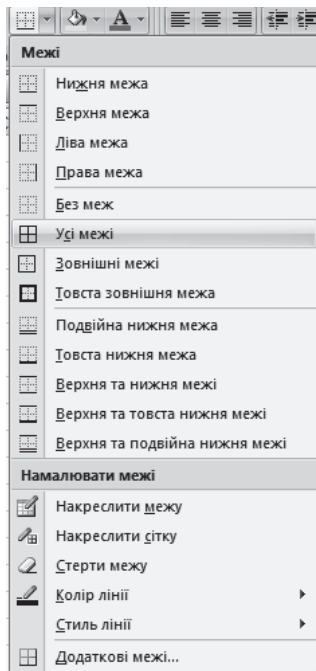


Рис. 6.3.20. Варіанти оформлення клітинок

Виконати оформлення таблиці можна, викликавши допоміжне меню, шляхом натиснення правої кнопки миші на вибраному діапазоні клітинок, потім вибрати команду *Формат клітинок...* і в однойменному діалоговому вікні, що з'явилося, перемкнутися на вкладку Межі, розділи якого показані на рисунку 6.3.21. Розділи цієї вкладки дозволяють встановити типи меж, їх колір, а також клацанням лівої кнопки миші встановити, до яких клітинок застосовувати вибраний тип оформлення. Після завершення всіх установок слід натиснути кнопку *ОК*.

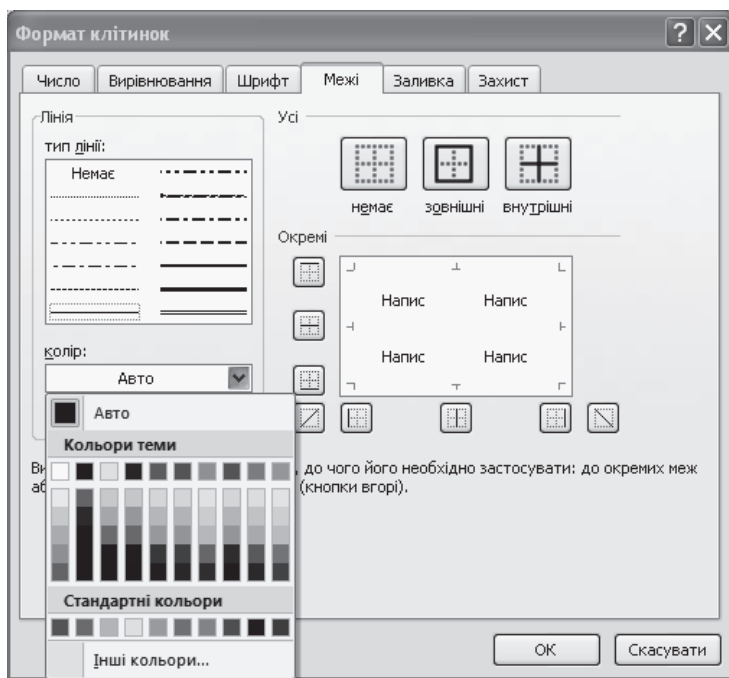


Рис.6.3.21. Діалогове вікно Формат клітинок з активною вкладкою Межі

Приклад 6.3.2. Заповнити таблицю, виконавши необхідні обчислення.

курс долара		
01.12.08	15.12.08	30.12.08
7,90	8,35	8,75

№	найменування товару	01.12.08			15.12.08			30.12.08			сума закупівлі за грудень (грн.)
		ціна закупу півлі в доларах	кількість (кг.)	сума закупівлі (грн.)	ціна закупу півлі в доларах	кількість (кг.)	сума закупівлі (грн.)	ціна закупу півлі в доларах	кількість (кг.)	сума закупівлі (грн.)	
1	ківі	2,91	20		2,82	25		2,55	35		
2	апельсини	2,39	24		2,42	30		2,49	27		
3	мандарини	1,49	28		1,54	28		1,68	40		

Виконання завдання. Спочатку оформимо у вигляді таблиць початкові дані. Для оформлення таблиці з вказаним курсом долара виділимо діапазон клітинок A1:C3. Перейдемо на вкладку Основне в розділ Шрифт і в списку кнопки Нижня межа, що розкривається, виберемо пункт Усі межі.

В результаті буде виконано необхідне обрамлення таблиці.

Виділимо діапазон клітинок A1:C1. На вкладці Основне перейдемо в розділ Вирівнювання і натиснемо на кнопку (за умовчанням) Об'єднати та розташувати в центрі або виберемо необхідну команду із списку, що розкривається, поряд з цією кнопкою. Після чого введемо текст «курс долара».

Виділимо клітинки з діапазону A2:C2 і встановимо для клітинок цього діапазону формат даних Дата. Для цього натиснемо праву кнопку миші по виділеному діапазону і в допоміжному вікні, що розкрилося, виберемо команду Формат клітинок..., виконання якої приведе до появи на екрані одностороннього діалогового вікна (рисунок 6.3.22). Перейдемо на вкладку Число (активна за умовчанням) і в розділі Числові формати виберемо Дата. У розділі Типи виберемо формат відображення дати, а також Мову (розташування). Якщо до установки формату у клітинки вводилися дані, то в розділі Зразок можна буде побачити їх відображення. Після виконання всіх налаштувань слід натиснути кнопку ОК і ввести у клітинки початкові дані – дати 01.12.08, 15.12.08, 30.12.08.

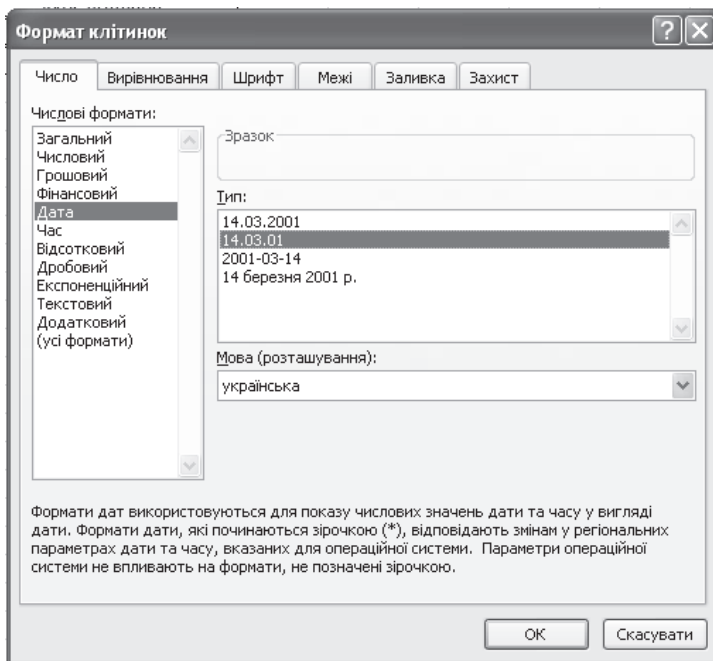


Рис.6.3.22. Приклад установки формату Дата

Виділимо діапазон клітинок A3:C3 і повторимо всі вищеписані дії для установки формату Грошовий, встановивши Кількість десяткових розрядів – 2, Позначення – грн. українська, Від’ємні числа – -1234,10 грн. Виконавши всі необхідні установки, слід натиснути на кнопку ОК, після чого ввести початкові дані завдання.

Аналогічно оформимо другу таблицю завдання, яка займе діапазон клітинок A5:L9. При заповненні клітинок, що містять текст, була виконана команда Формат клітинок... із завданням на вкладці Вирівнювання в розділі Відображення параметру Переносити після слів.

Для клітинок з діапазону C7:C9, F7:F9, I7:I9 слід встановити формат Грошовий, Позначення – \$ англійська (США). Для клітинок з діапазону E7:E9, H7:H9, K7:L9 слід встановити формат Грошовий, Позначення – грн. українська.

Після форматування таблиці в Excel мають наступний вигляд (рисунок 6.3.23).

	A	B	C	D	E	F	G	H	I	J	K	L
1	курс доллара											
2	01.12.08	15.12.08	30.12.08									
3	7,90 грн.	8,35 грн.	8,75 грн.									
4												
5			01.12.08		15.12.08			30.12.08			сума	
		найменування товару	ціна закупівлі в доларах	кількість (кг.)	сума закупівлі (грн.)	ціна закупівлі в доларах	кількість (кг.)	сума закупівлі (грн.)	ціна закупівлі в доларах	кількість (кг.)	сума закупівлі (грн.)	сума закупівлі за грудень (грн.)
6	№											
7	1	кiвi	\$2,91	20		\$2,82	25		\$2,55	35		
8	2	апелсини	\$2,39	24		\$2,42	30		\$2,49	27		
9	3	мандарини	\$1,49	28		\$1,54	28		\$1,68	40		

Рис.6.3.23. Приклад оформлення таблиць в MS Office Excel 2007

Сума закупівлі в гривнях обчислюється як добуток ціни закупівлі в доларах за один кілограм на кількість придбаного товару і на поточний курс долара. Для виконання необхідних обчислень у клітинці E7 введемо формулу вигляду $=C7*A3*D7$. Оскільки для заповнення клітинок з діапазону E8:E9 використовуватиметься вертикальне автозаповнення, то необхідно заборонити змінюватися індексу рядка в посиланні A3. Для цього перед номером рядка був поставлений знак \$. Аналогічно введемо у клітинку H7 формулу $=F7*B3*G7$ і виконаємо вертикальне автозаповнення для копіювання формули у клітинки H8, H9. Введемо у клітинку K7 формулу $=I7*C3*J7$ і виконаємо вертикальне автозаповнення для копіювання формули у клітинки K8, K9.

Для обчислення суми закупівлі за грудень по кожному найменуванню продукції введемо у клітинку L7 формулу $=E7+H7+K7$ і виконаємо вертикальне автозаповнення для копіювання цієї формули у клітинки L8, L9.

Результат виконання обчислень наведений на рисунку 6.3.24.

	A	B	C	D	E	F	G	H	I	J	K	L
1	курс доллара											
2	01.12.08	15.12.08	30.12.08									
3	7,90 грн.	8,35 грн.	8,75 грн.									
4												
5			01.12.08			15.12.08			30.12.08			
6	№	найменування товару	ціна закупівлі в доларах	кількість (кг.)	сума закупівлі (грн.)	ціна закупівлі в доларах	кількість (кг.)	сума закупівлі (грн.)	ціна закупівлі в доларах	кількість (кг.)	сума закупівлі (грн.)	сума закупівлі за грудень (грн.)
7	1	ківі	\$2,91	20	459,78 грн.	\$2,82	25	588,68 грн.	\$2,55	35	780,94 грн.	1 829,39 грн.
8	2	апельсини	\$2,39	24	453,14 грн.	\$2,42	30	606,21 грн.	\$2,49	27	588,26 грн.	1 647,62 грн.
9	3	мандарини	\$1,49	28	329,59 грн.	\$1,54	28	360,05 грн.	\$1,68	40	588,00 грн.	1 277,64 грн.

Рис.6.3.24. Приклад виконання обчислень в MS Office Excel 2007

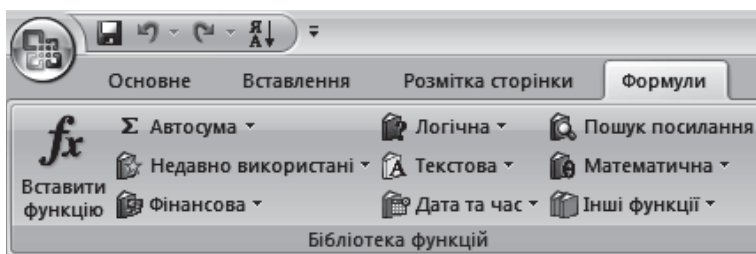
6.3.4. Робота з функціями і формулами в MS Office Excel 2007

➤ **Функції в Microsoft Excel** – це формули з одним або декількома аргументами. Вставити функцію у клітинку можна декількома способами: 1) натиснути комбінацію клавіш Shift+F3; 2) натиснути на кнопку *Вставка функції* справа в рядку формул, розташованому нижче за вкладки головного

меню і стрічки;



3) вибрати вкладку *Формули*, натиснувши на кнопку *Вставити функцію* в розділі *Бібліотека функцій* або вибрати в цьому розділі категорію (фінансова, логічна, математична і тому подібне), а в ній конкретну функцію.



В результаті на екрані з'явиться діалогове вікно *Вставка функції*, представлено на рисунку 6.3.25.

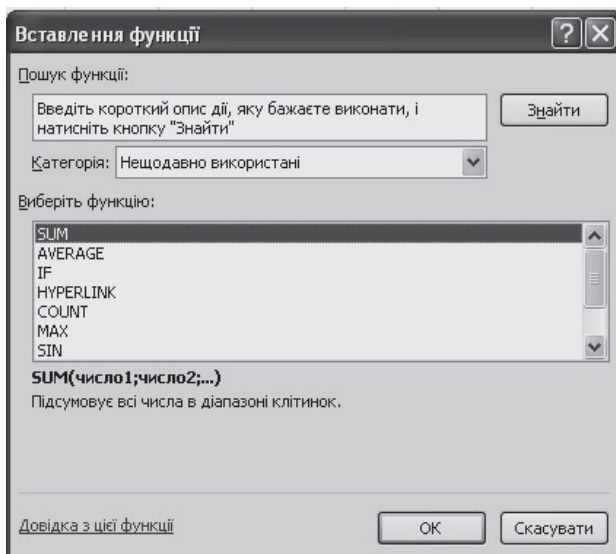


Рис.6.3.25. Діалогове вікно Вставлення функції

Розглянемо категорії і відповідні їм функції.

Категорія **Математичні функції** містить деякі наступні основні функції.

Назва функції (аналог в російськомовній версії)	Призначення функції з категорії Математичні функції
1	2
ABS (число)	по заданому числу повертає його абсолютне значення
ACOS (число)	повертає арккосинус числа в радіанах в діапазоні від 0 до π
ASIN (число)	повертає арксинус числа в радіанах від $-\pi/2$ до $\pi/2$
ATAN (число)	повертає арктангенс числа в радіанах від $-\pi/2$ до $\pi/2$
COS (число)	повертає косинус числа
DEGREES (кут) (ГРАДУСИ(радиан))	перетворює радіани в градуси
EXP (число)	повертає експоненту заданого числа
FACT (число)	повертає факторіал заданого числа

(ФАКТР(число))	
GCD(число1;число2;...) (НОД(число1;число2;...))	повертає найбільший спільний дільник заданих чисел
LSM(число1;число2;...) (НОК(число1;число2;...))	повертає найменше спільне кратне чисел
LN(число)	повертає натуральний логарифм числа
LOG(число;основа)	повертає логарифм числа із заданою основою
MDETERM(масив) (МОПРЕД(квадратный массив))	повертає визначник матриці
MINVERSE(масив) (МОБР(квадратный массив))	повертає зворотну матрицю по заданій квадратній матриці
MMULT(масив1;масив2) (МУМНОЖ(массив1;массив2))	повертає добуток матриць, що зберігаються в двох масивах, при цьому число стовпців матриці з першого масиву повинне збігатися з числом рядків матриці з другого масиву. Результуюча матриця матиме число рядків, рівне числу рядків матриці з першого масиву і число стовпців, рівне числу стовпців матриці з другого масиву
MOD(число;дільник) (ОСТАТ(делимое;делитель))	повертає залишок від ділення
MROUND(число;точність) (ОКРУГЛ(число;число разрядов))	повертає число, округлене із заданою точністю
POWER(число;ступінь) (СТЕПЕНЬ(основа; показатель_степени))	повертає результат зведення заданого числа в задану ступінь, замість цієї функції можна використовувати оператор ^ (число^ступінь)
PRODUCT(число1;число2;...) (ПРОИЗВЕД(число1;число2;...))	повертає добутки заданих чисел (всіх аргументів)
RADIANS(кут) (РАДИАНЫ(градусов))	перетворює градуси в радіани
RANDBETWEEN(нижн_межа;верх_межа) (СЛУЧМЕЖДУ(нижн_граница; верхн_граница))	повертає випадкове число між двома вказаними числами
ROUND(число;кількість_розрядів) (ОКРУГЛ(число; число_разрядов))	округлює число до заданої кількості десяткових знаків. Якщо кількість_розрядів менше 0, то початкове число округляється зліва від десяткової точки

SERIESSUM (x;n;m;коєфіцієнти) (РЯД.СУММ (x;n;m;коєфіцієнти))	повертає суму степінного ряду на основі формули
SIGN (число) (ЗНАК (число))	повертає знак числа
SIN (число)	повертає синус кута
SQRT (число) (КОРЕНЬ (число))	повертає квадратний корінь числа
SUBTOTAL (номер_функції; посилання1;.) (ПРОМЕЖУТОЧНЫЕ.ИТОГИ (номер_функции; столбец1;))	повертає проміжний підсумок в список або базу даних, аргумент номер_функції набуває значень від 1 до 11 і указує за допомогою якої функції буде обчислений проміжний підсумок. Наприклад, 1 – AVERAGE, 2 – COUNT, 3 – COUNTA, ..., 6 – PRODUCT, ..., 10 – VAR, 11 – VARP.
SUM (число1;число2;.) (СУММ (число1;число2;))	повертає суму всіх аргументів
SUMIF (діапазон;критерій;діапазон_для_суми) (СУММЕСЛИ (діапазон_проверки;критерій; діапазон суммирования))	обчислює суму значень тих клітинок з діапазон_для_суми, відповідні яким клітинки з діапазон задовольняють критерію.
SUMPRODUCT (масив1;масив2;масив3;.) (СУММПРОИЗ (діапазон1;діапазон2; діапазон3;))	повертає суму добутків відповідних елементів масивів або діапазонів, при цьому числові масиви повинні мати однакову розмірність
SUMSQ (число1;число2;число3;.) (СУММКВ (число1;число2;число3;))	повертає суму квадратів аргументів
SUMX2MY2 (масив_x;масив_y) (СУММПРАЗНКВ (діапазон_x;діапазон_y))	повертає суму різниць квадратів відповідних значень двох масивів, при цьому числові масиви повинні мати однакову розмірність
SUMX2PY2 (масив_x;масив_y) (СУММСУММКВ (діапазон_x;діапазон_y))	повертає суму сум квадратів відповідних значень двох числових масивів, при цьому числові масиви повинні мати однакову розмірність
SUMXMY2 (масив_x;масив_y) (СУММКВРАЗН (діапазон_x;діапазон_y))	повертає суму квадратів різниць відповідних значень двох масивів, при цьому числові масиви повинні мати однакову розмірність
TAN (число)	повертає тангенс кута
TRUNC (число;кількість_розрядів) (ОТБР (число;число_разрядов))	відкидає дробову частину числа, залишаючи тільки цілу частину

При вирішенні систем лінійних рівнянь, що часто виникають в багатьох економічних задачах, корисними виявляються математичні функції роботи з матрицями.

Пример 6.3.3. Для трьохгалузевої економічної системи задані матриця коефіцієнтів прямих матеріальних витрат $A = \begin{pmatrix} 0,3 & 0,1 & 0,4 \\ 0,2 & 0,5 & 0,0 \\ 0,3 & 0,1 & 0,2 \end{pmatrix}$ і вектор

кінцевої продукції $Y = \begin{pmatrix} 200 \\ 100 \\ 300 \end{pmatrix}$. Знайти вектор валової продукції.

Виконання завдання. Величини валової продукції трьох галузей, тобто вектор X , обчислюється згідно моделі «витрати-випуск» за формулою $X = (E - A)^{-1}Y$, де E – одинична матриця. Введемо елементи одиничної матриці в діапазоні A1:C3, а елементи матриці A – в діапазоні E1:G3. Елементи результуючої матриці $E - A$ розташуємо в діапазоні C5:E7. Для цього введемо формулу =A1-E1 в клітинку C5, натиснемо клавішу ENTER, а потім виконаємо автозаповнювання в горизонтальному напрямі до клітинки E5, а потім в вертикальному напрямі до клітинки E7, як показано на рисунку 6.3.26.

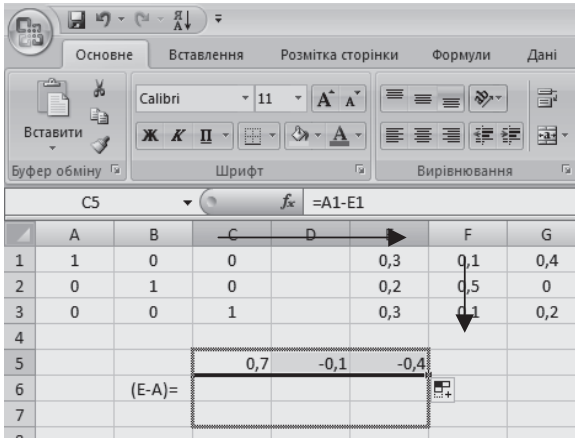



Рис.6.3.26. Обчислення матриці $E - A$

Обчислимо зворотно матрицю $(E - A)^{-1}$ за допомогою функції MINVERSE із категорії Математичні функції. Для цього спочатку виділимо діапазон H5:J7, призначений для розміщення елементів зворотної матриці.

Натиснемо на кнопку *Вставка функції* . У діалоговому однойменному вікні, що з'явилося, виберемо категорію *Математичні*, а в цій категорії – функцію MINVERSE. У результаті на екрані з'явиться діалогове вікно *Аргументи функції*, представлене на рисунку 6.3.27. У робочому полі Массив цього вікна слід увести діапазон C5:E7 і для розміщення елементів зворотної матриці не в окремій клітинці, а у виділеному діапазоні слід натиснути комбінацію клавіш Ctrl+Shift+ENTER. Якщо елементи зворотної матриці не з'явилися в клітинках виділеного діапазону, то слід клацнути лівою кнопкою миші в рядку формул і знову натиснути комбінацію клавіш Ctrl+Shift+ENTER.

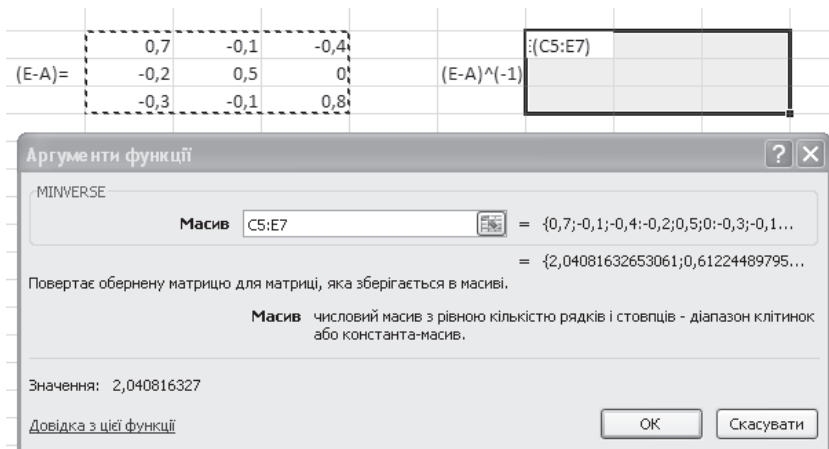



Рис. 6.3.27. Діалогове вікно Аргументи функції MINVERSE

Розташуємо елементи вектора кінцевої продукції в діапазоні M5:M7, виділимо діапазон H9:H11 для розміщення елементів вектора валової продукції X . Натиснемо на кнопку *Вставка функції* . У однойменному діалоговому вікні, що з'явилося, виберемо категорію *Математичні*, а в цій категорії – функцію MMULT. У результаті на екрані з'явиться діалогове вікно *Аргументи функції*, представлене на рисунку 6.3.28. У робочому полі Массив1 цього вікна слід увести діапазон H5:J7, а в робочому полі Массив2 – M5:M7 і для розміщення елементів вектора валової

продукції не в окремій клітинці, а у виділеному діапазоні слід натиснути комбінацію клавіш Ctrl+Shift+ENTER. Якщо елементи вектора валової продукції не з'явилися в клітинках виділеного діапазону, то слід клацнути лівою кнопкою миші в рядку формул і знову натиснути комбінацію клавіш Ctrl+Shift+ENTER.

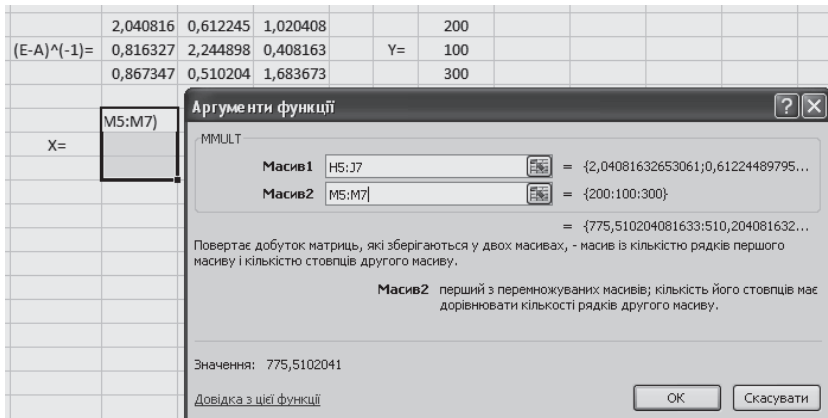


Рис.6.3.28. Диалогове вікно Аргументи функції MMULT

Категорія **Статистичні функції** містить деякі наступні основні функції.

Назва функції (аналог в російськомовній версії)	Призначення функції з категорії Статистичні функції
1	2
AVERAGE (число1;число2;...) (СПОТКЛ) (число1;число2;...))	повертає середнє абсолютних значень відхилень точок даних від середнього
AVERAGE (значення1;значення2;...) (СРЗНАЧ) (число1;число2;...))	повертає середнє (арифметичне) значенє аргументів
AVERAGEIF (діапазон;критерій; діапазон_середн) (СРЗНАЧЕСЛИ) (диапазон; условие; диапазон усреднения))	повертає середнє (арифметичне) для клітинок, визначених критерієм або умовою
BINOMDIST (кількість_успіхів; кількість_випробувань;...) (БИНОМРАСП) (число_успехов; число испытаний;...))	повертає окреме значення біноміального розподілу
CONFIDENCE (альфа;станд відхіл;	повертає довірчий інтервал для

розмір) (ДОВЕРИТ (альфа;станд_откл; размер выборки))	математичного очікування (середнього) генеральної сукупності
CORREL (масив1;масив2) (КОРРЕЛ (массив1;массив2))	повертає коефіцієнт кореляції між двома сукупностями даних (вибірками)
COUNT (значення1;значення2;.) (СЧЁТ (значения1;значения2;))	повертає кількість числових значень серед аргументів
COUNTA (значення1;значення2;.) (СЧЕТЗ (значения1;значения2;))	повертає кількість непорожніх клітинок в діапазоні (серед значень аргументів)
COUNTBLANK (діапазон) (СЧИТАТЬПУСТОТЫ (диапазон))	повертає кількість порожніх клітинок в діапазоні
COUNTIF (діапазон;критерій) (СЧЕТЕСЛИ (диапазон;критерий))	повертає кількість непорожніх клітинок діапазону, що задовольняють заданому критерію
COVAR (масив_x;масив_y) (КОВАР (массив1;массив2))	повертає коваріацію – середнє попарних добутків відхилень
EXPONDIST (x;лямбда;функція) (ЭКСПРАСП (x;лямбда;интегральная))	повертає експоненціальний розподіл
FDIST (x;ступні_вільності1; ступні_вільності2) (ФРАСП (x; степени свободы1; степени свободы2))	повертає F-розподіл вірогідності (ступінь відхилень) для двох наборів даних
FISHER (x) (ФИШЕР (x))	повертає перетворення Фішера
FORECAST (x;відомі_значення_y; відомі_значення_x) (ПРЕДСКАЗ (x;известные значения_y; известные значения x))	повертає прогнозоване (майбутнє) значення у(x) по відомих значеннях у і x з використанням лінійної регресії
FREQUENCY (масив_даних; масив_інтервалів) (ЧАСТОТА (массив данных; массив интервалов))	повертає частоти попадання елементів вибірки у вказані інтервали і повертає масив чисел
FTEST (масив1;масив2) (ФТЕСТ (массив1;массив2))	повертає результат F-теста
GAMMADIST (x;альфа;бета;функція) (ГАММАРАСП (x;альфа;бета;интегральная))	повертає гамма-розподіл
GEOMEAN (число1;число2;.) (СРГЕОМ (число1;число2;))	повертає середнє геометричне аргументів – ряда позитивних чисел
HARMEAN (число1;число2;.) (СРГАРМ (число1;число2;))	повертає середнє гармонійне аргументів
HYPGEOMDIST (кількість успіхів у вибірці;	повертає гіпергеометричний розподіл


розмір вибірки;...) (ГИПЕРГЕОМЕТ (число_успехов_ в_выборке; размер_выборки;...))	
INTERCEPT (відомі_значення_у; відомі_значення_х) (ОТРЕЗОК (известные_значения_у; известные_значения_х))	повертає точку перетину осі Y з лінією лінійної регресії, обчисленої на основі відомих значень x і y.
KURT (число1;число2;...) (ЭКЦЕСС (число1;число2;...))	повертає ексцес сукупності даних
LARGE (масив;k) (НАИБОЛЬШИЙ (массив;k))	повертає k-е найбільше значення в масиві даних
LINEST (відомі_значення_у; відомі_значення_х;конст;статистика) (ЛИНЕЙН (известные_значения_у; известные_значения_х;конст;))	повертає параметри лінійного наближення, обчислені за методом найменших квадратів
LOGEST (відомі_значення_у; відомі_значення_х;конст;статистика) (ЛГРФПРИБЛ (известные_значения_у; известные_значения_х;...))	повертає параметри експоненціального наближення
LOGNORMDIST (x;середне; стандартне_відхил) (ЛОГНОРМРАСП (x;среднее; стандартное_откл))	повертає інтегральний логарифмічно-нормальний розподіл x
MAX (число1;число2;...) (МАКС (число1;число2;...))	повертає найбільше значення із списку аргументів
MEDIAN (значення1;значення2;...) (МЕДИАНА (число1;число2;...))	повертає медіану вказаних значень аргументів
MIN (число1;число2;...) (МИН (число1;число2;...))	повертає найменше значення із списку аргументів
MODE (число1;число2;...) (МОДА (число1;число2;...))	повертає моду сукупності даних
NORMDIST (x;середне; стандартне_відхил;функція) (НОРМРАСП (x;среднее; стандартное_откл;лог_интегральная))	повертає нормальний розподіл з вказаними параметрами
NORMSDIST (z) (НОРМСТРАСП (z))	повертає стандартний нормальний інтегральний розподіл (з нульовим середнім і одиничним стандартним відхиленням)
PEARSON (масив1;масив2)	повертає коефіцієнт кореляції Пірсона
PERCENTILE (масив;k) (ПЕРСЕНТИЛЬ (массив;k))	повертає k-й процентиль для значень масиву (діапазону)
PERMUT (число;кількість_вибраних)	повертає кількість перестановок для

(ПЕРЕСТ (число;число_выбранных))	заданої кількості об'єктів, яку можна вибрати із спільної кількості об'єктів
POISSON (x;середне;функція) (ПУАССОН (x;середнее;интегральная))	повертає розподіл Пуассона
PROB (інтервал_x; інтервал_імовірностей;нижня_межа;.) (ВЕРОЯТНОСТЬ (x_интервал; интревал_вероятностей; нижний_предел;.))	повертає вірогідність того, що значення з інтервалу знаходиться у вказаних межах. Якщо значення верхня_межа не задане, то функція повертає вірогідність того, що значення з інтервал_x дорівнюють значенню аргументу нижня_межа
QUARTILE (масив;частка) (КВАРТИЛЬ (масив;часть))	повертає квартиль вибірки (масиву)
RANK (число;посилання;порядок) (РАНГ (число;ссылка;порядок))	повертає ранг числа в списку чисел
RSQ (відомі_значення_u; відомі_значення_x) (КВПИРСОН (известные_значения_u; известные значения x))	повертає квадрат коефіцієнта кореляції Пірсона, обчисленого по вказаних значеннях (точкам)
SKEW (число1;число2;...) (СКОС (число1;число2;...))	повертає асиметрію розподілу
SLOPE (відомі_значення_u; відомі_значення_x) (НАКЛОН (известные_значения_u; известные значения x))	повертає нахил лінії лінійної регресії по вказаних значеннях даних (точкам)
SMALL (масив;k) (НАИМЕНЬШИЙ (массив;k))	повертає k-е найменше значення в сукупності даних
STANDARDIZE (x;середне стандартне_відхил) (НОРМАЛІЗАЦІЯ (x;среднее; стандартное откл))	повертає нормалізоване значення для розподілу з вказаними параметрами – середнім і стандартним відхиленням
STDEV (число1;число2;...) (СТАНДОТКЛОН (число1;число2;...))	повертає стандартне відхилення на основі вибірки
STDEVР (число1;число2;...) (СТАНДОТКЛОНП (число1;число2;...))	повертає стандартне відхилення на основі генеральної сукупності
STEYX (відомі_значення_u; відомі_значення_x) (СТОШУХ (известные_значения_u; известные значения x))	повертає стандартну помилку прогнозованих значень у для кожного значення u регресії
TDIST (x;ступені_вільності;боки) (СТЬЮДРАСП (x;степени_свободы;хвосты))	повертає t-розподіл Стюдента
TREND (відомі_значення_u; відомі_значення_x;нові_значення_x;...) (ТЕНДЕНЦІЯ (известные значения u; известные значения x;новые значения x;...))	повертає значення щодо лінійного наближення, обчисленого за методом найменших квадратів на основі

известные значения $x; \dots$)	відомих даних
TTEST (масив1;масив2;боки;тип) (TTEST (массив1;массив2;хвосты;тип))	повертає ймовірність, яка відповідає t-тесту Стьюдента
VAR (число1;число2;.) (ДИСП (число1;число2;))	повертає дисперсію по заданій вибірці (ігноруючи логічні значення і текст)
VARP (число1;число2;.) (ДИСПР (число1;число2;))	повертає дисперсію по заданій генеральній сукупності (ігноруючи логічні значення і текст)
WEIBULL (х;альфа;бета;функція) (ВЕЙБУЛЛ (х;альфа;бета;интегральная))	повертає розподіл Вейбулла

Приклад 6.3.4. Обчислити вибіркові середнє і дисперсію для наступної вибірки: 4, 1, 4, 5, 1, 4, 13, 10, 2, 7, 4, 2, 2, 4, 6, 4, 5, 6, 2, 4.

Виконання завдання. Уведемо елементи вибірки в діапазон A1:A20.

Натиснемо на кнопку *Вставлення функції* . В однойменному діалоговому вікні, що з'явилося, виберемо категорію Статистичні, а в цій категорії – функцію AVERAGE для обчислення вибіркового середнього. В результаті на екрані з'явиться діалогове вікно *Аргументи функції*, представлене на рисунку 6.3.29. В робочому полі Значення1 цього діалогового вікна слід ввести діапазон A1:A20 і натиснути на кнопку *OK* для появи результату, рівного 4,5.

Аналогічно можна обчислити вибіркoву дисперсію. Для цього в категорії Статистичні функції слід вибрати функцію VAR.

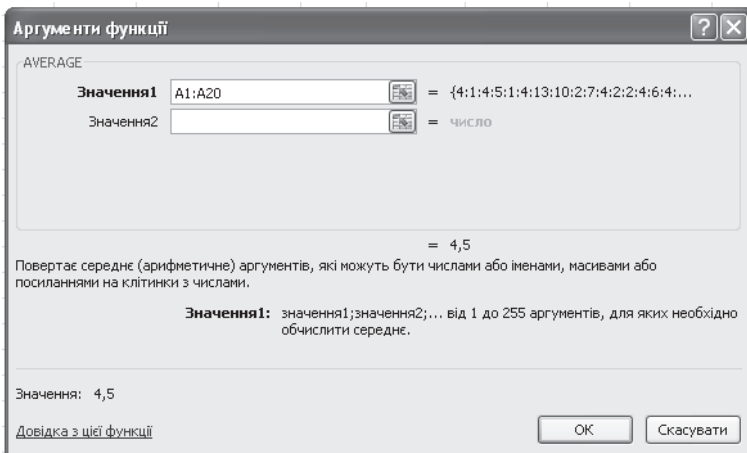


Рис. 6.3.29. Діалогове вікно Аргументи функції AVERAGE

Категорія **Дата й час** містить деякі наступні основні функції.

Назва функції (аналог в російськомовній версії)	Призначення функції з категорії Дата й година
1	2
DAY (дата_у_числовому_форматі) (ДЕНЬ (дата в числовом форматі))	Повертає день місяця (число від 1 до 31)
DAYS360 (поч_дата;кін_дата;метод) (ДЕНЬ360 (нач_дата;кон_дата;метод))	Повертає кількість днів між двома датами на основі 360-денного року (12 місяців по 30 днів кожен). Ця функція часто використовується в бухгалтерських розрахунках
HOURL (час_у_числовому_форматі) (ЧАС (время в числовом форматі))	Повертає годину (число від 0 до 23)
MINUTE (час_у_числовому_форматі) (МИНУТЫ (время в числовом формате))	Повертає хвилини (число від 0 до 59)
MONTH (дата_у_числовому_форматі) (МЕСЯЦ (дата в числовом формате))	Повертає місяць (число від 1 (січень) до 12 (грудень))
NETWORKDAYS (поч_дата;кінц_дата; свята) (ЧИСТРАБДНИ (нач_дата; кон_дата; праздники))	Повертає кількість цілих робочих днів між двома датами
NOW () (ТДАТА ())	Функція без параметрів, яка повертає поточну дату і час у форматі дати і часу

SECOND (час_у_числовому_форматі) (СЕКУНДЫ (время в числовом формате))	повертає секунди (число від 0 до 59)
TIME (години;хвилини;секунди) (ВРЕМЯ (часы;минуты;секунды))	повертає години, хвилини і секунди, задані як числа, у вигляді числа, записаного в коді часу Excel
TODAY() (СЕГОДНЯ())	функція без параметрів повертає поточну дату у форматі дати
WEEKDAY (дата_в_числовому_форматі; тип) (ДЕНЬНЕД (дата_в_числовом_формате; тип))	повертає число від 1 до 7, відповідне дню тижня для вказаної дати
WEEKNUM (порядковий_номер; тип_повернення) (НОМНЕДЕЛИИ (пор номер;тип возвр))	повертає номер тижня в році
YEAR (дата_в_числовому_форматі) (ГОД (дата в числовом формате))	повертає рік, тобто ціле число від 1900 до 9900.

Приклад 6.3.5. По даті народження співробітників обчислити їхній вік:

П.І.Б.	Дата народження
Акулич П.М.	17.03.1969
Іваненко А.А.	12.07.1982
Петренко І.В.	03.01.1955
Сідоренко А.П.	24.05.1976
Яковенко Р.П.	13.02.1958

Виконання завдання. Створимо таблицю в Excel, додавши стовпець «Вік». Заповнимо стовпець «Вік», використовуючи функції YEAR і TODAY, за допомогою формули (YEAR((TODAY())-адреса клітинки, у якій міститься дата народження співробітника)-1900. Таблиця й розрахунки наведені на рисунках 6.3.30, 6.3.31.

У стовпці «Вік» попередньо слід установити формат Числовий із числом десяткових розрядів, рівним нулю, а в стовпці «Дата народження» - формат Дата.

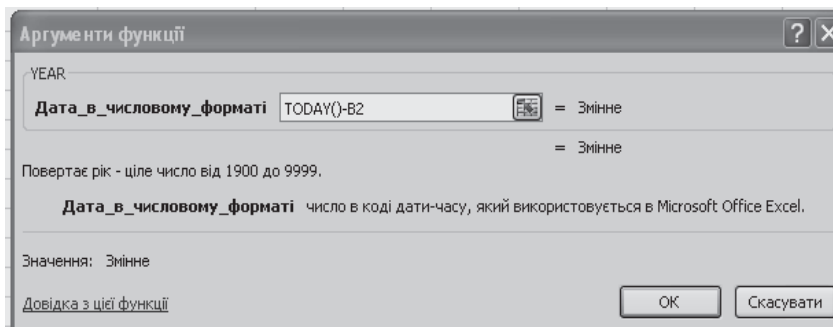


Рис.6.3.30 Діалогове вікно Аргументи функції YEAR

YEAR						
=YEAR(TODAY()-B2)-1900						
	A	B	C	D	E	F
1	П.І.Б.	Дата народження	Вік			
2	Акулич П.М.	17.03	=YEAR(TODAY()-B2)-1900			
3	Іваненко А.А.	12.07.1982				
4	Петренко І.В.	03.01.1955				
5	Сідоренко А.П.	24.05.1976				
6	Яковенко Р.П.	13.02.1958				

Рис.6.3.31. Приклад вводу формули для обчислення віку співробітників

Після введення формули в клітинку C2 і натискання клавіші ENTER необхідно, використовуючи автозаповнення, скопіювати цю формулу в діапазон C3:C6.

Категорія **Текстові** містить деякі наступні основні функції.

Назва функції (аналог в російськомовній версії)	Призначення функції з категорії Текстові
1	2
CHAR (число) (СИМВОЛ (число))	повертає символ з указаним кодом (параметр число), встановленим для комп'ютера (ANSI для Windows)
CLEAN (текст)	видаляє з тексту всі недруковані

ПЕЧСИМВ (текст))	символи
CODE (текст) (КОДСИМВ (текст))	повертає код першого символу в тексті згідно з таблицею символів, використовуюваною на комп'ютері (ANSI для Windows)
CONCATENATE (текст1;текст2,...) (СЦЕПИТЬ (текст1;текст2,...))	поєднує декілька текстів (замість цієї функції можна використовувати оператор &, тобто, наприклад, текст1&текст2)
DOLLAR (число;десяткові_знаки) (РУБЛЬ (число;число_знаков))	перетворює число на текст, використовуючи грошовий формат
EXACT (текст1; текст2) (СОВПАД (текст1; текст2))	порівнювання двох текстів (строк) з поверненням результату ІСТИНА, якщо тексти співпадають і результату – ХИБНІСТЬ, якщо тексти не співпадають, причому великі і малі букви розрізняються.
FIND (шуканий_текст;текст_перегляду; поч_позиція) (ПОИСК (искомий_текст; текст_для_поиска; нач_позиция))	повертає позицію шуканого текстового рядка в тексті перегляду, який його містить, причому великі і малі букви розрізняються.
FIXED (число;кількість_знаків; без_розділювачів) (ФИКСИРОВАННЫЙ (число; число_знаков; лог_без_разделителей))	повертає округлене до заданої кількості десяткових знаків число і перетворює його на текст
LEFT (текст;кількість_символів) (ЛЕВСИМВ (текст; количество_знаков))	повертає задану кількість символів, вибраних з початку тексту
LEN (текст) (ДЛСТР (текст))	повертає довжину тексту – кількість символів в тексті
LOWER (текст) (СТРОЧН (текст))	Перетворює всі букви в тексті на малі
MID (текст; поч_позиція; кількість_символів) (ПСТР (текст;начальная_позиция; число_знаков))	повертає задану кількість символів, вибрану з тексту, починаючи з указанної позиції
PROPER (текст) (ПРОПНАЧ (текст))	перетворює першу букву кожного слова в тексті на велику, а решту букв – на малі
REPLACE (старий_текст;поч_поз;	замінює вказану кількість символів

кількість_символів;новий_текст) (ЗАМЕНИТЬ(старый_текст; нач_поз; число_знаков; новый_текст))	тексту на інший текст
REPT(текст; кількість_повторів) (ПОВТОР(текст; число_повторений))	повторює текст задану кількість разів
RIGHT(текст;кількість_символів) (ПРАВСИМВ(текст;количество_знаков))	повертає задану кількість символів, вибрану з кінця тексту
SEARCH(шуканий_текст; текст_перегляду; поч_позиція) (ПОИСК(искомый_текст; текст_для_поиска; нач_позиция))	повертає позицію першого входження символу або тексту (без урахування регістру)
SUBSTITUTE(текст;стар_текст; нов_текст;номер_входження) ЗАМЕНИТЬ(старый_текст; нач_поз; число_знаков;новый_текст)	замінює старий текст на новий
TRIM(текст) (СЖПРОБЕЛЫ(текст))	видаляє з тексту всі, крім одиночних, пробіли між словами
UPPER(текст) (ПРОПИСН(текст))	перетворює всі букви на великі

Категорія **Логічні** містить деякі наступні основні функції.

Назва функції (аналог в російськомовній версії)	Призначення функції з категорії Текстові
1	2
AND(лог_значення1;лог_значення2;...) (И(лог_значение1;лог_значение2;...))	повертає значення ІСТИНА, якщо всі аргументи функції мають значення ІСТИНА, інакше повертає значення ХИБНІСТЬ
FALSE() (ЛОЖЬ())	повертає логічне значення ХИБНІСТЬ
IF(лог_вираз;значення_якщо_істина; значення_якщо_хибність) (ЕСЛИ(лог_выражение; значение_если_истина;...))	повертає одне значення (значення_ якщо_істина), якщо лог_вираз має значення ІСТИНА і друге значення (значення_якщо_хибність), якщо лог_вираз має значення ХИБНІСТЬ
NOT(лог_значення) (НЕ(логическое_выражение))	змінює значення ІСТИНА на ХИБНІСТЬ, значення ХИБНІСТЬ на ІСТИНА
OR(лог_значення1;лог_значення2;...)	повертає значення ХИБНІСТЬ, якщо всі

(ИЛИ (лог_значеніе1;лог_значеніе2;...))	аргументи мають значення ХИБНІСТЬ, інакше повертає значення ІСТИНА
TRUE() (ИСТИНА())	повертає логічне значення ІСТИНА

Категорія **Фінансові** містить деякі наступні основні функції.

Назва функції (аналог в російськомовній версії)	Призначення функції з категорії Текстові
1	2
ACCRINT (дата_випуску; перша_виплата; дата_угоди; ставка;...) (НАКОПДОХОД (дата_випуска; первый_доход; дата_согл))	повертає накопичений відсоток за цінними паперами з періодичною виплатою відсотків
ACCRINTM (дата_випуску; дата_угоди; ставка; номінал; базис) (НАКОПДОХОДПОГАШ (дата_випуска; дата_согл; ставка;...))	повертає накопичений відсоток за цінними паперами із виплатою відсотків у момент погашення
AMORDEGRC (поч_вартість; дата_придбан; перш_період;...) (АМОПУВ (стоимость; дата_приобр; первый_період))	повертає пропорційну лінійну амортизацію активів для кожного звітного періоду
COUPDAYBS (дата_угоди; дата_погаш; частота; базис) (ДНЕЙКУПОНДО (дата_согл; дата_вступл_в_силу_частота; базис))	повертає кількість днів від початку купонного періоду до дня розрахунку
COUPDAYS (дата_угоди; дата_погаш; частота; базис) (ДНЕЙКУПОН (дата_согл; дата_вступл_в_силу_частота; базис))	повертає кількість днів у купонному періоді, який містить дату розрахунку
COUPNUM (дата_угоди; дата_погаш; частота; базис) (ЧИСЛКУПОН (дата_согл; дата_вступл_в_силу_частота; базис))	повертає кількість купонів, які можна оплатити між датою розрахунку та датою погашення
CUMIPMT (ставка; кпер; пв; поч_період; кінц_період; тип) (ОБШДОХОД (ставка; кол_пер; нз; нач_період; кон_період; тип))	повертає сукупну суму відсотків, виплачувану між двома періодами
DISC (дата_угоди; дата_погаш; ціна; погашення; базис) (СКИДКА (дата_согл; дата_вступл_в_силу; цена; погашение; базис))	повертає дисконтну ставку для цінних паперів
DOLLARDE (дріб_грн; дріб)	перетворює ціну в гривнях, виражену як

(РУБЛЬ,ДЕС (дроб_руб; дроб))	дріб, на ціну в гривнях, виражену як десяткове число
DOLLARFR (дес_грн;дріб) (РУБЛЬ,ДРОБЬ (дроб_руб;дроб))	перетворює ціну в гривнях, виражену як десяткове число, на ціну в гривнях, виражену як дріб
EFFECT (номін_ставка;кількість_пер) (ЭФФЕКТ (номинальная_ставка; кол_пер))	повертає річну ефективну відсоткову ставку
FV (ставка;кпер;спл;зв;тип) (ПС (ставка;кпер;плт;бс;тип))	повертає майбутню вартість інвестиції на основі постійних періодичних виплат і постійної відсоткової ставки
INTRATE (дата_угоди;дата_погаш; інвестиція; погашення; базис) (ИНОРМА (дата_согл; дата_вступл_в_силу; инвестиция;...))	повертає відсоткову ставку для повністю інвестованих цінних паперів
IRR (значення; припущення) (ВСД (значения; предположение))	повертає внутрішню ставку прибутковості для ряду періодичних грошових переміщень
ISPMT (ставка;період;кпер;зв) (ПРОЦПЛАТ (ставка;період;кпер;пс))	повертає відсотки, сплачувані за певний період інвестиції
MIRR (значення; ставка_фінанс; ставка_реінвест) (МВСД (значения; ставка_финанс; ставка_реинвест))	повертає внутрішню ставку прибутковості для ряду періодичних грошових переміщень з урахуванням як витрат на інвестування, так і відсотка реінвестування надходжень
NOMINAL (ефект_ставка;кількість_пер) (НОМИНАЛ (факт_ставка; кол_пер))	повертає річну номінальну відсоткову ставку
NPV (ставка;значення1;значення2;...) (ЧПС (ставка; значение1; значение2;...))	повертає чисту зведену вартість інвестиції на основі дисконтної ставки та вартості майбутніх виплат (від'ємні значення) і надходжень (додатні значення)
PRICE (дата_угоди; дата_погаш; ставка; прибуток; погашення;...) (ЦЕНА (дата_согл; дата_вступл_в_силу; ставка; доход;...))	повертає ціну за 100 грн. номінальної вартості цінних паперів із періодичною виплатою відсотків
ODDFPRICE (дата_угоди;дата_погаш; дата_випуску;..) (ЦЕНАПЕРВНЕРЕГ (дата_согл; дата_вступл_в_силу;...))	повертає ціну за 100 грн. номінальної вартості цінних паперів із нерегулярним періодом
PMT (ставка;кпер;зв; мв; тип) (ПРПЛТ (ставка; період; кпер; пс; бс; тип))	повертає суму чергової виплати за позичкою на основі постійних періодичних виплат і постійної відсоткової ставки

PV (ставка;кпер;спл;мв;тип) (ПС (ставка;кпер;плт;бс;тип))	повертає теперішню (зведену) вартість інвестиції – загальну суму, яка на цей час дорівнює сукупності майбутніх виплат
RATE (кпер;спл;зв;мв;тип;припущення) (СТАВКА (кпер;плт;бс;тип; предположение))	повертає відсоткову ставку за позичкою або інвестицією за один період
RECEIVED (дата_угоди;дата_погаш; інвестиція;знижка;базис) (ПОЛУЧЕНО (дата_согл; дата_вступл_в_силу; інвестиція; скидка;...))	повертає суму, отриману на момент погашення повністю інвестованих цінних паперів
SLN (поч_вартість;зал_вартість; термін_експлуатації) (АПЛ (нач_стоимость; ост_стоимость; время_эксплуатации))	повертає величину амортизації активу за один період із використанням лінійного методу
SYD (поч_вартість;зал_вартість; термін_експлуатації;період) (АСЧ (нач_стоимость; ост_стоимость; время_эксплуатации;период))	повертає величину амортизації активу за вказаний період із використанням методу підсумовування річних чисел
TBILLEQ (дата_угоди;дата_погаш; знижка) (РАВНОКЧЕК (дата_согл; дата_вступл_в_силу; скидка))	повертає еквівалентній облігації прибуток за казначейським векселем
VDB (поч_вартість;зал_вартість; термін_експлуатації;...) (ДДОБ (нач_стоимость; ост_стоимость; время_эксплуатации;...))	повертає величину амортизації активу за будь-який указаний період, у тому числі за часткові періоди, з використанням методу подвійного зменшення залишку або іншого вказаного методу
XIRR (значення;дати;припущення) (ЧИСТВНДОХ (значения; даты; предп))	повертає внутрішню норму прибутковості для графіку грошових потоків
XNPV (ставка;значення;дати) (ЧИСТНЗ (ставка; значения; даты))	повертає чисту зведену вартість для графіку грошових потоків
YIELD (дата_угоди;дата_погаш;ставка; ціна;погашення;...) (ДОХОД (дата_согл; дата_вступл_в_силу; ставка; цена;...))	повертає прибуток за цінними паперами з періодичною виплатою відсотків

Приклад 6.3.6. Потрібно розрахувати значення показника IRR (внутрішньої норми рентабельності) для проекту, розрахованого на 3 року й потребуючого інвестицій у розмірі 2000 тис. грн., який має передбачувані грошові вступи в розмірі 1000 тис. грн., 1500 тис. грн. і 2000 тис. грн.

Виконання завдання. Уведемо початкові значення у вигляді таблиці Excel. Первісну інвестицію в розмірі 2000 тис. грн. слід записати зі знаком мінус. Для обчислення внутрішньої норми рентабельності проекту скористаємося функцією IRR, вибір якої приведе до появи діалогового вікна *Аргументи функції IRR*, представлено на рисунку 6.3.32.

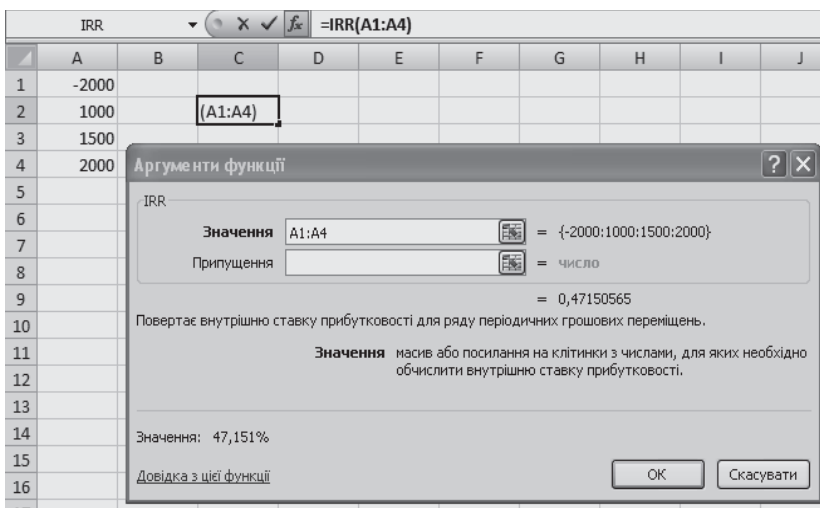


Рис.6.3.32. Діалогове вікно Аргументи функції IRR

У діалоговому вікні Аргументи функції IRR у текстовому полі Значення необхідно ввести діапазон клітинок, що містять вихідні дані задачі. Для нашого прикладу величина $IRR \approx 47,151\%$.

Присвоювання імені клітинки або діапазону клітинок. Найбільш простим способом присвоєння активній клітинці або активному діапазону клітинок імені полягає в тому, щоб просто ввести це ім'я в поле *Ім'я*, розташоване ліворуч від рядка формул і натиснути клавішу *ENTER*. Наприклад, привласнимо діапазону клітинок A1:E3 ім'я Матриця витрат. Для цього виділимо зазначений діапазон клітинок і введемо в поле *Ім'я* Матриця

витрат, як це показано на рисунку 6.3.33. Привласнювати клітинкам і діапазонам клітинок краще осмислені імена. Використання імен виправдовує себе при кількаразовому звертанні до діапазону клітинок, тому що в цьому випадку немає необхідності пам'ятати межі діапазону, досить пам'ятати ім'я й використовувати його при обчисленнях.

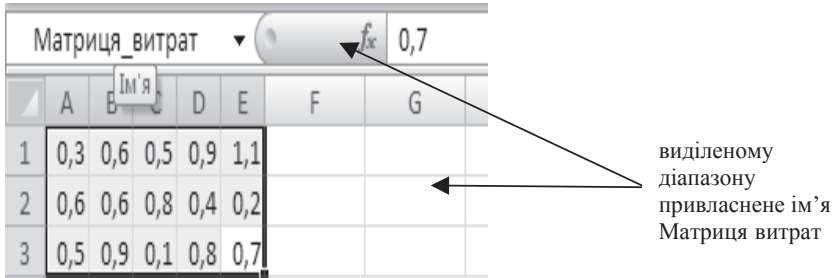


Рис.6.3.33. Приклад привласнення імені діапазону клітинок

Привласнити ім'я можна інакше. Для цього слід виділити діапазон і клацнути по ньому лівою кнопкою миші. В результаті на екрані з'явиться контекстне меню, в якому слід вибрати пункт *Назвати діапазон...*. Після чого на екрані з'явиться діалогове вікно **Нове ім'я** (рисунок 6.3.34), у полі *Ім'я* якого слід зазначити ім'я діапазону. У цьому діалоговому вікні можна також вказати *Область*, *Примітку* і побачити *Посилання*. Після виконання всіх необхідних налаштувань слід натиснути кнопку *ОК*.

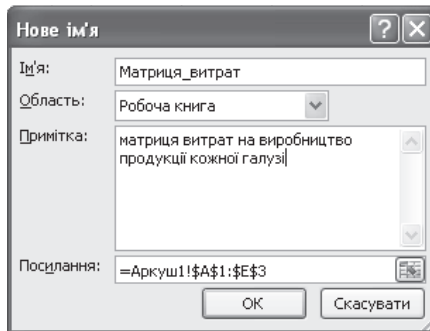


Рис.6.3.34. Діалогове вікно Нове ім'я

Перерахуємо типи основних помилок, які виникають в MS Office Excel 2007 при введенні формул.

1. ##### - ширина клітинки не дозволяє відображувати число в заданому форматі (в цьому випадку слід просто збільшити ширину клітинки);
2. #VALUE! – у формулі використаний неприпустимий тип аргументу;
3. #NAME? – не розпізнано ім'я, використане у формулі (наприклад, у формулі записана адреса з використанням кирилиці);
4. #DIV/0! – у формулі робиться спроба розділити на нуль;
5. #NUMBER! – порушені прийняті в математиці правила завдання операторів;
6. #Н/Д – як аргумент задано посилання на порожню клітинку;
7. #ССЫЛКА! – у формулі задане посилання на неіснуючу клітинку;
8. #ЕМPTY! – невірно вказаний перетин двох областей, які не мають спільних клітинок.

Питання і завдання для самостійної роботи

1. Перерахуйте основні функції категорії Математичні функції.
2. Перерахуйте основні функції категорії Дата і час.
3. Перерахуйте основні функції категорії Статистичні функції.

4. Вирішити систему рівнянь:

$$4.1. \begin{cases} 3x + 2y + z = 5, \\ 2x + 3y + z = 1, \\ 2x + y + 34z = 11; \end{cases}$$

$$4.2. \begin{cases} x + 2y + 3z - 2t = 6, \\ 2x + 4y - 2z - 3t = 18, \\ 3x + 2y - z + 2t = 4, \\ 2x - 3y + 2z + t = -8; \end{cases}$$

$$4.3. \begin{cases} 2x - 4y + 3z = 1, \\ 3x - y + 5z = 2, \\ x - 2y + 4t = 3, \\ 6x - 2y + 10z = 4; \end{cases} \quad 4.4. \begin{cases} 2x - y - z = 1, \\ x + 2y - 3z = 5, \\ 4x - 2y - 2z = 2, \\ x + 3y + 4z = 6. \end{cases}$$

5. Побудувати діаграму біноміальної інтегральної функції щільності розподілу при $n=10$ і $p=0,5$.

6. Побудувати діаграму нормальної функції щільності вірогідності при параметрах $m=24,3$ і $\sigma=1,5$.

7. Побудуйте емпіричні функції розподілу (відносні і накопичені частоти) для зростання (у див.) групи з 20 чоловіків: 181, 169, 178, 171, 179, 172, 181, 179, 168, 174, 167, 169, 171, 179, 181, 181, 183, 172, 176.

8. Дана відомість співробітників деякого підприємства. Обчислити число співробітників, які стануть пенсіонерами в наступному році.








9. Дана відомість студентів із вказівкою розміру стипендій у поточному місяці. Обчислити кількість студентів-відмінників.

10. Скласти відомість співробітників деякого підприємства із вказівкою окладів і розмірів відрахувань.

11. Дана відомість із вказівкою віку співробітників. Обчислити кількість співробітників у віці від 25 до 35 років.

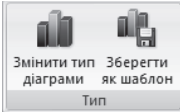
6.3.5. Побудова графіків і діаграм в MS Office Excel 2007

Часто при складанні, наприклад, презентації доповіді зручно деякі дані представити в графічному виді, побудувавши графік або діаграму. В Microsoft Excel реалізована можливість побудови діаграм і графіків різних типів. Наприклад, якщо числові дані зображуються у вигляді стовпчиків (рядів), то для їхнього відображення була обрана стовпчаста діаграма або гістограма, якщо у вигляді секторів, то секторна діаграма. Для побудови діаграми спочатку виділяють клітинки, що містять числові дані, а також клітинки з назвами категорій або рядів, які будуть використані в діаграмі. Потім слід перейти на вкладку *Вставлення* в розділ *Діаграми*. Стовпчасту

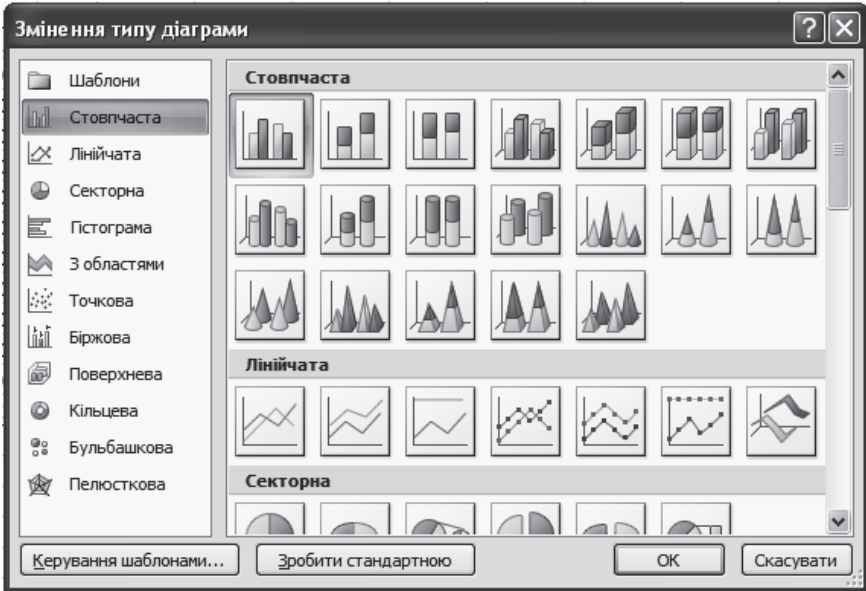
діаграму  або гістограму  обирають для порівняння декількох значень, секторну  або кільцеву – для відображення внеску кожного окремого значення в загальний підсумок, лінійчату  – для відображення змінення значень в часі, з областями  – для відображення різниці між різними рядами даних за визначений проміжок часу, точечну  – для порівняння пар значень тощо. Для побудови інших типів діаграм (біржева і т.п.) слід розкрити список .

Після введення даних користувач здійснює вибір типа діаграми. Після натиснення на кнопку відповідну вибраному типові діаграми на екрані з'являється сама діаграма і активна вкладка *Конструктор*, розділи якої дозволяють уточнити дані, за якими має бути побудована діаграма.

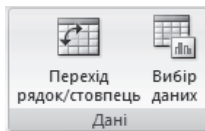
Опишемо коротко основні розділи *Конструктора*, їх склад і призначення. Активна вкладка *Конструктор (Знаряддя для діаграм)* містить розділи *Тип*, *Дані*, *Макети діаграм*, *Стилі діаграм*, *Розташування*.

Розділ *Тип*  дозволяє змінювати тип діаграми або зберегати побудовану діаграму в якості шаблону з ціллю застосування цього шаблону надалі при побудові діаграм.

При натисненні на кнопку *Змінити тип діаграми* з'являється діалогове вікно *Змінення типу діаграми*, в якому можна вибрати необхідний тип стандартної діаграми або один з раніше створених шаблонів і натиснути на кнопку *OK*.



При виборі кнопки *Зберегти як шаблон* на екрані з'являється діалогове вікно *Збереження шаблону діаграми*, в якому користувачеві пропонується ввести ім'я файлу шаблону діаграми.



Розділ *Дані* призначений для змінення діапазону даних, за якими побудована діаграма або поміняти дані, розташовані на осях *X* и *Y* місцями.

При натисненні на кнопку *Перехід рядок/стовпець* діаграма змінить вигляд, оскільки дані, відзначені на осі *X*, будуть переміщені на вісь *Y* і навпаки.

При натисненні на кнопку *Вибір даних* на екрані з'являється діалогове вікно *Вибір джерела даних* (рисунок 6.3.35).

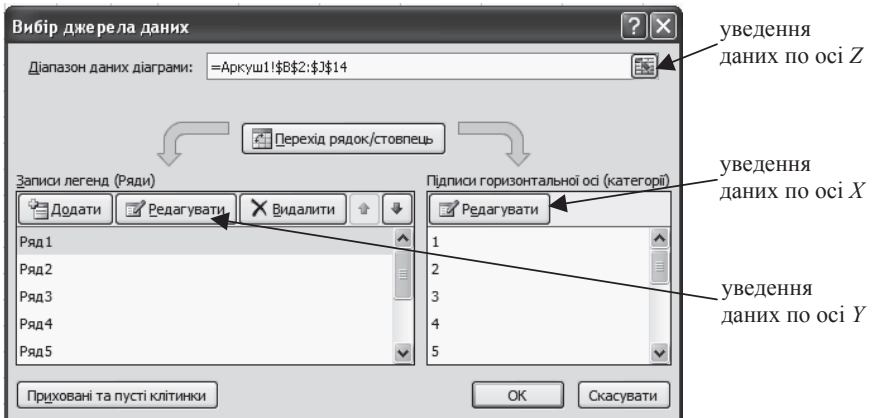
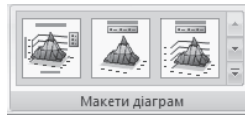


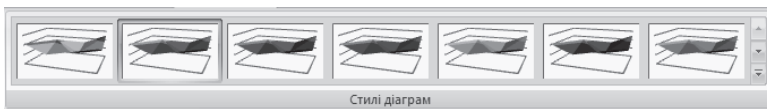
Рис. 6.3.35. Уведення даних по осях координат для поверхневих діаграм

Розділ *Макети діаграм* дозволяє змінити макет діаграми.



дозволяє змінити макет

Розділ *Стилі діаграм*



дозволяє змінити стиль діаграми.



Розділ *Розташування* дозволяє вказати робочий аркуш, на якому слід розташувати діаграму.

Для управління відображенням осей координат, сітки, підписів осей, легенд і діаграми слід перемкнутися на вкладку *Макет*.

Приклад 6.3.7. Побудувати частину гіперболічного параболоїда

вигляду $\frac{x^2}{18} - \frac{y^2}{8} = z$, що розташований в діапазоні $x \in [-3; 3]$ і $y \in [-2; 2]$

з кроком 0,5 для обох змінних.

Виконання завдання. Розташуємо значення x в діапазоні від -3 до 3 з кроком 0,5 в стовпці А. Початкове значення x введемо у клітинку А2, у клітинку А3 введемо формулу вигляду =А2+0,5 і натиснемо на клавішу ENTER. Зробимо активною клітинку А3, підведемо покажчик миші до маркера автозаповнення, натиснемо ліву кнопку миші і, утримуючи її, скопіюємо формулу у вертикальному напрямі від клітинки А2 до клітинки А14.

Розташуємо значення y в діапазоні від -2 до 2 з кроком 0,5 в першому рядку. Початкове значення y введемо у клітинку В1, у клітинку С1 введемо формулу вигляду =В1+0,5 і натиснемо на клавішу ENTER. Зробимо активною клітинку С1, підведемо покажчик миші до маркера автозаповнення, натиснемо ліву кнопку миші і, утримуючи її, скопіюємо формулу в горизонтальному напрямі від клітинки D1 до клітинки J1.

Введемо у клітинку В2 формулу $\frac{x^2}{18} - \frac{y^2}{8} = z$ для обчислення z


вигляду =А2^2/18-В1^2/8. Перед виконанням копіювання формули зафіксуємо за допомогою знаку \$ стовпець А (\$А2) і перший рядок (В\$1). Таким чином, у клітинку В2 буде записана формула = \$А2^2/18-В\$1^2/8. Зробимо активною клітинку В2, підведемо покажчик миші до маркера автозаповнення і виконаємо копіювання формули в горизонтальному напрямі від клітинки С2 до клітинки J2. Після виконання автокопіювання активним є діапазон клітинок від С2 до J2. Підведемо покажчик миші до маркера автозаповнення і виконаємо копіювання у вертикальному напрямі, заповнивши всі клітинки від В2 до J14.

B2		fx = \$A2^2/18-B\$1^2/8									
	A	B	C	D	E	F	G	H	I	J	
1	x\y	-2	-1,5	-1	-0,5	0	0,5	1	1,5	2	
2		0	0,219	0,38	0,469	0,5	0,47	0,4	0,219	0	
3		-2,5	-0,15	0,066	0,22	0,316	0,35	0,32	0,2	0,066	-0,15
4		-2	-0,28	-0,06	0,1	0,191	0,22	0,19	0,1	-0,06	-0,28
5		-1,5	-0,38	-0,16	0	0,094	0,13	0,09	0	-0,16	-0,38
6		-1	-0,44	-0,23	-0,1	0,024	0,06	0,02	-0,1	-0,23	-0,44
7		-0,5	-0,49	-0,27	-0,1	-0,02	0,01	-0	-0,1	-0,27	-0,49
8		0	-0,5	-0,28	-0,1	-0,03	0	-0	-0,1	-0,28	-0,5
9		0,5	-0,49	-0,27	-0,1	-0,02	0,01	-0	-0,1	-0,27	-0,49
10		1	-0,44	-0,23	-0,1	0,024	0,06	0,02	-0,1	-0,23	-0,44
11		1,5	-0,38	-0,16	0	0,094	0,13	0,09	0	-0,16	-0,38
12		2	-0,28	-0,06	0,1	0,191	0,22	0,19	0,1	-0,06	-0,28
13		2,5	-0,15	0,066	0,22	0,316	0,35	0,32	0,2	0,066	-0,15
14		3	0	0,219	0,38	0,469	0,5	0,47	0,4	0,219	0



Рис. 6.3.36. Приклад заповнення масиву даних і вибір типу діаграми – об’ємна поверхнева діаграма

Побудуємо поверхневу діаграму, виділивши діапазон клітинок від B2 до J14. Для цього перейдемо на вкладку *Вставлення*, натиснемо на кнопку *Інші діаграми* в розділі *Діаграми* і в списку, що розкрився, виберемо тип діаграми – об’ємна поверхнева діаграма.

На екрані з’явиться об’ємна поверхнева діаграма, яку необхідно правильно «настроїти» (увести правильні дані по осях координат, при необхідності вказати розміщення діаграми тощо). Натиснемо на кнопку *Вибір даних* і в діалоговому вікні, що з’явилося, натиснемо на кнопку *Редагувати* з розділу *Підписи горизонтальної осі (категорії)*. У результаті на екрані з’явиться ще одне діалогове вікно *Підписи осі*, після чого потрібно увести підписи горизонтальної осі, або вручну в полі, що редагується, вказати діапазон A2:A14, або натиснути ліву кнопку миші й, утримуючи її, виділити потрібний діапазон клітинок. Якщо діалогове вікно закриває необхідний діапазон даних, можна або натиснути на кнопку  для згортання вікна, або

натиснути ліву кнопку миші на заголовку вікна й, утримуючи її, відбуksирувати діалогове вікно убік. Після введення даних слід натиснути на кнопку *OK*, що знову приведе до появи на екрані вікна **Вибір джерела даних** (рисунок 6.3.35).

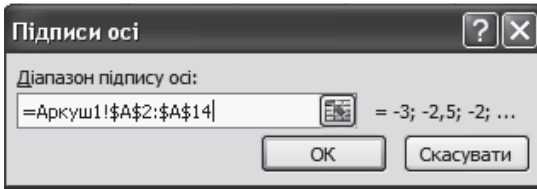


Рис. 6.3.37. Введення підпису горизонтальної осі (осі X)

Розділ *Записи легенд (Ряди)* дозволяє ввести підписи осі Y. Для цього потрібно клацнути на слові Ряд 1 і натиснути на кнопку *Редагувати*. На екрані з'явиться діалогове вікно **Редагування ряду**, в текстове поле *Ім'я ряду* введемо підпис, клацнувши лівою кнопкою миші по клітинці B1, що містить перше значення змінної у з допустимого діапазону, і натиснути на кнопку *OK*, що приведе до повернення в діалогове вікно **Вибір джерела даних**. Описану процедуру слід повторити з кожним рядом, поки не будуть введені всі допустимі значення змінної у. Після виконання всіх дій слід натиснути на кнопку *OK* (рисунок 6.3.38).

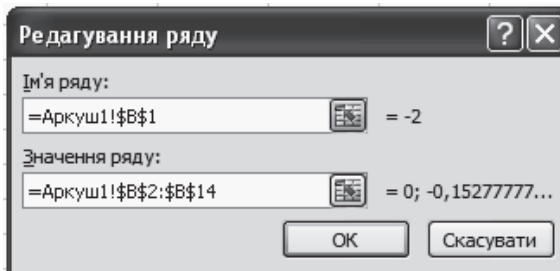
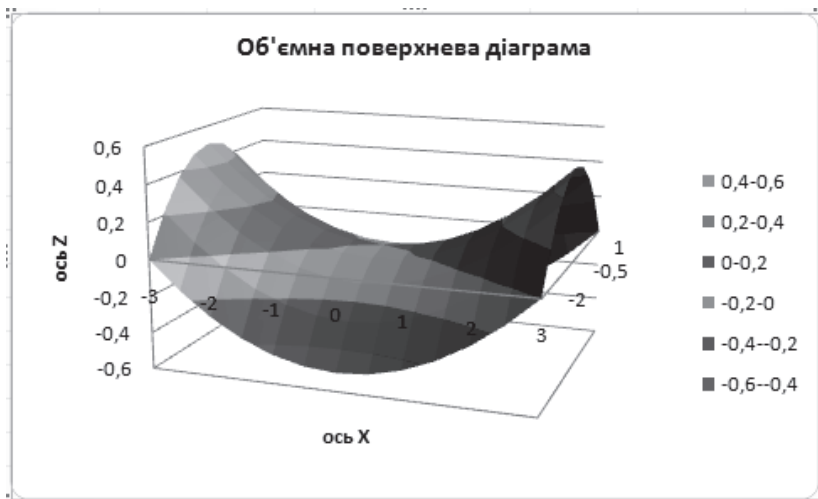


Рис. 6.3.38. Введення підпису осі Y для поверхневих діаграм

Для завдання назви діаграми і підписів до осей слід перемкнутися на вкладку *Макет*. У розділі Підписи натиснемо на кнопку *Назва діаграми*.

Отримаємо діаграму наступного вигляду:



При натисненні правої кнопки миші по побудованій діаграмі з'являється допоміжне меню, за допомогою якого можна виконати всі перераховані команди *Змінити тип діаграми...*, *Вибір даних...*, *Перемістити діаграму...*, *Обертання об'ємної фігури...*, *Формат області діаграми...*.

Питання і завдання для самостійної роботи

1. Побудувати верхню частину параболи $y^2 = x$ у діапазоні $x \in [0; 4]$ з кроком $\Delta = 0,25$.
2. Побудувати гіперболу $y = \frac{1}{2x}$ у діапазоні $x \in [0,1; 5,1]$ з кроком $\Delta = 0,25$.
3. Побудувати верхню половину еліпсу $\frac{x^2}{9} + \frac{y^2}{4} = 1$ в діапазоні $x \in [-3,5; 3,5]$ з кроком $\Delta = 0,5$.
4. Побудувати верхнє півколо $x^2 + y^2 = 4$ у діапазоні $x \in [-2; 2]$ з кроком $\Delta = 0,25$.

5. Графічно розв'язати системи:

$$5.1. \begin{cases} x^2 + y^2 = 4, \\ y = 2 \sin x \end{cases} \text{ в діапазоні } x \in [0; 3] \text{ з кроком } \Delta = 0,2;$$

$$5.2. \begin{cases} 9x^2 + 4y^2 = 36, \\ x^2/9 + y^2 = 1 \end{cases} \text{ в діапазоні } x \in [0; 3] \text{ з кроком } \Delta = 0,2.$$

6. Побудувати верхню частину еліпсоїда:

$$\frac{x^2}{4} + \frac{y^2}{9} + \frac{z^2}{4} = 1$$

Діапазони змінення змінних x і y : $x \in [-2; 2]$ з кроком $\Delta = 0,5$ і $y \in [-3; 3]$ з кроком $\Delta = 1$.

7. Побудувати верхню частину односмугового гіперboloїда:

$$\frac{x^2}{4} + \frac{y^2}{9} - \frac{z^2}{4} = 1$$

Діапазони змінення змінних x і y : $x \in [-3; 3]$ з кроком $\Delta = 0,5$ і $y \in [-4; 4]$ з кроком $\Delta = 1$.

8. Побудувати верхню частину конуса:

$$\frac{x^2}{4} + \frac{y^2}{9} - \frac{z^2}{4} = 0$$

Діапазони змінення змінних x і y : $x \in [-2; 2]$ з кроком $\Delta = 0,5$ і $y \in [-3; 3]$ з кроком $\Delta = 1$.

9. Вибрати тип і побудувати діаграму змінення курсу євро в поточному місяці.

10. Вибрати тип і побудувати діаграму успішності студентів деякої групи.

6.3.6. Використання пакетів-надбудов в MS Office Excel 2007

Надбудова – спеціальна програма, доступна при установці MS Office Excel 2007. Прикладами пакетів-надбудов є надбудова для пошуку рішення, пакет аналізу даних і так далі. Перш ніж використовувати можливості пакету-надбудови, його необхідно завантажити. Для цього слід клацнути по кнопці *Microsoft Office*, розташованій в лівому верхньому кутку і у вікні, що

розкрилося, клацнути лівою кнопкою миші по кнопці *Параметри Excel*. На екрані з'явиться однойменне діалогове вікно, в лівій частині якого слід вибрати команду *Надбудови*, при цьому в правій частині відображатимуться всі доступні надбудови, наприклад, *Майстер підстановок*, *Майстер підсумовування*, *Пакет аналізу*, *Пакет аналізу – VBA*, *Пошук розв'язання* і т.д. У нижній частині цього вікна слід натиснути на кнопку *Перейти*. У вікні *Доступні налаштування*, представленому на рисунку 6.3.39, слід встановити прапорець напроти пункту *Пошук розв'язання* (це інструмент для оптимізації та розв'язання рівнянь) і натиснути на кнопку *ОК*.

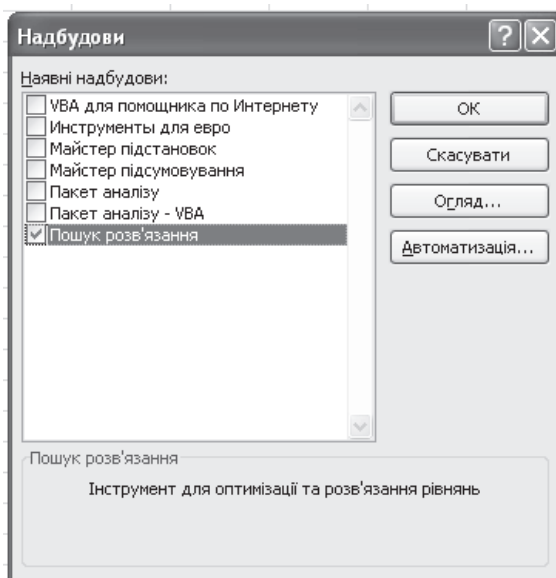


Рис. 6.3.39. Діалогове вікно Надбудови

Аналогічно, установивши прапорець для потрібної надбудови й натиснувши на кнопку *ОК*, можна завантажити будь-яку іншу з доступних надбудов. Якщо після натискання на кнопку *ОК* на екрані з'являється повідомлення, представлене на рисунку 6.3.40, то слід вставити установочний диск пакету Microsoft Office 2007 і натиснути на кнопку *Так*.

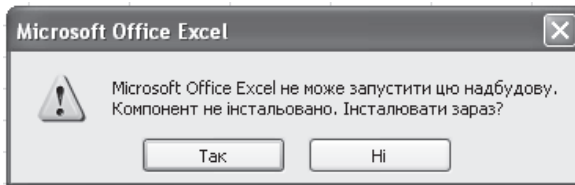
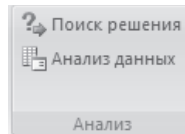


Рис.6.3.40. Запит на інсталяцію пакету-надбудови

Після інсталяції зазначених пакетів-надбудов на вкладці *Дані* з'являється група інструментів *Аналіз* із доступними пакетами-надбудовами



Розглянемо приклад розв'язання оптимізаційної задачі за допомогою можливостей пакету-надбудови *Пошук рішення*.

Приклад 6.3.7. Розв'язати задачу $F = 2x_1 + 2x_2 \rightarrow \max$

$$\begin{cases} x_1 + 3x_2 \leq 18, \\ 2x_1 + x_2 \leq 16, \\ x_2 \leq 5, \\ 3x_1 \leq 21. \end{cases}$$

$$x_1 \geq 0, x_2 \geq 0.$$

Виконання завдання. Введемо початкові дані в таблицю Excel. У діапазон клітинок A2:A5 введемо значення з правої частини системи обмежень, тобто 18, 16, 5 і 21. У клітинки C1 і D1 введемо початкові значення для невідомих x_1 і x_2 , рівні нулю. Під ними в діапазоні C2:C5 розмістимо коефіцієнти при невідомій x_1 , а в діапазоні D2:D5 – коефіцієнти при невідомій x_2 з нерівностей системи обмежень. Якщо невідома в нерівності системи

обмежень відсутня, то це означає, що відповідний нею коефіцієнт дорівнює нулю. У діапазоні B2:B5 вкажемо формули, розташовані в лівій частині нерівностей системи обмежень. Так, у клітинці B2 буде записана формула $=C2*SC\$1+D2*SD\1 , решту формул можна ввести, використовуючи автозаповнення. У клітинці B3 з'явиться формула $=C3*SC\$1+D3*SD\1 ; у клітинці B4 – формула $=C4*SC\$1+D4*SD\1 ; у клітинці B5 – формула $=C5*SC\$1+D5*SD\1 . У клітинку E1 занесемо формулу для розрахунку значення цільової функції F , тобто $=2*C1+3*D1$. Результат введення початкових даних наведено на рисунку 6.3.41.

E1		fx =2*C1+3*D1			
	A	B	C	D	E
1			0	0	0
2	18	0	1	3	
3	16	0	2	1	
4	5	0	0	1	
5	21	0	3	0	

Рис.6.3.41. Приклад введення початкових даних

Перейдемо на вкладку *Дані* і в групі інструментів *Аналіз* виберемо *Пошук рішення*. В результаті на екрані з'явиться однойменне діалогове вікно, представлено на рисунку 6.3.42.

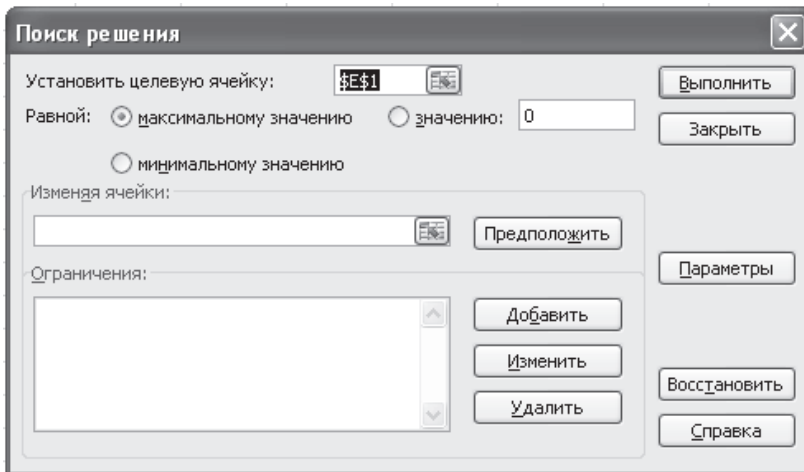


Рис.6.3.42. Діалогове вікно Пошук рішення

У робочому полі *Встановити цільову клітинку* за допомогою клацання лівою кнопкою миші по клітинці E1 укажемо клітинку, що містить формулу для обчислення значення цільової функції, як показано на рисунку 6.3.43. Установимо перемикач Рівної в положення максимального значення як показано на рисунку 6.3.43, оскільки за умовою задачі необхідно знайти

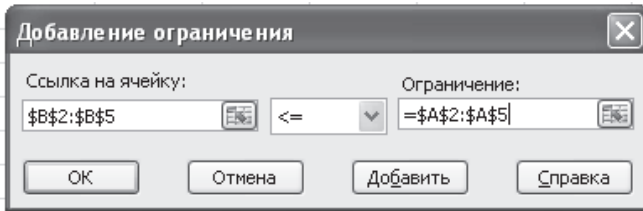


Рис. 6.3.43. Діалогове вікно Додавання обмеження

максимум цільової функції F . У робочому полі *Змінюючи клітинки* необхідно вказати діапазон, що містить початкові значення невідомих, тобто діапазон $\$C\$1:\$D\1 . Для заповнення поля *Обмеження* натиснемо на кнопку *Додати*. У результаті на екрані з'явиться діалогове вікно *Додавання обмеження*, представлене на рисунку 6.3.44.

У робочому полі діалогового вікна *Додавання обмеження* в робочому полі *Посилання на клітинку* необхідно за допомогою миші вказати діапазон, відповідний до лівих частин нерівностей системи обмежень, тобто діапазон $\$B\$2:\$B\5 , як показано на рисунку 6.3.44. У якості умови в списку, що розкривається, слід задати \leq , тобто знак нерівності в системі обмежень. У робочому полі *Обмеження* за допомогою миші вказати діапазон, відповідний до правих частин нерівностей системи обмежень, тобто діапазон $=\$A\$2:\$A\5 , як показано на рисунку 6.3.44. Після заповнення всіх полів слід натиснути на кнопку *ОК*, що приведе до повернення в діалогове вікно *Пошук розв'язку*. Знову натиснемо на кнопку *Додати* в розділі *Обмеження* діалогового вікна *Пошук розв'язку* для того, щоб увести умову незаперечності невідомих x_1 і x_2 . У діалоговому вікні, що з'явилося, Додавання обмеження в робочому полі *Посилання на гніздо* за допомогою миші слід вказати діапазон $\$C\$1:\$D\1 . У якості умови в списку, що розкривається, вибрати знак нерівності \geq . У робочому полі *Обмеження* завдамо значення нуль і натиснемо на кнопку *ОК*. Після повернення в діалогове вікно *Пошук розв'язку* слід натиснути на кнопку *Виконати*. На

екрані з'явиться діалогове вікно **Результати пошуку розв'язку**, представлена на рисунку 6.3.45. У клітинці E1 буде отримано максимальне значення цільової функції $F = 24$, яке досягається при $x_1 = 6$ (результат представлено в клітинці C1) і при $x_2 = 4$ (результат представлено в клітинці D1). В діалоговому вікні **Результати пошуку розв'язку** слід вибрати пункт **Зберегти знайдений розв'язок** и натиснути на кнопку **ОК**.

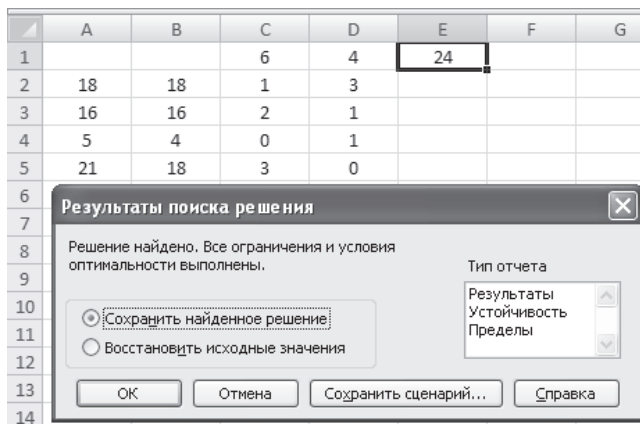


Рис.6.3.45. Отримання результату за допомогою пакету-надбудови Пошук розв'язку

Приклад 6.3.8. За даними проведеного опиту восьми груп сімей відомі дані зв'язки витрат населення на продукти харчування з рівнем доходів сім'ї

Витрати на продукти харчування y тис. грн.	0,9	1,2	1,8	2,2	2,6	2,9	3,3	3,8
Доходи сім'ї x тис. грн.	1,2	3,1	5,3	7,4	9,6	11,8	14,5	18,7

Побудувати лінійну регресійну модель залежності витрат на продукти харчування від доходів сім'ї.

Виконання завдання. Лінійне рівняння парної регресії має вигляд: $\hat{y} = \hat{a}_0 + \hat{a}_1 x$. Введемо початкові дані в таблицю Excel. Значення змінної x введемо у клітинки з діапазону A1:A8, а значення y – у клітинки з діапазону B1:B8. Перейдемо на вкладку дані і в групі інструментів *Аналіз* виберемо

Аналіз даних. В результаті на екрані з'явиться діалогове вікно *Аналіз даних*, серед *Інструментів аналізу* якого слід вибрати рядок *Регресія* і натиснути на кнопку *OK* (рисунок 6.3.46).

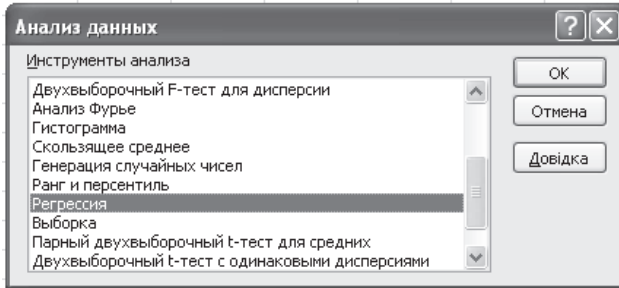


Рис.6.3.46. Діалогове вікно Аналіз даних

Після вибору Інструменту *Регресія* на екрані з'явиться однойменне діалогове вікно, представлене на рисунку 6.3.47. У робочому полі *Вхідний інтервал Y* за допомогою покажчика миші необхідно ввести діапазон значень змінної y , тобто $B\$1:B\8 , як показано на рисунку 6.3.47. У робочому полі *Вхідний інтервал X* за допомогою покажчика миші необхідно ввести діапазон значень змінної x , тобто $A\$1:A\8 , як показано на рисунку 6.3.47. У розділі *Параметри виводу* виберемо *Вихідний інтервал* і у відповідному робочому полі вкажемо за допомогою миші ліву верхню клітинку вихідного діапазону, наприклад $C\$1$. В разі накладення вихідного діапазону на початкові дані на екран буде виведено попередження. У розділі *Залишки* можна встановити прапорець для пункту *Графік підбору*. Такий графік дозволить візуально побачити відзнаку точок експериментальних даних від точок, передбачених на основі побудованої регресійної моделі. Після виконання всіх перерахованих дій слід натиснути на кнопку *OK*.

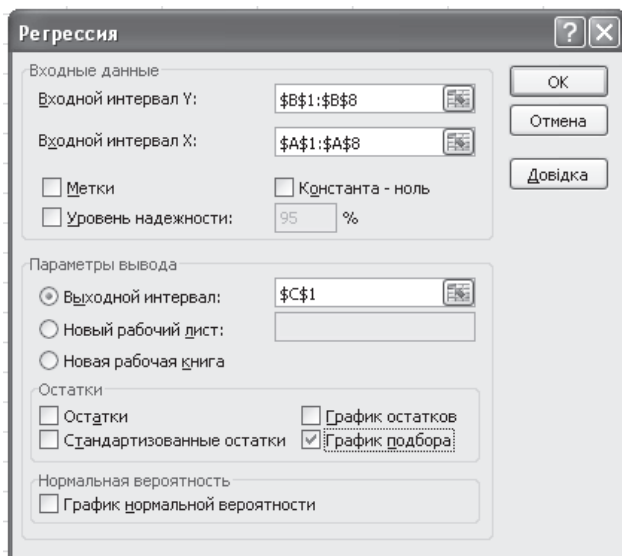


Рис. 6.3.47. Диалогове вікно Регресія

Результати аналізу наведено на рисунку 6.3.48. Результати аналізу включають результати дисперсійного аналізу, коефіцієнти регресії, стандартну погрішність обчислення Y , середньоквадратичні відхилення, число спостережень, стандартні погрішності для коефіцієнтів. Оцінки коефіцієнтів лінійної парної регресії можна побачити в стовпці коефіцієнти. Рядок Y -Перетинання цього стовпця містить значення коефіцієнта $\hat{a}_0 \approx 0,82$, а рядок змінна X_1 цього стовпця – значення коефіцієнта $\hat{a}_1 \approx 0,17$. Таким чином, рівняння парної лінійної регресії має вигляд: $\hat{y} \approx 0,82 + 0,17x$, тобто зі збільшенням доходу родини на 100 грн. витрати на харчування збільшуються на 17 грн. У результатах аналізу можна побачити, наприклад, значення коефіцієнта детермінації $R \approx 0,98$, що говорить про високу точність апроксимації, тобто парна лінійна регресія дуже добре описує досліджуване явище, що можна візуально побачити на графіку підбору.

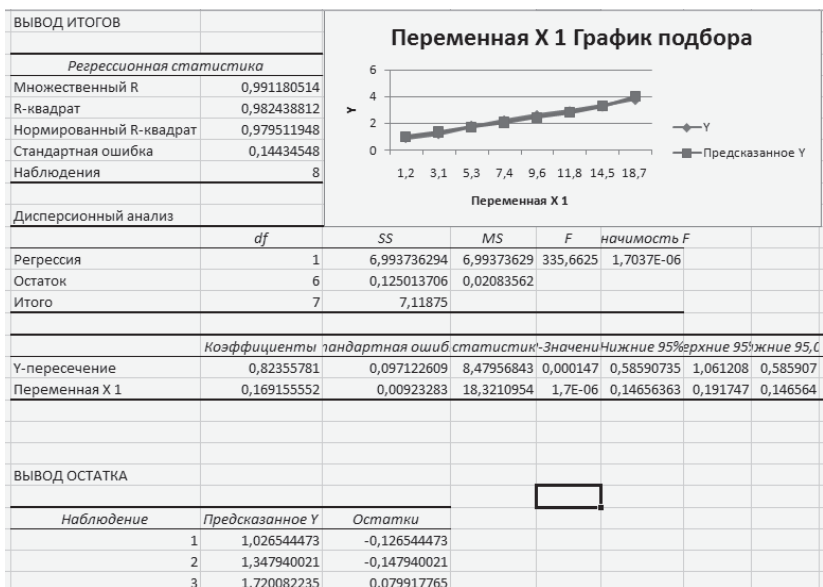


Рис.6.3.48. Результаты анализу, отримані за допомогою пакету Аналіза даних і інструмента Регресія

Питання і завдання для самостійної роботи

1. Розв'язати задачі:

$$1.1. Z = 18y_1 + 16y_2 + 5y_3 + 21y_4 \rightarrow \min$$

$$\begin{cases} y_1 + 2y_2 + 3y_4 \geq 2, \\ 3y_1 + y_2 + y_3 \geq 3, \end{cases}$$

$$y_i \geq 0, i = 1, 2, 3, 4.$$

$$1.2. F = x_1 + x_2 \rightarrow \max$$

$$\begin{cases} x_1 - x_2 \leq -2, \\ x_1 - 2x_2 \geq -13, \\ 3x_1 - x_2 \leq 6, \end{cases}$$

$$x_1 \geq 0, x_2 = 0.$$

$$1.3. F = 3x_1 + x_2 \rightarrow \max$$

$$\begin{cases} 4x_1 + 3x_2 \leq 18, \\ x_1 + 2x_2 \leq 6, \\ 0 \leq x_1 \leq 5, \\ 0 \leq x_2 \leq 4, \end{cases}$$

x_1, x_2 – цілі числа.

$$1.4. F = 2x_1 + 3x_2 \rightarrow \max$$

$$\begin{cases} 3x_1 + 5x_2 \leq 60, \\ 3x_1 + 4x_2 \leq 34, \\ x_2 \leq 8, \end{cases}$$

$$x_1 \geq 0, x_2 = 0.$$

x_1, x_2 – цілі числа.

2. По територіях регіону наводяться дані за 200X р. Побудувати лінійне рівняння парної регресії y від x .

2.1. Середній прожитковий мінімум в день одного працездатного, у.о. x (81, 77, 85, 79, 93, 100, 72, 90, 71, 89, 82, 111); середня заробітна плата, у.о. y (124, 131, 146, 130, 143, 159, 135, 152, 127, 154, 127, 162).

2.2. Середній прожитковий мінімум в день одного працездатного, у.о. x (74, 81, 90, 79, 89, 87, 77, 93, 70, 93, 87, 121); середня заробітна плата, у.о. y (122, 134, 136, 125, 120, 127, 125, 148, 122, 157, 144, 165).

2.3. Середній прожитковий мінімум в день одного працездатного, у.о. x (77, 85, 79, 93, 89, 81, 79, 97, 73, 95, 84, 108); середня заробітна плата, у.о. y (123, 152, 140, 142, 157, 181, 133, 163, 134, 155, 132, 156).

Список літератури

1. Гельман В.Я. Решение математических задач средствами Excel: Практикум. – Спб.: Питер, 2003. – 240 с.
2. Воробьев В.В. Microsoft Excel 2000: Пособие для начинающих. – К.: 2000. – 36 с.
3. Саймон, Джинджер Анализ данных в Excel: курс создания отчетов, диаграмм и сводных таблиц: Пер. с англ. – М.: Издательский дом «Вильямс», 2004. – 528 с.
4. Дубина А.Г., Орлова С.С., Шубина И.Ю., Хромов А.В. Excel для экономистов и менеджеров. – Спб.: Питер, 2004. – 295 с.
5. Златопольский Д.М. 1700 задач по Microsoft Excel. – Спб.: БХВ-Петербург, 2003. – 544 с.
6. Мур Джеффри, Уедерфорд Ларри Р. Экономическое моделирование в Microsoft Excel, 6-е изд.: Пер. с англ. – М.: Издательский дом «Вильямс», 2004. – 1024 с.
7. Васильев А.Н. Научные вычисления в Microsoft Excel. – М.: Издательский дом «Вильямс», 2004. – 512 с.

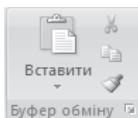
6.4. Microsoft Office Power Point 2007

Microsoft Office Power Point 2007 – це потужний засіб для створення презентацій різного ступеня складності з можливістю вбудовування графіків, діаграм, малюнків і так далі. Під *презентацією* в Microsoft Office PowerPoint прийнято розуміти послідовність слайдів, що демонструються в повноекранному режимі. Як правило, на слайдах презентації доповідач (користувач) відображує найбільш істотні моменти доповіді.

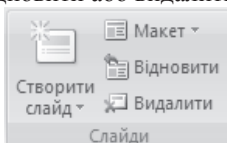
Перерахуємо вкладки стрічкового інтерфейсу Microsoft Office Power Point 2007, їх функціональне призначення і групи інструментів.

Вкладка *Основне* призначена для роботи з буфером обміну, налаштуванням шрифту, розміру, зображення тексту, що вводиться, вставки в сайт малюнка і редагування і містить 6 груп інструментів.

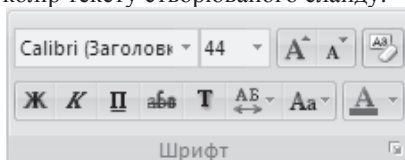
Група інструментів *Буфер обміну* надає користувачеві можливість копіювання або вирізування, вставки тексту в слайд, очищення буфера обміну.



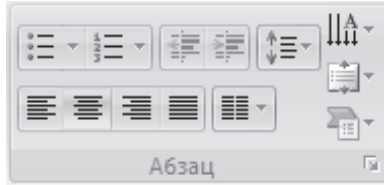
Група інструментів *Слайди* дозволяє створити новий слайд, вибрати необхідний макет слайду, відновити або видалити слайд.



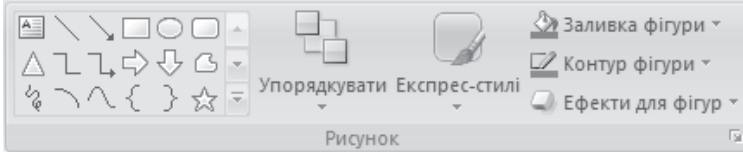
Група інструментів *Шрифт* дозволяє вибрати потрібний шрифт, його розмір, зображення, колір тексту створюваного слайду.



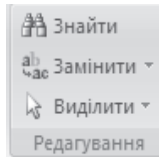
Група інструментів *Абзац* дозволяє встановити тип вирівнювання тексту, створити маркований або нумерований список, завдати напрям тексту, абзацний відступ.



Група інструментів *Рисунок* містить інструменти вставки фігур, завдання експрес-стилів, установки заливки, контура або ефекту для фігури.



Група *Редагування* призначена для пошуку і заміни тексту, а також містить інструменти виділення об'єктів.



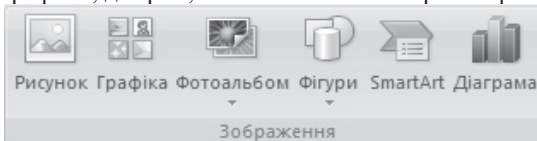
Робота з інструментами вкладки *Основне* багато в чому аналогічна роботі з цими інструментами в інших програмах пакету Microsoft Office 2007.

Вкладка *Вставлення* призначена для вставки таблиць, різних зображень, посилань, різних символів, дати і часу, видео- і звукових файлів і містить 5 груп інструментів.

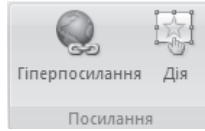
Група інструментів *Таблиці* призначена для створення таблиць різного ступеня складності і їх розміщення на слайді.



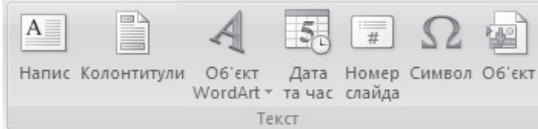
Група інструментів *Зображення* призначена для вставки рисунків з файлів, фігур, графіків, діаграм, схем SmartArt і створення фотоальбому.



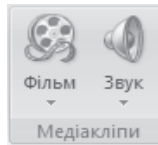
Група інструментів *Посилання* дозволяє вставляти в слайд гіперпосилання і додавати дії до виділеного об'єкту.



Група інструментів *Текст* дозволяє вставляти в слайд об'єкти WordArt, колонтитули, номер слайду, різні символи, об'єкти, дату і час.



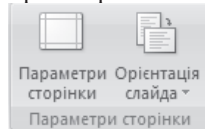
Група інструментів *Медіакліпи* дозволяє вставляти в слайд анімовану графіку і музичні файли.



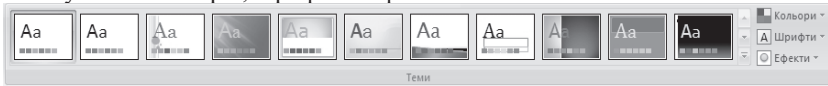
Робота з інструментами вкладки *Вставлення* аналогічна роботі з цими інструментами в програмі Word 2007.

Вкладка *Конструктор* призначена для установки параметрів сторінки, завдання теми слайду і містить 3 групи інструментів.

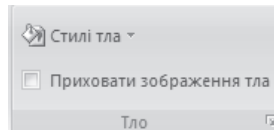
Група *Параметри сторінки* призначена для завдання орієнтації слайду і налаштування параметрів сторінки.



Група інструментів *Теми* призначена для вибору теми слайду, налаштування кольорів, шрифтів і ефектів.

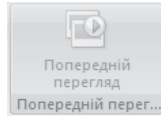


Група інструментів *Тло* призначена для завдання стилів тла і утаєння зображення тла.

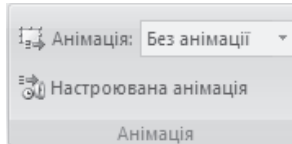


Вкладка *Анімація* призначена для налаштування режиму прогляду слайдів презентації і містить 3 групи інструментів.

Група інструментів *Попередній перегляд* призначена для попереднього перегляду слайдів презентації.



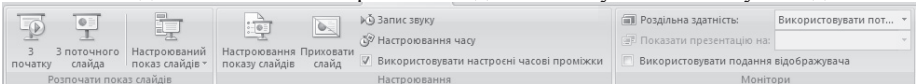
Група інструментів *Анімація* призначена для налаштування способу появи анімації.



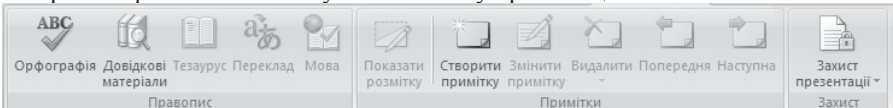
Група інструментів *Перехід до цього слайду* призначена для завдання способу переходу до наступного слайду і вибору спеціального ефекту, який буде використаний при зміні слайдів, а також для налаштування звуку і швидкості переходу від слайду до слайду.



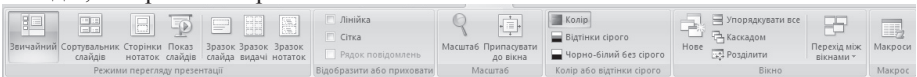
Вкладка *Показ слайдів* призначена для налаштування показу слайдів



Вкладка *Рецензування* призначена для перевірки орфографії, створення приміток і налаштування захисту презентації.



Вкладка *Вигляд* призначена для налаштування способу відображення слайдів, створення макросів.



У робочій області екрану після запуску програми Power Point 2007 можна побачити простий слайд, що містить три області:

- 1) Заголовок слайду
- 2) Підзаголовок слайду
- 3) Нотатки до слайду.

Простий слайд представлений на рисунку 6.4.1.

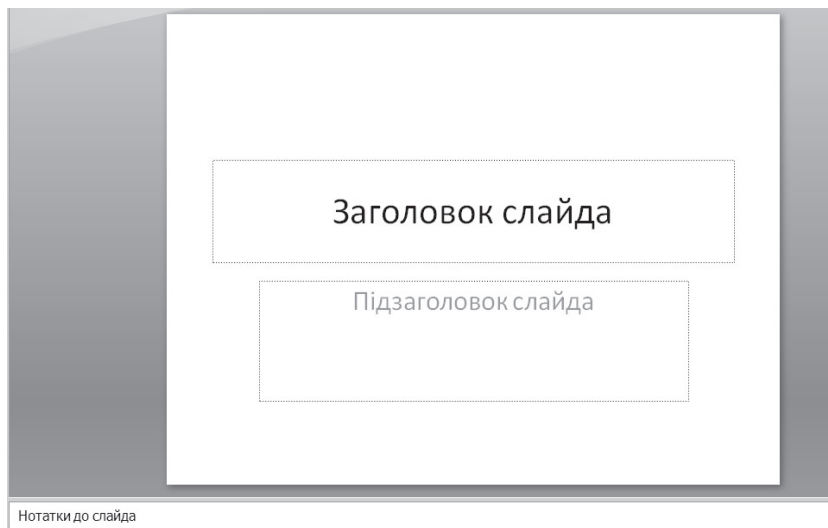


Рис.6.4.1. Структура найпростішого слайду

На рисунку 6.4.1 представлений так званий титульний слайд. Клацанням лівою кнопкою миші в полі *Заголовок слайду* користувач може ввести назву (тему) презентації, а в полі *Підзаголовок слайду* користувач може ввести анотацію тексту своєї доповіді.

6.4.1. Створення презентації в Microsoft Office Power Point 2007

Використання вбудованих шаблонів для створення слайдів в MS Office Power Point 2007. У Power Point 2007 користувачеві доступні 5 готових шаблонів-презентацій з повним оформленням і встановленими переходами від слайду до слайду презентації. Їх основне призначення – допомогти недосвідченому користувачеві в найкоротші терміни створити власну повноцінну презентацію. Перерахуємо готові шаблони презентацій: класичний фотоальбом, сучасний фотоальбом, знайомство з Power Point 2007, вікторина, рекламний буклет, широкоекранна презентація і т. д.

Для завантаження шаблону користувачеві слід натиснути на кнопку «Office» і в списку команд, що розкрився, вибрати команду *Створити*. В результаті на екрані з'явиться діалогове вікно *Створення презентації*, представлене на рисунку 6.4.2. У лівій частині цього діалогового вікна користувач може вибрати тип шаблонів – Інсталювані шаблони, при цьому в правій частині діалогового вікна відображатиметься весь список доступних шаблонів як показано на рисунку 6.4.2.

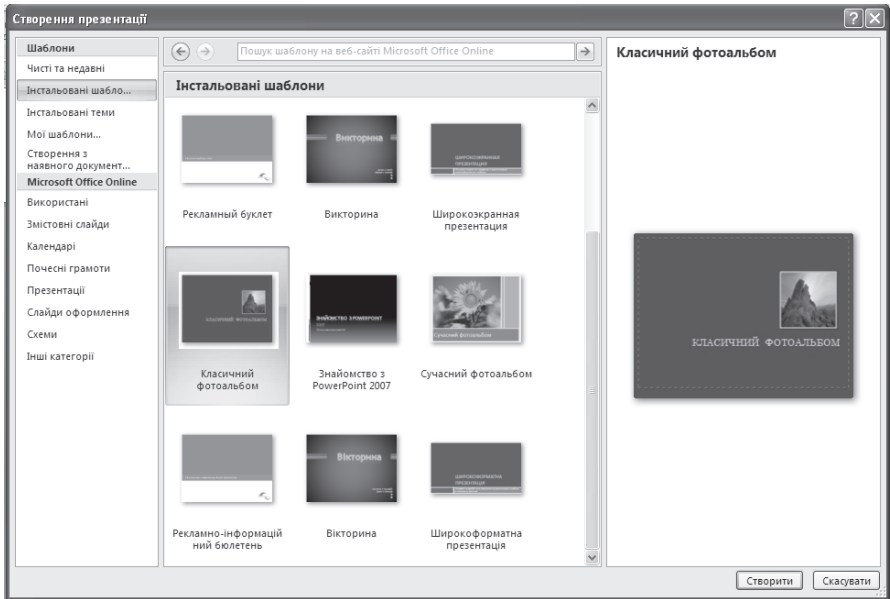


Рис.6.4.2. Створення презентації на основі шаблону Класичний фотоальбом

Після вибору шаблону слід натиснути кнопку *Створити*, що приведе до завантаження слайдів з можливістю подальшого їх редагування. Змінити напис на слайді можна, клацнувши лівою кнопкою миші на потрібному написі і ввівши свій напис (при необхідності видаливши старий напис). Змінити будь-який рисунок на слайді можна, натиснувши праву кнопку миші і вибравши в допоміжному меню, що розкрилося, пункт *Змінити рисунок...*, як показано на рисунку 6.4.3. В результаті на екрані з'явиться діалогове вікно *Вставлення рисунка*, в якому можна вибрати і завантажити інший рисунок, який буде вставлений із збереженням пропорцій. Рисунок на слайді можна повернути під іншим кутом або змінити його розміри за допомогою покажчика

миші подібно до того, як користувач працював з рисунками в MS Office Word 2007.

Зручність слайдів одного шаблону полягає в тому, що вони виконані в єдиному стилі і можуть бути з легкістю змінені з урахуванням побажань користувача.

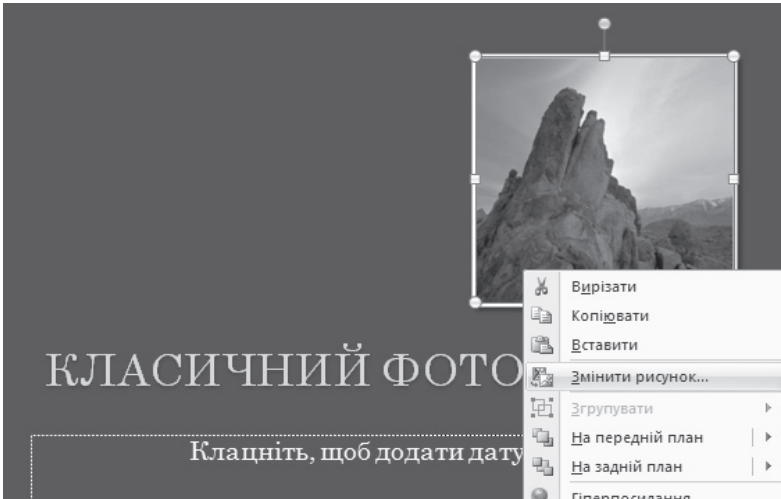


Рис. 6.4.3. Вибір команди Змінити рисунок...

У MS Office Power Point 2007 можна завантажити галерею для вибору способу розмітки слайду. Для цього користувачеві слід перемкнутися на вкладку *Основне* і в групі інструментів *Слайди* натиснути на кнопку *Макет*, що розкривається. На екрані з'являться різні варіанти розмітки слайдів – макети, представлені на рисунку 6.4.4.

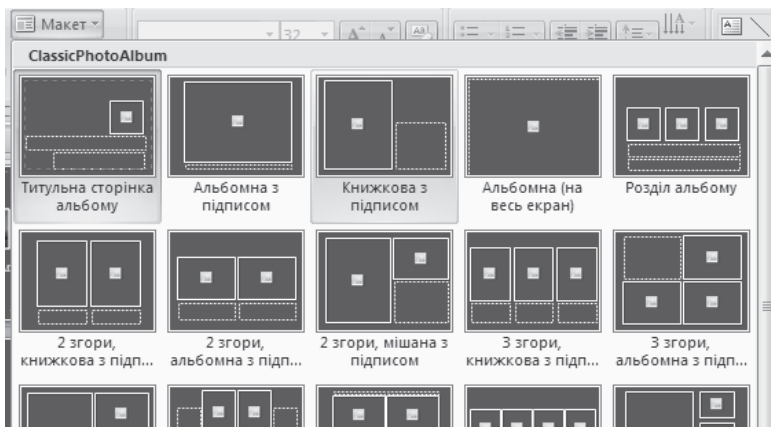


Рис.6.4.4. Галерея макетів

Додавання і видалення слайдів презентації. Для додавання слайду презентації слід перейти на вкладку *Основне* і в групі інструментів *Слайди* натиснути на кнопку *Створити слайд*, що розкривається. Після вибору макету слайду в презентацію буде вставлений слайд з вибраним типом розмітки. Серед списку команд *Створити слайд*, що розкривається, є можливість дублювання слайдів (створення точної копії) – команда *Дублювати вибрані слайди*.

Для видалення поточного слайду слід на вкладці *Основне* в групі інструментів *Слайди* натиснути на кнопку *Видалити*.

Створення нової презентації без використання шаблонів можна виконати за допомогою натиснення комбінації клавіш *Ctrl+N* або натисненням на кнопку «*Office*» і вибором команди *Створити* із списку команд цієї кнопки, що розкрився. У діалоговому вікні *Створення презентації*, що з'явилося, слід вибрати *Нова презентація*.

6.4.2. Оформлення слайдів презентації

Оформлення зовнішнього вигляду слайду за допомогою інструментів вкладки Конструктор. Швидко оформити слайд дозволяє вибір однієї з готових тем. Галерею тем можна проглянути і налаштувати, скориставшись групою інструментів *Теми*. У групі інструментів *Параметри*

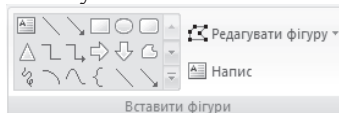
сторінки можна встановити орієнтацію слайду (книжна або альбомна) і параметри слайду.

Презентація виглядатиме ефектною, якщо разом з текстом міститиме малюнки, що пояснюють текст, формули, таблиці, відеоролики або анімацію.

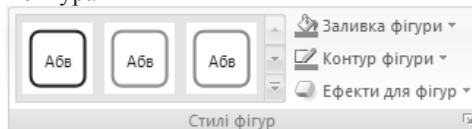
Вставка в слайд елементів за допомогою інструментів вкладки Вставлення. Для вставки напису слід натиснути на кнопку *Напис*, розташовану серед інструментів групи *Текст* і клацнути лівою кнопкою миші в тому місці слайду, де необхідно ввести текст.

Для вставки об'єкту **WordArt** слід натиснути на кнопку *Об'єкт WordArt*, що розкривається, розташовану серед інструментів групи *Текст*, і вибрати тип заливки об'єкту WordArt і його контур. На слайді з'явиться об'єкт WordArt з текстом *Текст* напису, поміщеним у пунктирний прямокутник. Для введення власного тексту слід клацнути лівою кнопкою миші усередині пунктирного прямокутника і видалити текст, що міститься в ньому, після чого ввести свій текст. Для роботи з об'єктом WordArt в MS Office Power Point 2007 з'являється допоміжна вкладка *Засоби малювання*, що містить вкладку *Формат*. Робота з об'єктом WordArt в Power Point аналогічна роботі з цим об'єктом в програмі MS Office Word 2007. На вкладці *Формат* розташовано 5 груп інструментів. Перерахуємо їх коротко і вкажемо призначення.

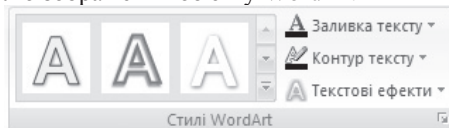
1. Група інструментів *Вставити фігури* призначена для вставки фігури або зміни форми напису WordArt



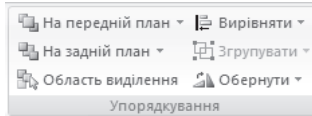
2. Група інструментів *Стилі фігур* призначена для налаштування візуального стилю фігури або лінії, вибору кольору заливки фону об'єкту WordArt або його контура



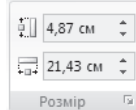
3. Група інструментів *Стилі WordArt* призначена для вибору способу заливки об'єкту WordArt або його контура, а також для завдання текстових ефектів і зміни стилю зображення об'єкту WordArt



4. Група інструментів *Упорядкування* дозволяє задати спосіб розміщення об'єкту WordArt щодо інших об'єктів, а також згрупувати декілька об'єктів, обернути, дзеркально відображувати або вирівняти по одному краю.



5. Група *Розмір* дозволяє задати розміри об'єкту WordArt



Для вставки в слайд малюнка, фігури, діаграми, графічного об'єкту SmartArt слід скористатися відповідними кнопками з групи інструментів *Зображення* на вкладці *Вставка*. Робота з цими об'єктами аналогічна роботі з ними в програмі MS Office Word 2007 і тому в даному пункті описуватися не буде.

Для вставки в слайд відеоролика або звукового повідомлення слід скористатися відповідно кнопками *Фільм* і *Звук*, що розкриваються, розташованими в групі інструментів *Медіакліпи* на вкладці *Вставка*. Команди кнопок, що розкриваються, дозволяють завантажити фільм або звук з файлу або з колекції кліпів.

Налаштування анімаційних ефектів для елементів слайдів за допомогою групи інструментів вкладки Анімація. При розміщенні елементів слайдів можна задати дії, які відбуватимуться з цими елементами у момент проглядання презентації, тобто задати анімаційний ефект для елементів слайду. Задати анімаційні ефекти для елементів слайду можна за допомогою груп інструментів, розташованих на вкладці *Анімація*. Група інструментів *Перехід до цього слайду* дозволяє налаштувати ефект появи слайду (поступове проявлення, проявлення через чорне, розчинення, поява знизу, поява зліва, поява зправа тощо). Після затримки покажчика миші над ефектом появи можна проглянути, як з'явиться слайд. Вибір здійснюється клацанням по ефекту лівою кнопкою миші. Галерея ефектів представлена на рисунку 6.4.5, а застосування ефекту *Розчинення* – на рисунку 6.4.6.

У групі інструментів *Перехід до цього слайду* можна також задати звуковий ефект, з яким з'являється слайд. Для цього в списку *Звук переходу*, що розкривається, треба вибрати звуковий ефект, наприклад, *Дзвіночки*. У списку *Швидкість переходу*, що розкривається, можна вибрати швидкість переходу до наступного слайду: повільно, середньо, швидко. У цій же групі інструментів можна задати спосіб зміни слайду – або змінювати слайди

презентації клацанням правої кнопки миші, або змінювати слайди автоматично через певний часовий проміжок.

Для налаштування анімаційного ефекту окремого елемента слайду треба виділити цей елемент і натиснути на кнопку *Налаштування анімації*. У правій частині екрану з'явиться однойменна панель з кнопкою *Додати ефект*, що розкривається і містить чотири групи ефектів.

1. Група *Вхід* містить ефекти, спостережувані при появі елемента на слайді.
2. Група *Виділення* містить ефекти виділення елемента при виконанні деякої дії, наприклад, клацання кнопкою миші.
3. Група *Вихід* містить ефекти зникнення елемента із слайду.
4. Група *Шляхи переміщення* містить ефекти переміщення елемента по слайду.

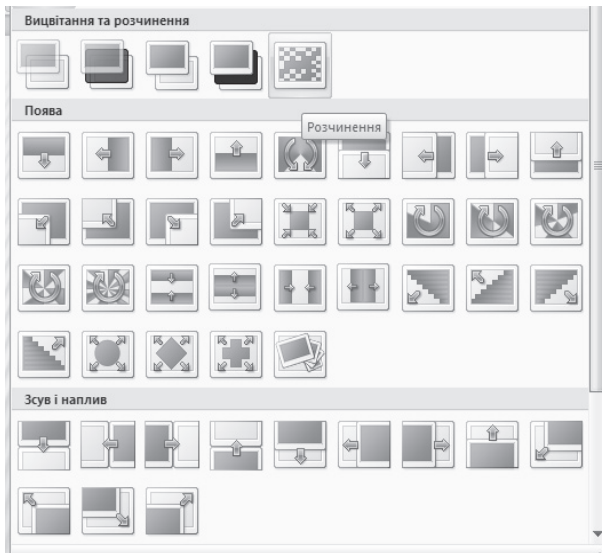


Рис.6.4.5. Галерея ефектів появи слайду з вибором ефекту Розчинення

Після вибору ефекту для елемента слайду кнопка *Додати ефект* стає кнопкою *Змінити*, що розкривається, список якої містить команди зміни призначеного для елемента ефекту. Нижче за кнопки *Змінити* і *Видалити* з'являються три списки, що розкриваються: *Початок*, *Напрямок*, *Швидкість*.

Список *Початок*, що розкривається, дозволяє вказати дію, у відповідь на яку буде застосований ефект. Список *Напрямок*, що розкривається, дозволяє задати напрям появи ефекту. Список *Швидкість*, що розкривається, дозволяє встановити швидкість виконання ефекту. У нижній частині панелі *Настроювання анімації* є кнопки *Перегляд* і *Показ слайдів*, скориставшись якими можна проглянути встановлений для елемента слайду ефект. Проглядання ефектів можна виконати також, натиснувши на кнопку *Попередній перегляд* з однойменної групи інструментів, розташованої на вкладці *Анімація*.

6.4.3. Настроювання режиму перегляду презентації

Для перегляду готової презентації слід натиснути функціональну клавішу F5 або натиснути на кнопку *Показ слайдів*, розташовану в групі інструментів *Режими перегляду презентації* вкладки *Вигляд*. Проглянути презентацію можна також скориставшись кнопками *З початку* або *З поточного слайду*, розташованими в групі інструментів *Розпочати показ слайдів* на вкладці *Показ слайдів*.

Перед прогляданням презентації рекомендується налаштувати слайди презентації за допомогою кнопки *Настроювання показу слайдів*, розташованої в групі інструментів *Настроювання* на вкладці *Показ слайдів*. В результаті на екрані з'явиться діалогове вікно ***Настроювання презентації***, представлене на рисунку 6.4.6.

У розділі *Показ слайдів* діалогового вікна ***Настроювання презентації*** можна вибрати один з наступних режимів:

1) керований доповідачем (у весь екран) – перемикає слайдів по клацанню миші або за допомогою клавіатури;

2) керований користувачем (вікно) – слайди презентації демонструватимуться з елементами програми MS Office Power Point 2007;

3) автоматичний (у весь екран) - слайди чергуються через встановлені часові інтервали.

У розділі *Параметри показу* діалогового вікна ***Настроювання презентації*** можна вибрати один з наступних режимів:

1) безперервний цикл до натискання клавіші Esc;

2) без мовленнєвого супроводу – відключає звукові ефекти в презентації;

3) без анімації – відключає анімаційні ефекти в презентації.

У розділі *Слайди* можна вказати діапазон слайдів для показу (всі слайди або з одного номера по інший).

У розділі *Зміна слайдів* можна вказати спосіб зміни слайдів презентації (у ручну або за часом).

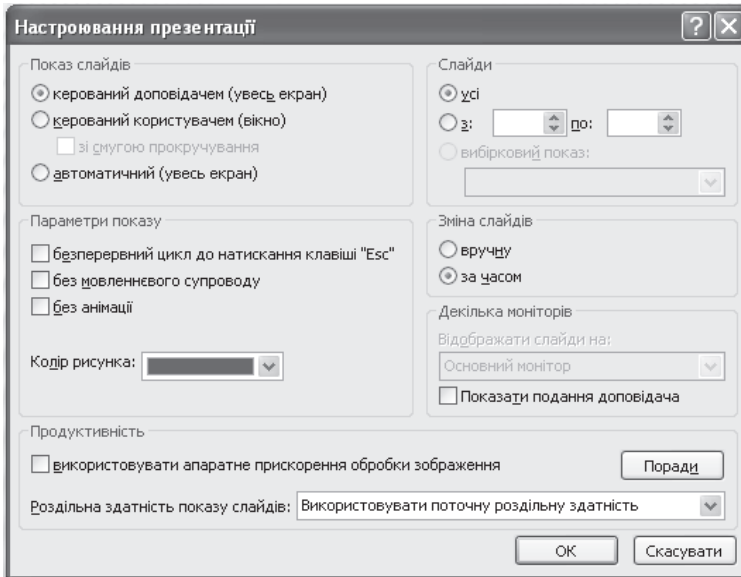


Рис.6.4.6. Диалогове вікно Настроювання презентації

Питання і задання для самостійної роботи

1. Створіть презентацію про методи розв'язання задачі лінійного програмування.
2. Створіть презентацію про основні поняття економічної кібернетики
3. Створіть презентацію про покоління обчислювальної техніки
4. Створіть презентацію про системи числення і алгоритми переказу чисел з однієї системи числення в іншу
5. Створіть презентацію про роботу машини Тюрінга

6.5. Microsoft Office Access 2007

При проектуванні бази даних, перш за все, вирішуються дві задачі: організація логічної структури всієї бази даних, яка (структура) визначає типи зв'язків між даними, і організація структури самих даних. *Логічна структура бази даних* також називається її логічним уявленням і може мати вид таблиці, дерева або мережі. Окрім логічного, існує також фізичне представлення бази даних. *Фізичне представлення бази даних* – це спосіб її розміщення в пам'яті комп'ютера, як правило, на жорсткому магнітному диску. Для розробника бази даних і її користувача не має значення, в якому місці диска зберігається той або інший запис, як саме розташовані дані в комп'ютері. Наприклад, таблиця може зберігатися цілком (кожен її рядок є записом файлу) або по стовпцях (кожен стовпець, відповідний однотипному полю запису, – у вигляді окремого файлу). Головне при цьому – забезпечити ті зв'язки, які відображені в логічній структурі бази даних.

Програми, що забезпечують зв'язок між фізичними даними і їх логічним описом, а також обробку даних, називаються *системою управління базами даних* (СУБД).

Існують декілька вимог до баз даних, основними з яких є несуперечність даних і їх цілісність. База даних може використовуватися різними категоріями користувачів, проте доступ до операцій виправлення даних, занесення нових даних та їх видалення повинна мати одна людина або обмежена група. Для захисту даних використовується так званий «замок секретності», який може зберігатися в довільному записі або вводитися як пароль на початку роботи з базою даних.

Сучасні СУБД дозволяють швидко описати структуру даних – запис і її поля, створити таблицю (двовимірний файл) і організувати пошук і сортування даних. Проте, для опису складної структури доводиться застосовувати спеціальні засоби, наприклад, представляти мережеву модель у вигляді таблиць прямого і зворотного перегляду.

Одна з поширених СУБД для персональних комп'ютерів називається Access.

Програма Microsoft Office Access 2007 – це потужний засіб для роботи з базами даних, реалізованими у вигляді сукупності таблиць, кожна з яких містить записи у вигляді рядків і поля у вигляді стовпців, з можливістю організації зв'язків між таблицями через однойменні поля. По суті, в Microsoft Office Access 2007 реалізована можливість роботи з реляційними базами даних. Кожна пара таблиць реляційної бази даних може бути зв'язана за допомогою одного з наступних видів стосунків: відношення «один-до-одного» (кожному запису в першій таблиці зіставляється рівно один запис в

другій таблиці і, навпаки, кожному запису в другій таблиці зіставляється рівно один запис в першій таблиці); відношення «один-до-багатьох» (кожному запису в першій таблиці зіставляється декілька записів в другій таблиці і кожному запису в другій таблиці зіставляється рівно один запис в першій таблиці); відношення «багато-до-багатьох» (кожному запису в першій таблиці зіставляється декілька записів в другій таблиці і кожному запису в другій таблиці зіставляється декілька записів в першій таблиці).

Основними об'єктами будь-якої бази даних Microsoft Office Access 2007 є таблиці, запити, форми, звіти, сторінки доступу до даних, макроси і модулі. Коротко охарактеризуємо основні об'єкти бази даних.

- *Таблиця* – структура, в якій зберігається основна інформація бази даних. Вона складається із записів (рядків) і полів (стовпців). Часто база даних реалізується у вигляді сукупності зв'язаних таблиць.
- *Запит* – таблиця, сформована на основі однієї або декількох таблиць бази даних, яка містить інформацію, що задовольняє деякому критерію.
- *Форма* – це діалогове вікно, за допомогою якого реалізований своєрідний «діалог» між таблицями бази даних і користувачем.
- *Звіт* – електронний варіант документа, сформованого на основі бази даних і підготовленого для виводу на друк.

У Microsoft Office Access 2007 за допомогою сторінок доступу до даних реалізована можливість роботи з базою даних через локальну мережу або Інтернет, надана можливість створення призначених для користувача макросів і модулів з використанням вбудованої мови програмування VBA.

При запуску **Microsoft Office Access 2007** на екрані з'являється стартове вікно *Початок роботи з Microsoft Office Access*, представлене на рисунку №.6.5.1. За допомогою цього зручного вікна користувач може вибрати варіант створення бази даних – створення нової бази даних, створення бази даних з використанням шаблону або відкриття існуючої бази даних.

Після вибору варіанту створення бази даних і збереження бази даних на екрані з'явиться звичний стрічковий інтерфейс з розміщенням в лівому верхньому кутку кнопки *Microsoft Office*, а поруч – панелі швидкого доступу. Розглянемо основні вкладки стрічки програми Microsoft Office Access 2007 і коротко охарактеризуємо їх функціональне призначення.

Вкладка *Основне* призначена для оформлення бази даних (вибір шрифту, зображення, розміру і тому подібне), роботи з буфером обміну, пошуку, заміни і сортування записів бази даних.

Вкладка *Створити* призначена для створення таблиць, шаблонів таблиць, списків SharePoint, форм, звітів, запитів, макросів, а також використання при їх створенні конструкторів таблиць, форм, звітів, запитів.

Вкладка *Зовнішні дані* призначена для експорту та імпорту даних з інших застосувань, а також для збору даних і роботи із списками SharePoint.




Рис. 6.5.1. Запуск стартового вікна Microsoft Office Access 2007

Вкладка *Знаряддя бази даних* призначена для відображення схеми даних, аналізу і захисту даних і тому подібне.

В нижній частині робочої області розміщений статусний рядок з

кнопками *Режим форми* , *Режим розмічування*  і *Конструктор*

 в її правій частині.

6.5.1. Створення нової бази даних в MS Office Access 2007

Перед реалізацією власної бази даних в Microsoft Office Access 2007 користувачеві необхідно продумати призначення бази даних, визначитися з об'єктами та їх атрибутами, визначитися з типами полів, відповідних атрибутам, установкою ключів і індексів, описати взаємозв'язки між об'єктами-таблицями бази даних, продумати форму представлення результатів роботи з базою даних, тобто виконати інфологічне проектування бази даних.

Створимо спрощену базу даних ВИКЛАДАЧІ ЕКОНОМІЧНОГО ФАКУЛЬТЕТУ, що складається з трьох таблиць-об'єктів: таблиця КАФЕДРА, таблиця ВИКЛАДАЧ, і таблиця РОЗКЛАД. Опишемо поля (атрибути) і їх типи для кожної таблиці-об'єкту.

Таблиця-об'єкт КАФЕДРА містить наступні поля:

- 1) Код кафедри (тип поля текстовий, 2 символи);
- 2) Назва (тип поля – текстовий, 40 символів);
- 3) Прізвище, ім'я, по батькові завідувача_кафедри (тип поля – текстовий, 50 символів);
- 4) номер аудиторії кафедри (тип поля – текстовий, 5 символів).

Приклад таблиці-об'єкту КАФЕДРА:

Код кафедри	Назва	Прізвище, ім'я, по батькові зав. кафедри	Номер аудиторії кафедри
01	Банківська справа	Білоненко Павло Петрович	115-А
02	Бухгалтерський облік і аудит	Авраменко Олексій Тихонович	120-А
03	Міжнародна економіка	Рівенко Людмила Олексіївна	117-А
04	Економічна кібернетика	Іваненко Ганна Вікторівна	125-А

Таблиця-об'єкт ВИКЛАДАЧ містить наступні поля:

- 1) Прізвище ім'я по батькові викладача (тип поля – текстовий, 50 символів);
- 2) Стать (тип поля – текстовий, 1 символ);
- 3) Вчений ступінь (тип поля – текстовий, 10 символів);
- 4) Вчене звання (тип поля – текстовий, 10 символів);

- 5) Посада (тип поля – текстовий, 15 символів);
- 6) Код кафедри (тип поля – Майстер підстановок);
- 7) Дата народження (тип поля – дата й година);
- 8) Місце народження (тип поля – текстовий, 50 символів);
- 9) Адреса мешкання (тип поля – текстовий, 50 символів);
- 10) Контактний телефон (тип поля – текстовий, 15 символів).

Приклад таблиці-об'єкту ВИКЛАДАЧ:

Прізвище, ім'я, по батькові викладача	стать	Вчен ний ступ інь	Вчене званн я	Посад а	Код каф едри	Дата народже ння	Місце народж ення	Адреса мешкання	Контактний телефон
Іваненко Анатолій Вікторови ч	м	к.е.н	доцен т	доцен т	01	15.02.195 6	м. Дніпро петровс ьк	м. Дніпропетр овськ, вул. Леніна, 25А, кв.13	0675431278
Іваненко Ганна Вікторівна	же	д.е.н	профе сор	профе сор	04	18.01.196 0	м. Дніпро петровс ьк	м. Дніпропетр овськ., вул. Леніна, 25А, кв.13	0501237819
Трофімов Сергій Олександр ович	м	к.е.н	немає	старш ий викал дач	02	30.03.197 3	м. Київ	м. Харків, вул. Київська, 15, кв.28	0934561879
Петренко Олександр Сергійови ч	м	д.ф.- м.н.	профе сор	профе сор	03	19.02.195 3	м. Сімфер ополь	м. Сімферопол ь, просп. Вернадського, 12, кв.5	0652341768
Дмітренко Галина Василівна	же	к.т.н	доцен т	доцен т	04	23.09.197 9	м. Сімфер ополь	м. Сімферопол ь, вул. Набережна, 29, кв.14	0504318923

Таблиця-об'єкт РОЗКЛАД містить наступні поля:

- 1) Прізвище, ім'я, по батькові викладача (тип поля – Майстер підстановок, 50 символів);
- 2) Учбова дисципліна, час початку і закінчення заняття (тип поля – текстовий, 35 символів);
- 3) День тижня (тип поля – текстовий, 15 символів);
- 4) Дата проведення заняття (дата й година);
- 5) Номер аудиторії, в якій відбудеться заняття (тип поля – текстовий, 5 символів).

Приклад таблиці-об'єкту РОЗКЛАД:

Прізвище, ім'я, по батькові викладача	Учбова дисципліна, час початок і закінчення заняття	День тижня	Дата проведення заняття	Номер аудиторії, в якій проходить заняття
Іваненко Анатолій Вікторович	Фінанси, 12:50-14:10	понеділок	16.02.2009	307-А
Дмітренко Галина Василівна	Економетрія, 12:50-14:10	понеділок	16.02.2009	304-А
Дмітренко Галина Василівна	Економіко-математичне моделювання, 14:20-15:40	вівторок	17.02.2009	232-А

Встановимо зв'язки між таблицями-об'єктами. Таблиця-об'єкт ВИКЛАДАЧ буде пов'язана з таблицею-об'єктом КАФЕДРА через поле Код кафедри, тип зв'язку – багато до одного. Таблиця-об'єкт ВИКЛАДАЧ буде пов'язана з таблицею-об'єктом РОЗКЛАД через поле Прізвище, ім'я, по батькові викладача, тип зв'язку – один до багатьох (за умови, що кожен викладач працює на одній кафедрі). Схема зв'язків між таблицями бази даних представлена на рисунку 6.5.2.

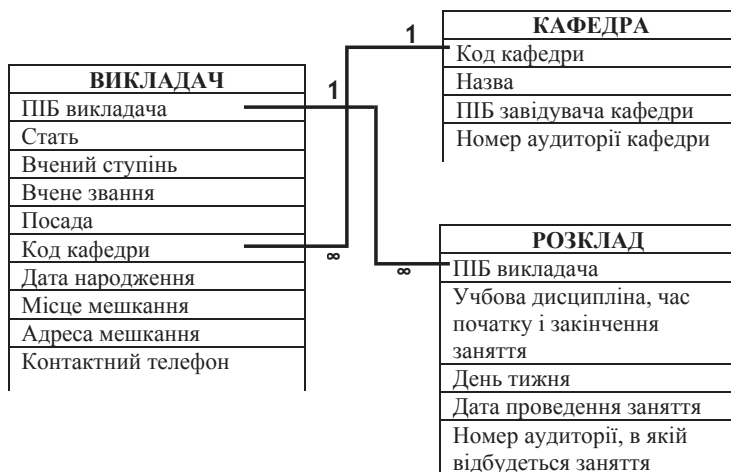


Рис.6.5.2. Схема зв'язків між таблицями бази даних ВИКЛАДАЧІ ЕКОНОМІЧНОГО ФАКУЛЬТЕТУ

Перейдемо до опису реалізації бази даних в Microsoft Office Access 2007. Користувач може створити нову таблицю трьома способами: за допомогою безпосереднього введення даних, на основі готового шаблону або за допомогою конструктора. Перший спосіб вважається за універсальний, хоча і трудомісткий спосіб створення бази даних. Він дозволяє створювати бази даних різного ступеня складності і надає користувачеві можливість визначення типів полів і видів зв'язків між таблицями. Другий спосіб зазвичай використовується для створення простих баз даних, а також в тому випадку, якщо користувачеві важко з визначенням типів полів. Третій спосіб використовується, коли користувач створює часто використовувану базу даних, для якої розроблений шаблон.

Розглянемо на прикладах створення нової таблиці за допомогою кожного з перерахованих трьох способів.

Створення бази даних за допомогою Конструктора. В діалоговому вікні *Початок роботи з Microsoft Office Access* слід натиснути на кнопку *Нова пуста база даних* або натиснути на кнопку *Microsoft Office* і вибрати команду *Створити*. В правій частині вікна з'явиться діалогове вбудоване вікно, в якому необхідно ввести ім'я файлу бази даних з можливістю установки шляху для розміщення цього файлу і натиснути кнопку *Створити* (рисунок 6.5.3).

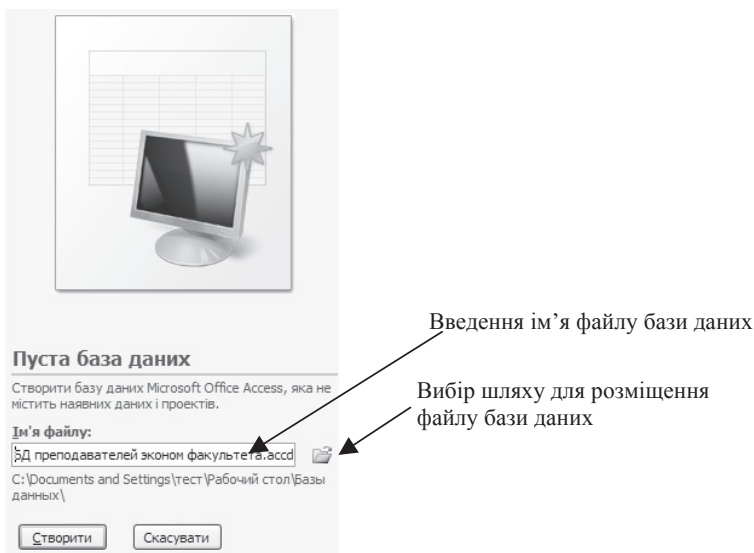


Рис.6.5.3. Створення порожньої бази даних

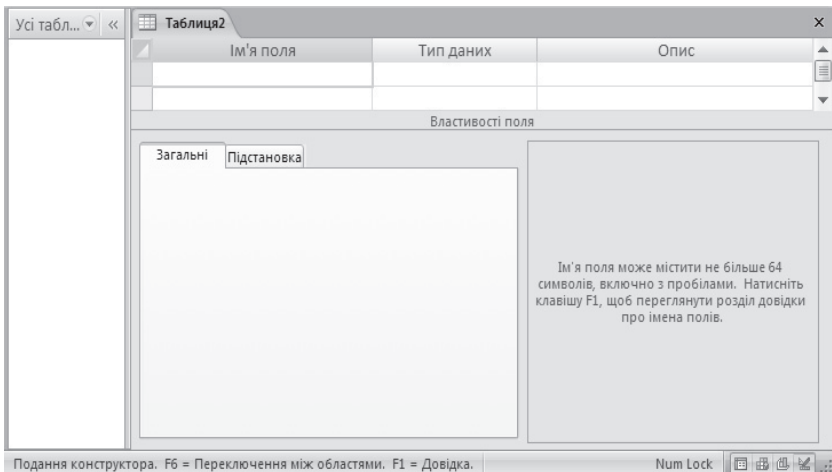


Рис.6.5.4. Вікно Конструктора таблиць

Опишемо створення таблиці КАФЕДРА. У стовпці *Ім'я поля* клацнемо лівою кнопкою миші на найверхній порожній клітинці (рисунок 6.5.4). Після появи курсора в цій клітинці введемо назву першого поля Код кафедри таблиці КАФЕДРА. Для переходу до наступного елементу таблиці слід натиснути клавішу *Tab*, або клацнути лівою кнопкою миші по верхній порожній клітинці стовпця *Тип даних*, або натиснути клавішу *Enter* і вибрати в списку, що розкривається, тип *Текстовий*. Конструктор таблиць дозволяє (при необхідності) встановити окрім типа текст, наступні типи полів: примітка, число, дата й година, грошова одиниця, авто нумерація, так/ні, об'єкт OLE, гіперпосилання, вкладення, майстер підстановок.

Повернемося до нашого прикладу. Після установки типа *Текст* для поля Код кафедри в розділі *Властивості поля* на вкладці *Загальні* можна встановити розмір поля, рівний двом символам. За умовчанням розмір текстового поля містить 255 символів. У стовпці *Опис* для поля Код кафедри користувач може ввести коротке пояснення про призначення цього поля. Заповнення стовпця *Опис* для полів є обов'язковим.

Аналогічно, слід увести всі інші поля таблиці КАФЕДРА. Для повного завдання структури таблиці КАФЕДРА залишилися тільки визначити ключі й індекси. У якості ключового поля встановимо поле Код кафедри. Для цього слід зробити активним поле Код кафедри й натиснути по ньому правою кнопкою миші. У контекстному меню, що з'явилося, слід вибрати пункт

Ключове поле. Установити ключове поле можна інакше. При активному полі



Код кафедри нажати на кнопку *Ключове поле* на стрічці вкладки *Конструктор*. З лівої сторони від назви поля з'явиться зображення ключа. Зверніть увагу на те, що у властивості Індєксовано для ключового поля встановлене значення Так (Без повторень). Це означає, що значення поля Код кафедри в таблиці КАФЕДРА не будуть повторюватися (рисунок 6.5.5).

Збережемо таблицю КАФЕДРА в базі даних ВИКЛАДАЧІ ЕКОНОМІЧНОГО ФАКУЛЬТЕТУ. Для цього, наприклад натиснемо на кнопку *Зберегти* на панелі швидкого доступу. При цьому на екрані з'явиться діалогове вікно *Зберегти як...*, в якому слід ввести ім'я таблиці, – КАФЕДРА і натиснути на кнопку *ОК* (рисунок 6.5.6).

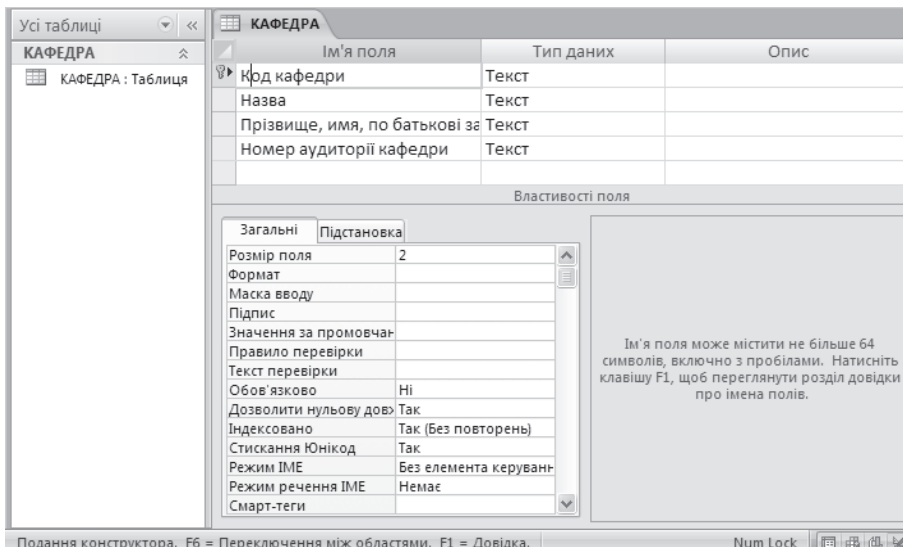


Рис.6.5.5. Установка ключового поля Код кафедри

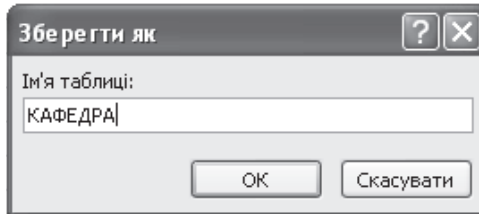


Рис.6.5.6. Збереження таблиці КАФЕДРА

Аналогічно за допомогою Конструктора можна створити таблиці ВИКЛАДАЧ і РОЗКЛАД бази даних ВИКЛАДАЧІ ЕКОНОМІЧНОГО ФАКУЛЬТЕТУ.

Використання Майстра підстановок для заповнення полів. При створенні таблиці ВИКЛАДАЧ за допомогою Конструктора поле Код кафедри має тип даних Майстер підстановок. При виборі зі списку, що розкривається, для цього поля типу Майстер підстановок на екрані з'являється перше діалогове вікно **Майстра підстановок**, представлене на рисунку 6.5.7.

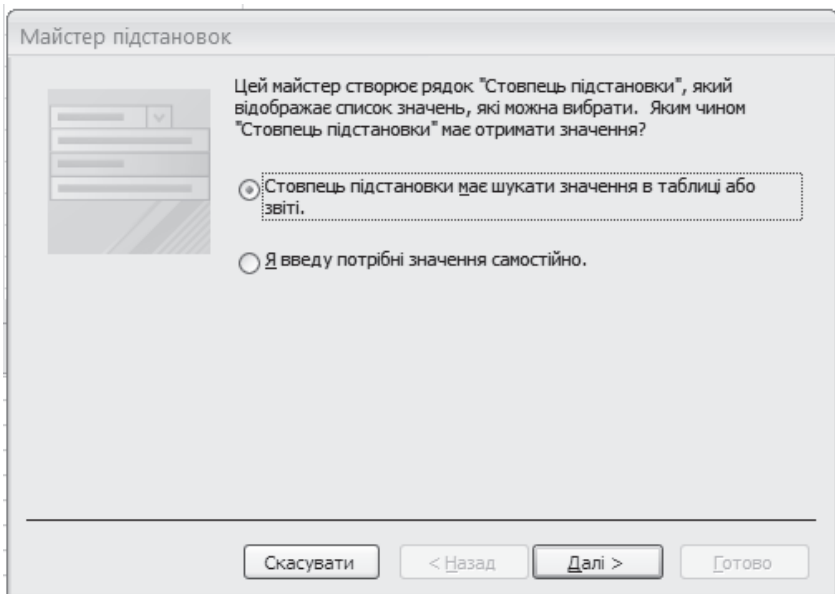


Рис.6.5.7. Перше діалогове вікно Мастера подстановок

У першому діалоговому вікні *Майстра підстановок* слід вибрати пункт *Стовпець* підстановки має шукати значення в таблиці або звіті (пункт обраний за замовчуванням), як показано на рисунку 6.5.7, і натиснути на кнопку *Далі*. На екрані з'явиться друге діалогове вікно *Майстра підстановок*, що містить перелік таблиць бази даних, у якому необхідно вибрати таблицю, з якої будуть запозичатися дані (рисунок 6.5.8), і натиснути на кнопку *Далі* для підтвердження вибору.

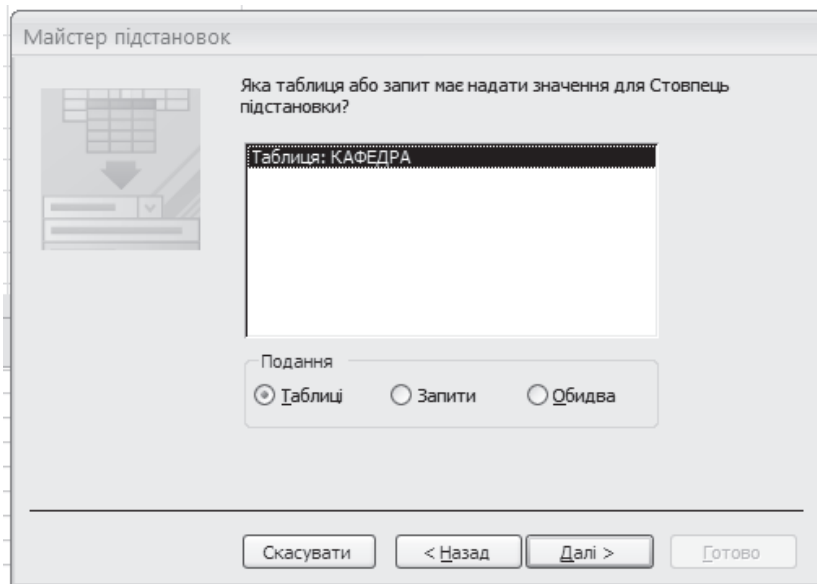



Рис.6.5.8. Друге діалогове вікно Мастера подстановок

У результаті на екрані з'явиться третє діалогове вікно *Майстра підстановок* (рисунок 6.5.9), у якому необхідно вказати поле таблиці, з якого будуть запозичатися дані. Третє діалогове вікно *Майстра підстановок* розділене на дві частини: *Доступні поля* (усі поля таблиці, обраної в другому діалоговому вікні *Майстра підстановок*) і *Вибрані поля*. Виберемо поле *Код кафедри* й натиснемо на кнопку  для перенесення цього поля в розділ *Вибрані поля*, після чого підтвердемо дію натисканням на кнопку *Далі*.

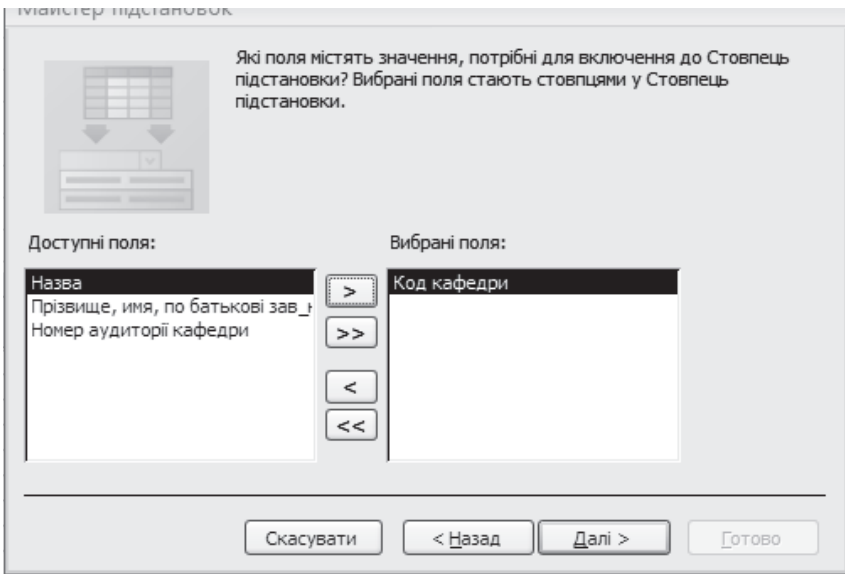


Рис.6.5.9. Третє діалогове вікно Мастера подстановок

В результаті на екрані з'явиться четверте діалогове вікно **Майстра підстановок**, представлене на рисунку 6.5.10, в якому треба вказати порядок сортування (якщо це необхідно) елементів списку і підтвердити вибір натисненням на кнопку *Далі*. У п'ятому діалоговому вікні **Майстра підстановок**, що з'явилося, користувач може встановити ширину стовпця простим перетягуванням за допомогою покажчика миші правої межі стовпця, після чого слід натиснути на кнопку *Далі*. В результаті на екрані з'явиться останнє шосте діалогове вікно **Майстра підстановок**, в якому слід вказати підпис для об'єкту «Стовпець підстановки» і натиснути на кнопку *Готово*.

Якщо на момент створення стовпця підстановок таблиця містила незбережений дані, то на екрані з'явиться повідомлення про необхідність збереження таблиці. У відповідь на це повідомлення треба відповісти «Так» і при необхідності (якщо створюється нова таблиця) ввести ім'я таблиці.

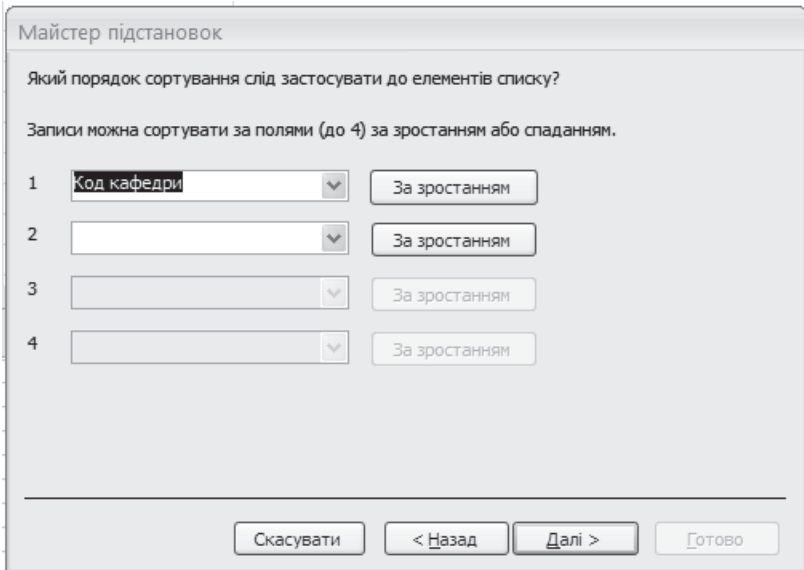


Рис.6.5.10. Четверте діалогове вікно Мастера підстановок

Продемонструємо, як працює стовпець підстановки. На вбудованій панелі в лівій частині екрану в розділі *Усі таблиці* клацнемо двічі лівою кнопкою миші по таблиці ВИКЛАДАЧ і приступимо до заповнення полів цієї таблиці. У стовпці Код кафедри з'явиться список, що розкривається, в якому користувач одним клацанням миші може вибрати код, як показано на рисунку 6.5.11.

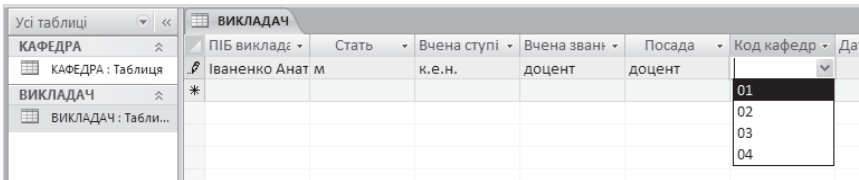


Рис.6.5.11. Приклад використання стовпця підстановки

Створення бази даних за допомогою введення даних. Цей спосіб зручний в тому випадку, якщо користувач з якихось причин не визначився з властивостями всіх полів нової таблиці. Розглянемо створення таблиці

КАФЕДРА цим способом. Для появи порожньої табличної форми необхідно натиснути на кнопку «Office», вибрати команду *Створити* і у вбудованій панелі (рисунок 6.5.3), що з'явилася, задати ім'я файлу бази даних. У порожній табличній формі, що з'явилася, клацнемо лівою кнопкою миші по полю Ідентифікатор і натиснемо праву кнопку миші. У допоміжному меню, що з'явилося, виберемо команду *Перейменувати стовпець* і введемо нову назву Код кафедри. Аналогічно, клацнувши лівою кнопкою миші по заголовку *Додати нове поле* і вибравши команду *Перейменувати стовпець* введемо ім'я поля – Назву. Натиснемо клавішу *ENTER* і продовжимо вводити імена полів таблиці КАФЕДРА, як показано на рисунку 6.5.12. Після заповнення всіх полів першого запису слід натиснути клавішу *ENTER* і продовжити введення решти записів.

Заповнення порожньої табличної форми нічим не відрізняється від заповнення таблиці Excel. Переміщення від клітинки до клітинки можна здійснювати або клацанням миші, або натисненням клавіші *Tab*. Після заповнення полів всіх записів таблицю слід зберегти.

Створення бази даних за допомогою готового шаблону. У Microsoft Office Access 2007 реалізована можливість завантаження готового шаблону з Інтернету, а також в категорії *Зразок* можна вибрати повністю створену базу даних Northwind 2007, розроблену спеціально для навчання роботі з базами даних в Microsoft Office Access.

Перерахуємо локальні шаблони, представлені в Microsoft Office Access 2007: Активи, Завдання, Кадри, Контакти, Маркетинг, Питання, Події, Продажі, Проекти, Студенти.

Шаблон *Активи* використовується для управління відомостями про активи, включаючи відомості про власників активів.

Шаблон *Завдання* використовується для управління відомостями про робочі завдання, які мають бути виконані робочою групою.

Шаблон *Кадри* використовується для управління відомостями про співробітників разом з їх координатами в разі непередбаченої ситуації і відомостями про рівень освіти.

Шаблон *Контакти* використовується для управління відомостями про людей, з якими працює група, наприклад, відомості про клієнтів або постачальників.

Шаблон *Маркетинг* використовується для управління відомостями про маркетингові проекти.

Шаблон *Питання* використовується для управління відомостями про наявні питання і проблеми з урахуванням їх пріоритету.

Шаблон *Події* використовується для управління відомостями про майбутні наради, засідання та інші події.

Шаблон *Продажі* використовується для управління відомостями про майбутні продажі.

Шаблон *Проекти* використовується для управління відомостями про проекти і призначення завдань різним споживачам.

Шаблон *Студенти* використовується для управління відомостями про студентів.

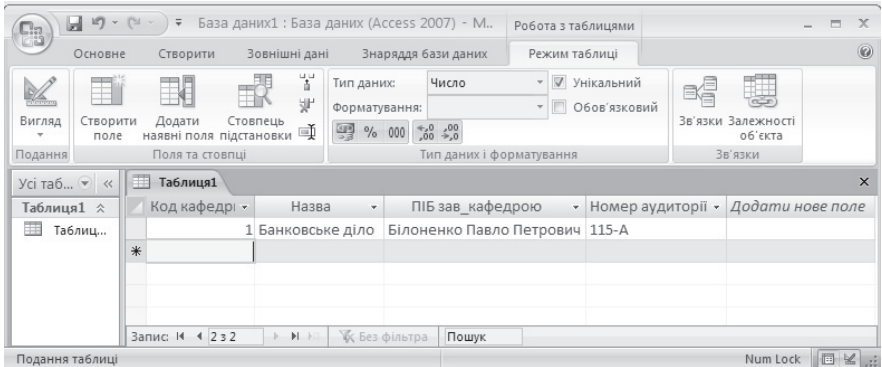


Рис.6.5.12. Приклад заповнення порожньої табличної форми

Виберемо клацанням лівої кнопки миші, наприклад, шаблон *Кадри* і натиснемо на кнопку *Створити* у вбудованій панелі, що з'явилася в правій частині вікна. На екрані з'явиться шаблон таблиці (рисунок 6.5.13), відповідний вибраному шаблону *Кадри*, який необхідно буде заповнити і зберегти.

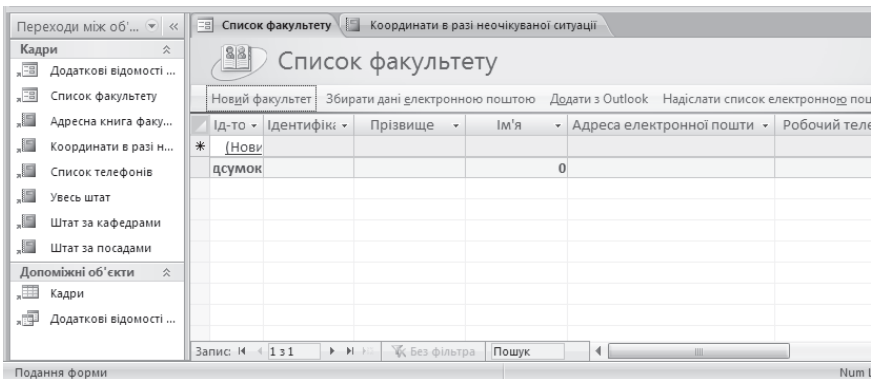


Рис.6.5.13. Створення бази даних на основі шаблону *Кадри*

Відкриття бази даних і додавання нової таблиці. Для відкриття заздалегідь створеної бази даних слід натиснути на кнопку «Office» і в списку команд, що розкрився, вибрати команду *Відкрити*, після чого в діалоговому вікні *Відкриття файлу бази даних*, що з'явилося, вказати шлях до файлу бази даних і натиснути на кнопку *ОК*. В результаті буде завантажена база даних зі всіма складовими її таблицями. Додати нову таблицю в базу даних можна, перемкнувшись на вкладку *Створити* і вибравши в групі інструментів *Таблиці* один із способів створення нової таблиці (кнопка *Таблиця*, кнопка *Шаблони таблиць*, кнопка *Списки SharePoint*, кнопка *Конструктор таблиць*).

6.5.2. Зв'язування таблиць бази даних

Визначення ключового поля. Під *ключовим полем* або *ключем*, як правило, розуміють одне або декілька полів, комбінація яких однозначно визначає кожен запис таблиці. Якщо в Microsoft Access 2007 деяке поле визначити як ключове, то автоматично буде виключено введення порожніх значень, що повторюються.

У Microsoft Access 2007 виділяють три типи ключових полів: лічильник, простий ключ, складений ключ. *Ключове поле-лічильник* – це поле зі встановленим типом даних Лічильник. Значення поля лічильника заповнюються автоматично або послідовними, або випадковими значеннями. *Простий ключ* – це одне поле, значення якого унікальні. *Складений ключ* – це сукупність декількох полів, значення яких (в сукупності, але не поодиноці) однозначно визначають кожен запис таблиці. Приклад установки ключового поля показаний на рисунку 6.5.5. Ключові поля використовуються для швидкого пошуку і установки зв'язків між даними з різних таблиць.

Створення індексів. Під *індексом* розуміють спеціальний засіб, що забезпечує швидкий доступ до даних в таблиці на основі значень одного або декількох полів. Індексом є впорядкований список значень і посилань на ті записи, в яких зберігаються ці значення. Розрізняють прості і складені індекси. *Простий індекс* – це індекс, побудований по одному полю таблиці. *Складений індекс* – це індекс, побудований по декількох полях таблиці. Будь-яке ключове поле таблиці індексується автоматично.

Розглянемо приклад створення простого індексу по полю ПІБ викладача з таблиці РОЗКЛАД. Для цього перейдемо в режим Конструктора для таблиці РОЗКЛАД. Виберемо таблицю РОЗКЛАД із списку Усі таблиці бази даних. Перейдемо на вкладку *Режим таблиці*, потім натиснемо на

кнопку *Вигляд*, що розкривається, розташовану в групі інструментів *Подання* і виберемо команду *Конструктор*, як показано на рисунку 6.5.14.

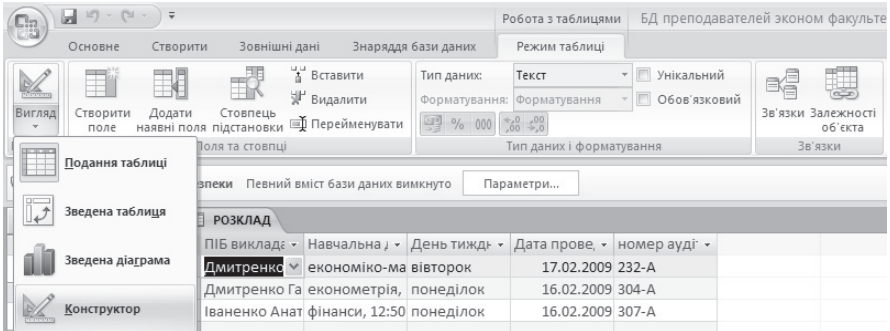


Рис.6.5.14. Вибір команди Конструктор

Після переходу в режим *Конструктора* слід виділити Ім'я поля ПІБ викладача і перемкнутися на вкладку *Загальні* в розділі *Властивості поля*. Клацнути лівою кнопкою миші в полі Індексоване. У списку значень, що розкрився, для цього поля слід вибрати значення Так (Повторення дозволені), як показано на рисунку 6.5.15. Вибір такого значення пояснюється тим, що один і той же викладач може вести декілька дисциплін, тому в розкладі може неодноразово повторюватися ПІБ викладача.

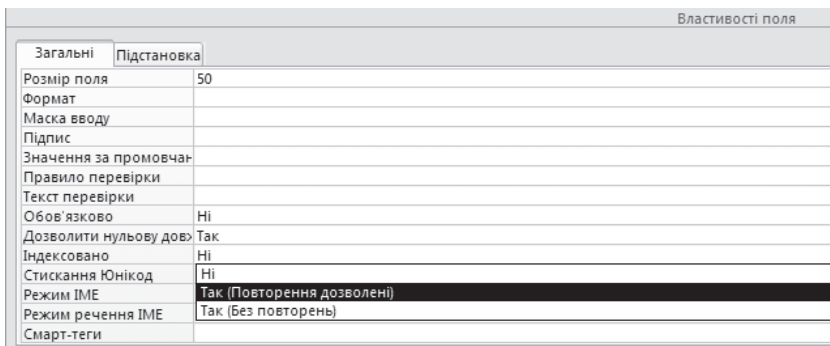


Рис.6.5.15. Приклад установки індексу по полю ПІБ викладача в таблиці РОЗКЛАД

Встановимо для таблиці ВИКЛАДАЧ бази даних, представленої на рисунку 6.5.2, ПІБ викладача – ключове поле, код кафедри - індекс із значенням Так (Повторення дозволені).

Для установки зв'язків між таблицями бази даних слід перемкнутися на вкладку *Знаряддя бази даних* у групу інструментів *Відобразити або приховати* й натиснути на кнопку *Зв'язки*. У результаті на екрані з'явиться вікно зв'язків, а на стрічці інструментів – допоміжна вкладка *Знаряддя для зв'язків* із вкладкою *Конструктор*. Для додавання таблиці й установки зв'язку слід натиснути на кнопку *Відобразити таблицю*, розміщену в групі інструментів *Зв'язок*. У результаті на екрані з'явиться діалогове вікно **Відображення таблиці** (рисунок 6.5.16), у якому треба вибрати ім'я таблиці й натиснути на кнопку *Додати*.

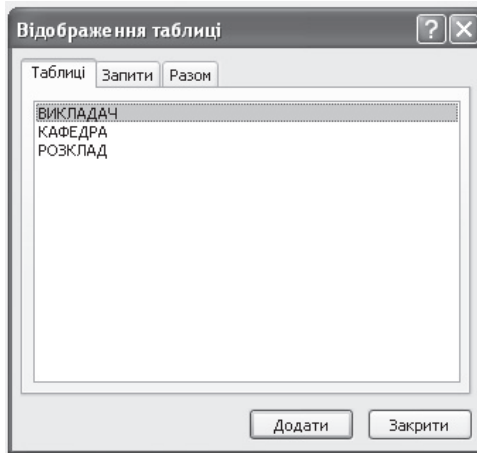


Рис.6.5.16. Діалогове вікно Відображення таблиці

Встановлені зв'язки між таблицями представлені на рисунку 6.5.17.

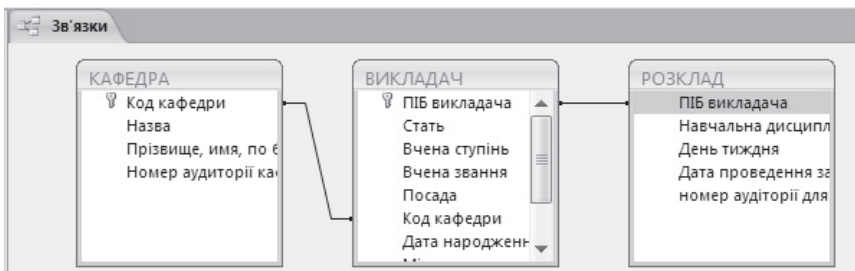


Рис.6.5.17. Приклад зв'язків між таблицями

Для зміни зв'язків між таблицями на вкладці *Конструктор (Знаряддя для зв'язків)* слід натиснути на кнопку *Змінити зв'язки*, розташовану в групі інструментів *Знаряддя*. На екрані з'явиться діалогове вікно *Редагування зв'язків*, представлене на рисунку 6.5.18.

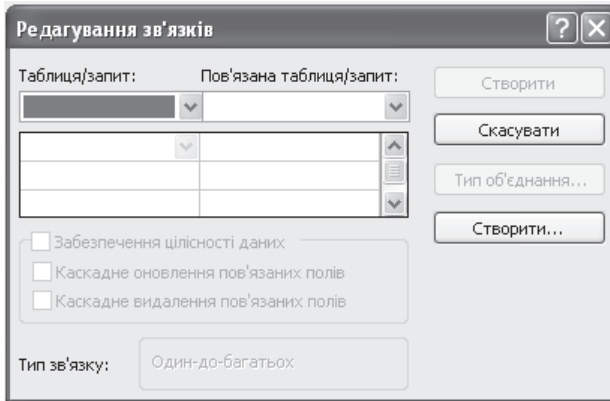


Рис.6.5.18. Діалогове вікно Редагування зв'язків

У діалоговому вікні *Редагування зв'язків* слід вибрати *Таблицю/запит* із запропонованого списку, що розкривається, і *Пов'язану таблицю/ запит*, забезпечити цілісність даних і натиснути на кнопку *Створити*.

6.5.4. Створення запитів в MS Office Access 2007

Під *запитом* розумітимемо звернення до даних з метою отримання деякої інформації і виконання дій з даними.

У MS Office Access 2007 реалізовано чотири способи створення нового запиту: за допомогою Майстра запитів; за допомогою Конструктора запитів; за допомогою запиту в режимі SQL-редактора (у цьому навчальному посібнику не розглядається); за допомогою існуючого фільтру.

Використання Майстра запитів. Організуємо простий запит - одержання відомостей ПІБ викладача, посаду й контактний телефон з таблиці

ВИКЛАДАЧ. Для цього перемкнемося на вкладку *Створити*, перейдемо в групу інструментів *Додатково* й натиснемо на кнопку *Майстер запитів*



На екрані з'явиться діалогове вікно **Новий запит** Майстра запитів на вибірку, представлене на рисунку 6.5.19. У цьому діалоговому вікні можна вибрати Майстер простих запитів (установлюється для нашого прикладу), Майстер перехресних запитів, Майстер пошуку повторюваних записів, Майстер пошуку незв'язаних записів, після чого натиснути на кнопку *OK*, підтвердивши вибір.

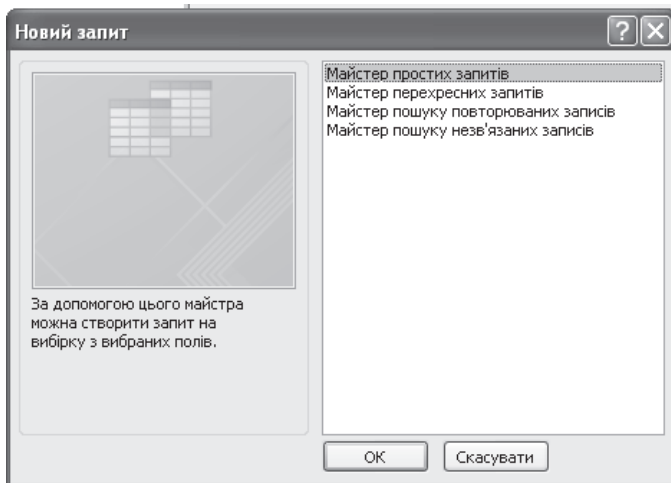



Рис.6.5.19. Перше діалогове вікно Новий запит Майстра запитів

Наступним на екрані з'явиться друге діалогове вікно **Майстер простих запитів**, представлене на рисунку 6.5.20. У цьому діалоговому вікні необхідно вказати таблицю, з якої будуть позичатися відомості, а також поля, значення яких беруть участь у запиті, тобто ПІБ викладача, посада, контактний телефон. Діалогове вікно містить Доступні поля таблиці й Вибрані поля. Поля, що беруть участь у запиті, повинні бути розміщені в розділі Вибрані поля (рисунок 6.5.20) за допомогою кнопки . Після

виконання всіх дій слід натиснути на кнопку *Далі*. У результаті на екрані з'явиться третє діалогове вікно *Майстер простих запитів* (рисунок 6.5.21), у якому слід увести назву запиту й натиснути на кнопку *Готово*.

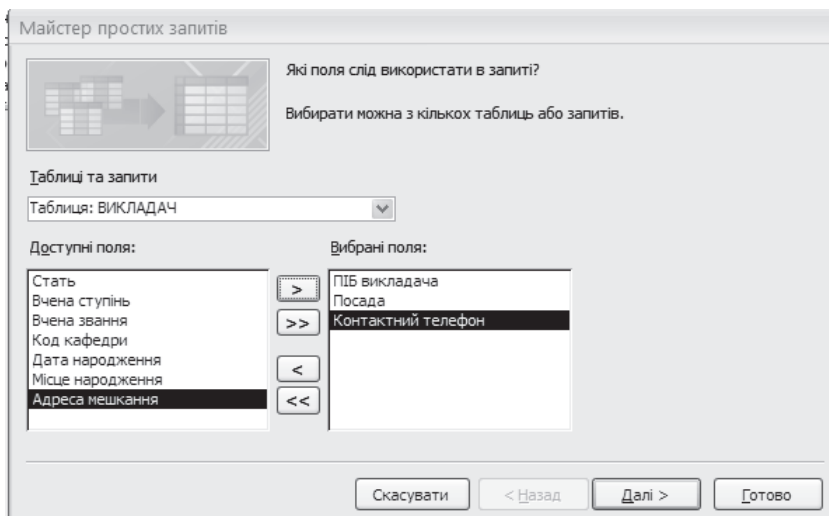


Рис.6.5.20. Друге діалогове вікно Майстер простих запитів

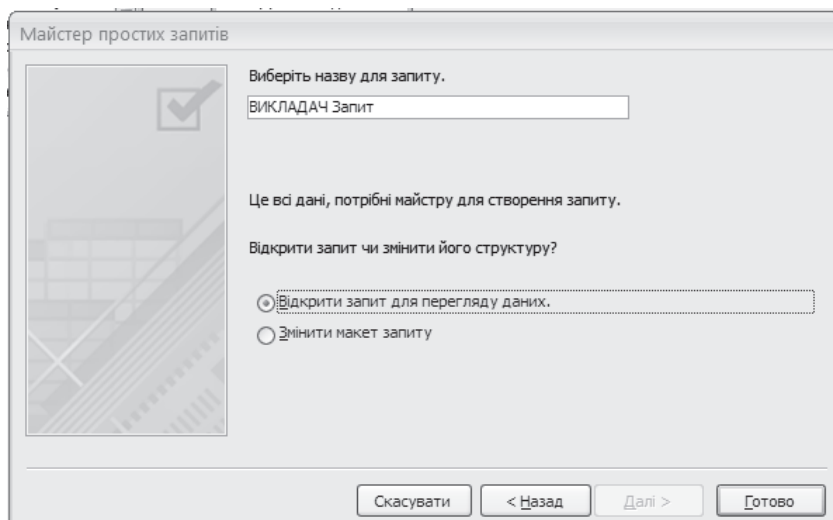


Рис.6.5.21. Третє діалогове вікно Майстер простих запитів

Результат виконання запиту на вибірку представлений на рисунку 6.5.22.

Усі таблиці		Зв'язки		
КАФЕДРА		ПІБ викладача	Посада	Контактний телефон
КАФЕДРА : Таблиця		Дмитренко Галіна Ва	доцент	80504318923
ВИКЛАДАЧ		Іваненко Анатолій Ві	доцент	80675431278
ВИКЛАДАЧ : Табли...		Іваненко Ганна Вікто	професор	80501237819
ВИКЛАДАЧ Запит		Петренко Олександр	професор	80652341768
РОЗКЛАД		Трофімов Сергій Оле	старший викл.	80934561879
РОЗКЛАД : Табли...		*		

Рис.6.5.22. Результат простого запиту на вибірку за допомогою Майстра запитів

Використання конструктора запитів. Організуємо за допомогою конструктора запитів наступний запит: вивести на екран за заданим кодом кафедри, наприклад, 04 відомості про завідувача цієї кафедри, а також ПІБ викладачів цієї кафедри, їх посади і дати проведення занять і дисципліни, що за ними закріплені. У такому запиті братимуть участь всі таблиці бази даних, представленої на рисунку 6.5.2.

Для запуску Конструктора запитів необхідно перемкнутися на вкладку *Створити*, перейти в групу інструментів *Додатково* й натиснути на



кнопку *Конструктор запиту*. На екрані з'явиться діалогове вікно **Відображення таблиці** (рисунк 6.5.23) і бланк запиту в Конструкторі (рисунк 6.5.25).

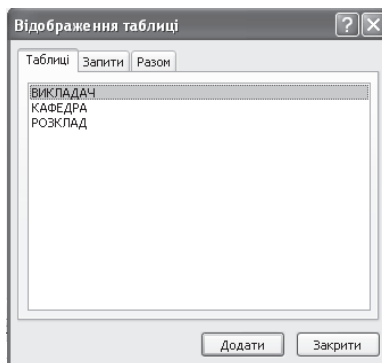


Рис.6.5.23. Діалогове вікно Відображення таблиці

Оскільки в запиті беруть участь відомості зі всіх таблиць, то за допомогою діалогового вікна треба додати всі запропоновані таблиці бази даних, після чого закрити це діалогове вікно.

Бланк запиту конструктора складається з двох частин: у верхній частині зазначені таблиці, відомості з яких повинні бути витягнуті, а в нижній частині перелічуються поля таблиць, значення яких повинні бути видані в якості результату запиту. Заповнимо нижню частину бланка запиту конструктора. У списку Поле, що розкривається, вкажемо ім'я поля Код кафедри з таблиці КАФЕДРА (рисунок 6.5.24). Відобразити код кафедри не треба, тому слід зняти прапорець для Відображення, у якості значення критерію слід ввести 04.

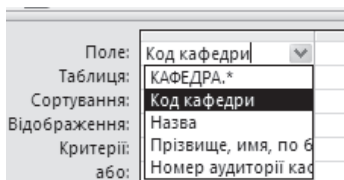


Рис.6.5.24. Приклад введення ім'я поля

Розкриємо другий список Поле й виберемо значення КАФЕДРА.прізвище, ім'я, по-батькові зав_кафедрою, а також установимо прапорець для відображення значення цього поля на екрані згідно із запитом. Аналогічно, укажемо в списку Поле ВИКЛАДАЧ.ПІБ викладача, ВИКЛАДАЧ.Посаду, РОЗКЛАД.Навчальна дисципліна, РОЗКЛАД.Дата проведення (рисунок 6.5.25).

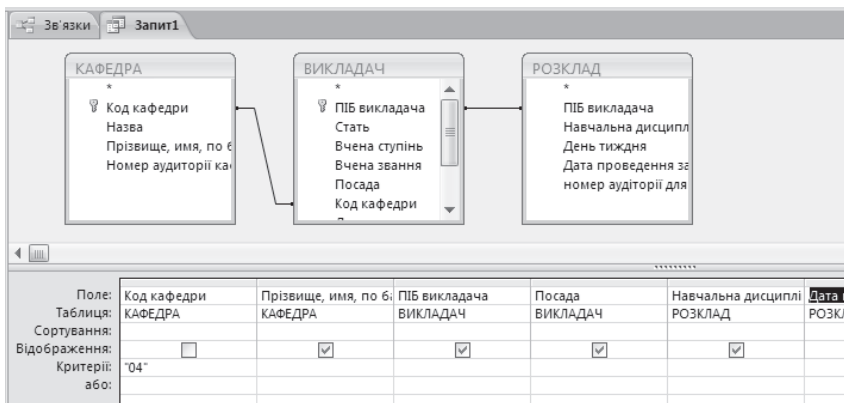


Рис.6.5.25. Приклад заповнення бланка запиту в Конструкторі

Для перегляду результату виконання запиту слід на вкладці *Конструктор* перейти в групу інструментів *Результати* і натиснути на



кнопку *Запуск* (рисунок 6.5.26).

Прізвище, імя, по ф	ПІБ викладача	Посада	Навчальна дисципліна	Дата проведення
Іваненко Ганна Віктор	Дмитренко Галіна Василів	доцент	економетрія, 12:50-14:10	16.02.2009
Іваненко Ганна Віктор	Дмитренко Галіна Василів	доцент	економіко-математичне	17.02.2009

Рис.6.5.26. Результати виконання запиту на вибірку за допомогою Конструктора

Питання і завдання для самостійної роботи

Виконайте проектування і реалізацію в Microsoft Office Access 2007 наступних баз даних. На підставі мінімального набору атрибутів (який, при необхідності, може бути розширений) створіть таблиці-об'єкти і обґрунтуйте види зв'язків між ними. Придумайте і реалізуйте запити на вибірку за допомогою Майстра запитів і Конструктора запитів.

1. БД «Мережа магазинів». Мінімальний набір атрибутів: назва магазину, ПІБ директора магазину, юридична адреса магазину, тип магазину (продовольчий, господарський і тому подібне), назва підприємства-постачальника, юридична адреса підприємства-постачальника, тип товарів на складі підприємства-постачальника.

2. БД «Бібліотека». Мінімальний набір атрибутів: номер читацького квитка, ПІБ читача, назва книги, автор книги, рік видання книги, дата, коли читач узяв і повернув книгу.

3. БД «Відділ кадрів». Мінімальний набір атрибутів: ПІБ співробітника, посада, оклад.

4. БД «Студенти економічного факультету». Мінімальний набір атрибутів: ПІБ студента, рік вступу до ВУЗУ, середній бал.

5. БД «Автомайстерня». Мінімальний набір атрибутів: марка автомобіля, номерний знак автомобіля, характер несправності, ПІБ механіка, що здійснює ремонт.

Список літератури

1. Тихомиров А.Н., Прокди А.К., Колосков П.В., Клеандрова И.А. и др. Microsoft Office 2007. Все программы пакета: Word, Excel, Access, PowerPoint, Publisher, Outlook, OneNote, InfoPath, Groove. Самоучитель. – Спб.: Наука и Техника, 2008. – 608 с.
2. Microsoft Access 2003. Шаг за шагом/Практ.пособ./Пер. с англ. – М.: «СП ЭКОМ», 2004. – 432 с.
3. Андерсен В. Базы данных Microsoft Access. Проблемы и решения: Практ. пособ. / Пер. с англ. – М.: Издательство ЭКОМ, 2001. – 384 с.

Глава VII. КОМП'ЮТЕРНІ МЕРЕЖІ І ТЕЛЕКОМУНІКАЦІЇ

Мета даного розділу - дати основні уявлення про побудову, організацію і використання комп'ютерних мереж.

7.1. Принципи організації комп'ютерних мереж

➤ **Комп'ютерна мережа** – це сукупність комп'ютерів і різних пристроїв, що забезпечують інформаційний обмін між комп'ютерами в мережі без використання яких-небудь проміжних носіїв інформації.

Все різноманіття комп'ютерних мереж можна класифікувати по групі ознак:

- 1) територіальна поширеність;
- 2) відомча приналежність;
- 3) швидкість передачі інформації;
- 4) тип середовища передачі;

По територіальній поширеності мережі можуть бути *локальними*, *глобальними* і *регіональними*. *Локальні* – це мережі, що перекривають територію не більше 10 м², *регіональні* – розташовані на території міста або області, *глобальні* – на території держави або групи держав, наприклад, усесвітня мережа Internet.

За приналежністю розрізняють *відомчі* і *державні мережі*. *Відомчі* належать одній організації і розташовуються на її території. *Державні мережі* – мережі, використовувані в державних структурах.

За швидкістю передачі інформації комп'ютерні мережі діляться на *низько-*, *середньо-* і *високошвидкісні*.

За типом середовища передачі розділяються на мережі *коаксіальні*, *на витій парі*, *оптоволоконні*, *з передачею інформації по радіоканалах*, *в інфрачервоному діапазоні*.

Комп'ютери можуть з'єднуватися кабелями, утворюючи різну топологію мережі (зоряна, шинна, кільцева та ін.).

Слід розрізнити *комп'ютерні мережі* і *мережі терміналів* (термінальні мережі). *Комп'ютерні мережі* зв'язують комп'ютери, кожен з яких може працювати і автономно. *Термінальні мережі* зазвичай зв'язують потужні комп'ютери (майнфрейми), а в окремих випадках і ПК з пристроями (терміналами), які можуть бути достатньо складні, але поза мережею їх робота або неможлива, або взагалі втрачає сенс. Наприклад, мережа

банкоматів або кас з продажу авіаквитків. Будуються вони на абсолютно інших, ніж комп'ютерні мережі, принципах і навіть на іншій обчислювальній техніці.

У класифікації мереж існує два основні терміни: *LAN* і *WAN*.

➤ *LAN* (Local Area Network) – локальні мережі, що мають замкнену інфраструктуру до виходу на послуги постачальників. Термін «LAN» може обслуговувати і маленьку офісну мережу, і мережу рівня великого заводу, що займає декілька сотен гектарів. Зарубіжні джерела дають навіть наближену оцінку – приблизно шести миль (10 км) в радіусі; використання високошвидкісних каналів.

➤ *WAN* (Wide Area Network) – глобальна мережа, що покриває великі географічні регіони, що включають як локальні мережі, так і інші телекомунікаційні мережі і пристрої. Приклад WAN – мережі з комутацією пакетів (Frame Relay), через яку можуть «розмовляти» між собою різні комп'ютерні мережі.

➤ Термін «корпоративна мережа» також використовується в літературі для позначення об'єднання декількох мереж, кожна з яких може бути побудована на різних технічних, програмних і інформаційних принципах.

Види мереж, розглянуті вище, є мережами закритого типу, доступ до них дозволений тільки обмеженому колу користувачів, для яких робота в такій мережі безпосередньо пов'язана з їх професійною діяльністю. Глобальні мережі орієнтовані на обслуговування будь-яких користувачів.

➤ Під *локальною обчислювальною мережею* (ЛОМ) розуміють спільне підключення декількох окремих комп'ютерних робочих місць (робочих станцій) до єдиного каналу передачі даних.

Завдяки обчислювальним мережам ми дістали можливість одночасного використання програм і баз даних декількома користувачами.

Поняття локальна обчислювальна мережа належить до географічно обмежених (територіально або виробничо) апаратно-програмних реалізацій, в яких декілька комп'ютерних систем пов'язані одна з однією за допомогою відповідних засобів комунікацій. Завдяки такому з'єднанню користувач може взаємодіяти з іншими робочими станціями, підключеними до цієї ЛОМ.

У виробничій практиці ЛОМ грають дуже велику роль. За допомогою ЛОМ в систему об'єднуються персональні комп'ютери, розташовані на багатьох видалених робочих місцях, які використовують спільне обладнання, програмні засоби і інформацію. Робочі місця співробітників перестають бути ізольованими і об'єднуються в єдину систему. Розглянемо переваги, що отримуються при мережевому об'єднанні персональних комп'ютерів у вигляді внутрішньовиробничої обчислювальної мережі.

Розділення ресурсів. Розділення ресурсів дозволяє економно використовувати ресурси, наприклад, управляти периферійними пристроями, такими як лазерні друкуючі пристрої, зі всіх приєднаних робочих станцій.

Розділення даних. Розділення даних надає можливість доступу і управління базами даних з периферійних робочих місць, що потребують інформації.

Розділення програмних засобів. Розділення програмних засобів надає можливість одночасного використання централізованих, раніше встановлених програмних засобів.

Розділення ресурсів процесора. При розділенні ресурсів процесора можливе використання обчислювальних потужностей для обробки даних іншими системами, що входять в мережу. Можливість, що надається, полягає в тому, що на наявні ресурси не «накидаються» вміть, а тільки через спеціальний процесор, доступний кожній робочій станції.

Багатокористувацький режим. Багатокористувацькі властивості системи сприяють одночасному використанню централізованих прикладних програмних засобів, раніше встановлених і керованих, наприклад, якщо користувач системи працює з іншим завданням, то поточна виконувана робота відсовується на задній план.

Локальні обчислювальні мережі підрозділяються на два класи, що кардинально розрізняються: *однорангові* (однорівневі або Peer to Peer) *мережі* і *ієрархічні* (багаторівневі).

➤ **ОДНОРАНГОВІ МЕРЕЖІ.** *Однорангова мережа* є мережею рівноправних комп'ютерів, кожен з яких має унікальне ім'я (ім'я комп'ютера) і зазвичай пароль для входу в нього під час завантаження ОС. Ім'я і пароль входу призначаються власником ПК засобами ОС. Однорангові мережі можуть бути організовані за допомогою таких операційних систем, як LANtastic, Windows'3.11, Novell NetWare Lite. Вказані програми працюють як з DOS, так і з Windows. Однорангові мережі можуть бути організовані також на базі всіх сучасних 32-розрядних операційних систем – Windows'95 OSR2, Windows NT Workstation версії, OS/2) і деяких інших.

➤ **ІЄРАРХІЧНІ МЕРЕЖІ.** В *ієрархічних локальних мережах* є один або декілька спеціальних комп'ютерів – серверів, на яких зберігається інформація, спільно використовувана різними користувачами.

Сервер в ієрархічних мережах – це постійне сховище ресурсів, що розділяються. Сам сервер може бути клієнтом тільки сервера більш високого рівня ієрархії. Тому ієрархічні мережі інколи називаються мережами з виділеним сервером. Сервери зазвичай є високопродуктивними комп'ютерами, можливо, з декількома паралельно працюючими процесорами, з вінчестерами великої ємності, з високошвидкісною мережевою картою (100

Мбіт/с і більш). Комп'ютери, в яких здійснюється доступ до інформації на сервері, називаються *станціями* або *клієнтами*.

ЛОМ класифікуються за призначенням.

- *Мережі термінального обслуговування*. У них включається ЕОМ і периферійне обладнання, використовуване в монопольному режимі комп'ютером, до якого воно підключається, або бути загальномережевим ресурсом.

- Мережі, на базі яких побудовані системи управління виробництвом. Вони об'єднуються групою стандартів МАРІ/ТОРИ. У МАРІ описуються стандарти, використовувані в промисловості. ТОР описують стандарти для мереж, вживаних в офісних мережах.

- Мережі, які об'єднують системи автоматизації, проектування. Робочі станції таких мереж зазвичай базуються на достатньо потужних персональних ЕОМ, наприклад фірми Sun Microsystems.

- Мережі, на базі яких побудовані розподілені обчислювальні системи.

Всі ЛОМ працюють в одному стандарті, прийнятому для комп'ютерних мереж - в стандарті Open Systems Interconnection (OSI) організації і використання комп'ютерних мереж.

7.1.1. Технологія Клієнт-Сервер

Характер взаємодії комп'ютерів в локальній мережі прийнято пов'язувати з їх функціональним призначенням. Як і в разі прямого з'єднання, в рамках локальних мереж використовується поняття клієнт і сервер.

➤ *Технологія клієнт-сервер* — це особливий спосіб взаємодії комп'ютерів в локальній мережі, при якому один з комп'ютерів (сервер) надає свої ресурси іншому комп'ютеру (клієнтові). Відповідно до цього розрізняють *однорангові мережі* і *серверні мережі*.

➤ При *одноранговій архітектурі* в мережі відсутні виділені сервери, кожна робоча станція може виконувати функції клієнта і сервера. В цьому випадку робоча станція виділяє частину своїх ресурсів в спільне користування всім робочим станціям мережі. Як правило, однорангові мережі створюються на базі однакових по потужності комп'ютерів. Однорангові мережі є достатньо простими в налазді і експлуатації. У тому випадку, коли мережа складається з невеликого числа комп'ютерів і її основною функцією є обмін інформацією між робочими станціями, однорангова архітектура є найбільш прийнятним рішенням. Подібна мережа може бути достатньо швидко і просто реалізована засобами такої популярної операційної системи як Windows XP.

Наявність розподілених даних і можливість зміни своїх серверних ресурсів кожною робочою станцією ускладнює захист інформації від несанкціонованого доступу, що є одним з недоліків однорангових мереж. Розуміючи це, розробники починають приділяти особливу увагу питанням захисту інформації в однорангових мережах.

Іншим недоліком однорангових мереж є їх низька продуктивність. Це пояснюється тим, що мережеві ресурси зосереджені на робочих станціях, яким доводиться одночасно виконувати функції клієнтів і серверів.

➤ У *серверних мережах* здійснюється чітке розділення функцій між комп'ютерами: одні з них постійно є клієнтами, а інші — серверами. Враховуючи різноманіття послуг, що надаються комп'ютерними мережами, існує декілька типів серверів, а саме: *мережевий сервер, файловий сервер, сервер друку, поштовий сервер* та ін.

➤ *Мережевим сервером* є спеціалізований комп'ютер, орієнтований на виконання основного об'єму обчислювальних робіт і функцій по управлінню комп'ютерною мережею. Цей сервер містить ядро мережевої операційної системи, під управлінням якої здійснюється робота всієї локальної мережі. Мережевий сервер володіє достатньо високою швидкістю і великим об'ємом пам'яті. При подібній мережевій організації функції робочих станцій зводяться до введення-виводу інформації і обміну нею з мережевим сервером.

➤ Термін *файловий сервер* стосується комп'ютера, основною функцією якого є зберігання, управління і передача файлів даних. Він не обробляє і не змінює файли, що зберігаються і передаються ним. Сервер може «не знати», чи є файл текстовим документом, графічним зображенням або електронною таблицею. У спільному випадку на файловому сервері може бути навіть відсутньою клавіатура і монітор. Всі зміни у файлах даних здійснюються з клієнтських робочих станцій. Для цього клієнти прочитують файли даних з файлового сервера, здійснюють необхідні зміни даних і повертають їх назад на файловий сервер. Подібна організація найбільш ефективна при роботі великої кількості користувачів із спільною базою даних. В рамках великих мереж може одночасно використовуватися декілька файлових серверів.

➤ *Сервером друку* (принт-сервер) є друкуючий пристрій, який за допомогою мережевого адаптера підключається до передаючого середовища. Подібний мережевий друкуючий пристрій є самостійним і працює незалежно від інших мережевих пристроїв. Сервер друку обслуговує заявки на друк від всіх серверів і робочих станцій. Як сервери друку використовуються спеціальні високопродуктивні принтери.

При високій інтенсивності обміну даними з глобальними мережами в рамках локальних мереж виділяються поштові сервери, за допомогою яких обробляються повідомлення електронної пошти.

БАЗОВА ЕТАЛОННА МОДЕЛЬ ВЗАЄМОДІЇ ВІДКРИТИХ СИСТЕМ OSI (OPEN SYSTEM INTERCONNECTION)

Для того щоб взаємодіяти, люди використовують спільну мову. Якщо вони не можуть розмовляти один з одним безпосередньо, вони застосовують відповідні допоміжні засоби для передачі повідомлень.

Показані вище стадії необхідні, коли повідомлення передається від відправника до одержувача.

Для того, щоб привести в рух процес передачі даних, використовували машини з однаковим кодуванням даних і зв'язаних одна з іншою. Для єдиного представлення даних в лініях зв'язку, по яких передається інформація, сформована Міжнародна організація по стандартизації (англ. ISO - International Standards Organization).

ISO призначена для розробки моделі міжнародного комунікаційного протоколу, в рамках якої можна розробляти міжнародні стандарти. Для наочного пояснення розчленуємо її на сім рівнів.

Міжнародна організація по стандартизації (ISO) розробила базову модель взаємодії відкритих систем (англ. Open System Incorporation (OSI)). Ця модель є міжнародним стандартом для передачі даних і містить сім окремих рівнів.

Рівень 1: фізичний - бітові протоколи передачі інформації;

Рівень 2: каналний - формування кадрів, управління доступом до середовища;

Рівень 3: мережевий - маршрутизація, управління потоками даних;

Рівень 4: транспортний - забезпечення взаємодії видалених процесів;

Рівень 5: сеансовий - підтримка діалогу між видаленими процесами;

Рівень 6: представлення даних - інтерпретація передаваних даних;

Рівень 7: прикладний - призначений для управління даними користувачем.

Комунікації: забезпечення точної доставки даних між кінцевими станціями.	7. Прикладний рівень	Програмне забезпечення
	6. Показний рівень	
	5. Сеансовий рівень	
	4. Транспортний рівень	
З'єднання: управління фізичною доставкою даних по мережі.	3. Мережевий рівень	Маршрутизатори, комутатори рівня 3
	2. Канальний рівень	Мости, комутатори
	1. Фізичний рівень	Кабелі, повторювачі, концентратори, модеми

Основна ідея цієї моделі полягає в тому, що кожному рівню відводиться конкретна роль у тому числі і транспортному середовищу. Завдяки цьому спільне завдання передачі даних розчленовується на окремі легко осяжні завдання. Необхідні угоди для зв'язку одного рівня з вище- і нижче розташованими називають *протоколом*.

Оскільки користувачі потребують ефективного управління, система обчислювальної мережі представляється як комплексна будова, яка координує взаємодію завдань користувачів.

З урахуванням вищевикладеного можна вивести наступну рівневу модель з адміністративними функціями, що виконуються в призначеному для користувача прикладному рівні.

Окремі рівні базової моделі проходять в напрямі вниз від джерела даних (від рівня 7 до рівня 1) і в напрямі вгору від приймача даних (від рівня 1 до рівня 7). Призначені для користувача дані передаються в розташований нижче рівень разом із специфічним для рівня заголовком до тих пір, поки не буде досягнутий останній рівень.

На приймальній стороні дані, що поступають, аналізуються і, у міру потреби, передаються далі у вище розташований рівень, поки інформація не буде передана в призначений для користувача прикладний рівень.

Рівень 1 – фізичний. На фізичному рівні визначаються електричні, механічні, функціональні і процедурні параметри для фізичного зв'язку в системах. Фізичний зв'язок і нерозривна з нею експлуатаційна готовність є основною функцією 1-го рівня. Стандарти фізичного рівня включають рекомендації V.24 МККТТ (CCITT), EIA RS232 і X.21. Стандарт ISDN (Integrated Services Digital Network) в майбутньому зіграє визначальну роль для функцій передачі даних. Як середовище передачі даних використовують багатожильний мідний дріт (екранована вита пара), коаксіальний кабель, оптоволоконний провідник і радіорелейну лінію.

Рівень 2 - канальний. Канальний рівень формує з даних, що передаються 1-м рівнем, так звані «кадри», послідовності кадрів. На цьому рівні здійснюються управління доступом до передаючого середовища, використовуваного декількома ЕОМ, синхронізація, виявлення і виправлення помилок.

Рівень 3 - мережевий. Мережевий рівень встановлює зв'язок в обчислювальній мережі між двома абонентами. З'єднання відбувається завдяки функціям маршрутизації, які вимагають наявності мережевої адреси в пакеті. Мережевий рівень повинен також забезпечувати обробку помилок, управління потоками даних. Найвідоміший стандарт, що належить до цього рівня, - рекомендація X.25 МККТТ (для мереж спільного користування з комутацією пакетів).

Рівень 4 – *транспортний*. Транспортний рівень підтримує безперервну передачу даних між двома призначеними для користувача процесами, що взаємодіють один з одним. Якість транспортування, безпомилковість передачі, незалежність обчислювальних мереж, сервіс транспортування з кінця в кінець, мінімізація витрат і адресація зв'язку гарантують безперервну і безпомилкову передачу даних.

Рівень 5 – *сеансовий*. Сеансовий рівень координує прийом, передачу і видачу одного сеансу зв'язку. Для координації необхідні контроль робочих параметрів, управління потоками даних проміжних накопичувачів і діалоговий контроль, що гарантує передачу, що є у розпорядженні даних. Крім того, сеансовий рівень містить додатково функції управління паролями, підрахунку плати за користування ресурсами мережі, управління діалогом, синхронізації і відміни зв'язку в сеансі передачі після збою унаслідок помилок в розташованих нижче рівнях.

Рівень 6 – *представлення даних*. Рівень представлення даних призначений для інтерпретації даних; а також підготовки даних для призначеного для користувача прикладного рівня. На цьому рівні відбувається перетворення даних з кадрів, використовуваних для передачі даних в екранний формат або формат для друкуючих пристроїв крайової системи.

Рівень 7 – *прикладний*. У прикладному рівні необхідно надати в розпорядження користувачів вже перероблену інформацію. З цим може справитися системне і призначене для користувача прикладне програмне забезпечення.

Для передачі інформації по комунікаційним лініям дані перетворюються в ланцюжок наступних один за одним бітів (двійкове кодування за допомогою двох станів: «0» і «1»).

Передавані алфавітно-цифрові знаки представляються за допомогою бітових комбінацій. Бітові комбінації розташовують в певній кодовій таблиці, що містить 4-, 5-, 6-, 7- або 8-бітові коди.

Кількість представлених знаків в ході залежить від кількості бітів, використовуваних в коді: код з чотирьох бітів може представити максимум 16 значень, 5-бітовий код - 32 значення, 6-бітовий код - 64 значення, 7-бітовий - 128 значень і 8-бітовий код - 256 алфавітно-цифрових знаків.

При передачі інформації між однаковими обчислювальними системами і типами комп'ютерів, що розрізняються, застосовують наступні коди.

На міжнародному рівні передача символної інформації здійснюється за допомогою 7-бітового кодування, що дозволяє закодувати заголовні і рядкові букви англійського алфавіту, а також деякі спецсимволи.

Національні і спеціальні знаки за допомогою 7-бітового коду представити не можна. Для представлення національних знаків застосовують найбільш вживаний 8-бітовий код.

Для правильної і, отже, повної і безпомилкової передачі даних необхідно дотримуватися узгоджених і встановлених правил. Всі вони обумовлені в протоколі передачі даних.

Протокол передачі даних вимагає наступної інформації.

- Синхронізація. Під синхронізацією розуміють механізм розпізнавання початку блоку даних і його кінця.

- Ініціалізація. Під ініціалізацією розуміють встановлення з'єднання між взаємодіючими партнерами.

- Блокування. Під блокуванням розуміють розбиття передаваної інформації на блоки даних строго певної максимальної довжини (включаючи пізнавальні знаки початку блоку і його кінця).

- Адресація. Адресація забезпечує ідентифікацію різного використовуваного устаткування даних, які обмінюються один з одним інформацією під час взаємодії.

- Виявлення помилок. Під виявленням помилок розуміють установку бітів парності і, отже, обчислення контрольних бітів.

- Нумерація блоків. Поточна нумерація блоків дозволяє встановити помилково передану або таку інформацію, що загубилася.

- Управління потоком даних. Управління потоком даних служить для розподілу і синхронізації інформаційних потоків. Так, наприклад, якщо не хватає місця в буфері пристрою даних або дані не достатньо швидко обробляються в периферійних пристроях (наприклад, принтерах), повідомлення і / або запити накопичуються.

- Методи відновлення. Після переривання процесу передачі даних використовують методи відновлення, щоб повернутися до певного положення для повторної передачі інформації.

- Дозвіл доступу. Розподіл, контроль і управління обмеженнями доступу до даних ставляться в обов'язок пункту дозволу доступу (наприклад, «тільки передача» або «тільки прийом»).

7.1.2. Сітові пристрої і засоби комунікацій

Як засоби комунікації найчастіше використовуються вита пара, коаксіальний кабель, оптоволоконні лінії. При виборі типа кабелю враховують наступні показники:

- вартість монтажу і обслуговування;
- швидкість передачі інформації;
- обмеження на величину відстані передачі інформації (без додаткових підсилювачів-повторювачів (репітерів);
- безпека передачі даних.

Головна проблема полягає в одночасному забезпеченні цих показників, наприклад, найвища швидкість передачі даних обмежена максимально можливою відстанню передачі даних, при якому ще забезпечується необхідний рівень захисту даних. Легкі нарощуваність і простота розширення кабельної системи впливають на її вартість.

Вита пара. Найбільш доступним кабельним з'єднанням є вите двожильне дротяне з'єднання, часто зване «витою парою» (twisted pair). Вона дозволяє передавати інформацію з швидкістю до 10 Мбіт/с, легко нарощується, проте є перешкодоне захищеною. Довжина кабелю не може перевищувати 1000 м при швидкості передачі 1 Мбіт/с. Перевагами є низька ціна і нескладна установка. Для підвищення перешкодозахисної інформації часто використовують екрановану виту пару, тобто виту пару, поміщену в екрануючу оболонку, подібно до екрану коаксіального кабелю. Це збільшує вартість витої пари і наближає її ціну до ціни коаксіального кабелю.

Коаксіальний кабель. Коаксіальний кабель добре перешкодозахищений і застосовується для зв'язку на великі відстані (декілька кілометрів). Швидкість передачі інформації від 1 до 10 Мбіт/с, а в деяких випадках може досягати 50 Мбіт/с. Коаксіальний кабель використовується для основної і ширококутної передачі інформації і має середню ціну.

Оптоволоконні лінії. Найбільш якісним середовищем передачі даних є оптопровідники, звані також скловолоконним кабелем. Швидкість розповсюдження інформації по ним досягає декілька гігабіт в секунду. Допустиме видалення більше 50 км. Зовнішня дія перешкод практично відсутня. На даний момент це найбільш дороге з'єднання для ЛОМ. Застосовується там, де виникають електромагнітні поля перешкод або потрібна передача інформації на дуже великі відстані без використання повторювачів. Вони володіють протипідслуховуваними властивостями, оскільки техніка відгалужень в оптоволоконних кабелях дуже складна. Оптопровідники об'єднуються в ЛОМ за допомогою зіркоподібного з'єднання.

Показники трьох типів середовищ для передачі наведені в таблиці.

Показники	Середовище передачі даних		
	Двожильний кабель - вита	Коаксіальний кабель	Оптоволоконний кабель

	пара		
Ціна	Невисока	Відносно висока	Висока
Нарощування	Дуже просте	Проблематичне	Просте
Захист від прослуховування	Незначний	Хороший	Високий
Проблеми із заземленням	Немає	Можливі	Немає
Сприйнятливість до перешкод	Існує	Існує	Відсутня

КОМУНІКАЦІЙНІ ПРОТОКОЛИ ЛОКАЛЬНИХ МЕРЕЖ. Формалізовані правила, визначальна послідовність і формат повідомлень, якими обмінюються мережеві компоненти, що знаходяться на одному рівні, але в різних вузлах, називаються *протоколом*.

Під *протоколами локальних мереж* мається на увазі набір протоколів першого і другого рівнів базової еталонної моделі (OSI), що визначають архітектуру локальної мережі, у тому числі її топологію, що передає середовище, технічні засоби і протоколи. Основоположними для локальних мереж є стандарти серії IEEE. За допомогою цих стандартів були визначені: основна термінологія, архітектура і протоколи двох нижніх рівнів Еталонної моделі взаємодії відкритих систем.

Структура стандарту IEEE. Стандарт IEEE 802.1 є спільним документом, який визначає архітектуру і прикладні процеси системного управління мережею, методи об'єднання мереж на підрівні управління доступом до середовища, що передає. Відповідно до даного стандарту каналний рівень розбитий на два підрівні: УЛК — управління логічним каналом і УДС — управління доступом до фізичного середовища.

Стандарт IEEE 802.2 визначає протоколи управління логічним каналом, у тому числі специфікує інтерфейси з мережевим рівнем і підрівнем управління доступом до передаючого середовища. Кожен з решти стандартів, починаючи з IEEE 802.3, визначає метод доступу і специфіку фізичного рівня для конкретного типу локальної комп'ютерної мережі. Так, стандарт IEEE 802.3 описує характеристики і процедури множинного доступу з контролем передачі і виявленням зіткнень. Стандарт IEEE 802.4 визначає протокол маркерного доступу до моноканалу. Процедури і характеристики маркерного методу доступу до кільцевої мережі визначається стандартом IEEE 802.5. Для локальних мереж, що охоплюють площу радіусом до 25 км. і використовують технічні засоби кабельного телебачення, розроблений стандарт IEEE 802.6. Цей стандарт передбачає передачу даних, мови, зображень і дозволяє створювати так звані міські локальні мережі. В даний час продовжуються роботи по стандартизації локальних комп'ютерних мереж. Так, в підкомітеті

IEEE 802.11 розробляється стандарт на радіомережі для мобільних комп'ютерів, а в комітеті IEEE 802.12 розглядується стандарт на високошвидкісні комп'ютерні мережі "100VG-AnyLAN.

У 1985 році серія стандартів IEEE 802 була прийнята Міжнародною організацією стандартів за основу міжнародних стандартів фізичного і каналного рівнів ISO/DIS 8802/2.2 — ISO/DIS 8802/5. Крім того, ці стандарти були доповнені стандартом ISO/DIS 8802/7 на мережі з тактованим методом доступу до кільця, розробленим на основі протоколів мережі Cambridge Ring.

Транспортні протоколи. Транспортні протоколи призначені для забезпечення надійного зв'язку в процесі обміну інформацією між абонентами комп'ютерної мережі. Як відомо, якість передачі інформації багато в чому визначається використовуваною лінією зв'язку. Наприклад, комутовані телефонні канали мереж спільного користування характеризуються відносно високим рівнем перешкод. При використанні подібних каналів в комп'ютерних мережах необхідно приймати додаткові заходи з підвищення надійності передачі даних. У свою чергу, оптоволоконні лінії зв'язку характеризуються низьким рівнем перешкод. В даному випадку досить використовувати мінімальний набір транспортних послуг і простий протокол обміну інформацією. Особливе значення транспортні протоколи набувають в комп'ютерних мережах, передаюче середовище яких характеризується відносно високим рівнем помилок і низькою надійністю передачі даних.

Одним з перших протоколів транспортного рівня є протокол АННР (ARPA Host-to-Host Protocol), розроблений для мережі ARPA. Основна увага в протоколі АННР приділялася управлінню потоком даних, адресації користувачів, а також взаємодії з програмами, що реалізують протоколи верхніх рівнів. Розвиток мережі ARPA у напрямі використання мереж передачі даних спільного користування привів до появи нового, надійнішого протоколу, відомого в даний час під назвою «*протокол управління передачею*» або *TCP* (скорочення від Transmission Control Protocol). Протокол TCP виявився достатньо вдалим і був покладений в основу стандартного міжнародного протоколу транспортного рівня. Відповідно, МККТТ визначив рекомендацію X.224 для даного транспортного протоколу, а також рекомендацію X.214 для транспортної служби.

З метою вибору оптимального набору транспортних послуг стандартним протоколом визначено три типи (А, В, С) мережевих з'єднань і п'ять класів (0, 1, 2, 3, 4) транспортного протоколу. Залежно від характеристик конкретної мережі передачі даних визначається тип мережевого з'єднання, якому вона задовольняє. Потім, з урахуванням

необхідного рівня якості передачі, вибирається необхідний клас транспортного протоколу.

СТАНДАРТНІ СТЕК КОМУНІКАЦІЙНИХ ПРОТОКОЛІВ. Ієрархічно організований набір протоколів, достатній для організації взаємодії вузлів в мережі, називається *стеком комунікаційних протоколів*.

Найважливішим напрямом стандартизації в області обчислювальних мереж є стандартизація комунікаційних протоколів. В даний час в мережах використовується велика кількість стеків комунікаційних протоколів. Найбільш популярними є стеки: TCP/IP, IPX/SPX, OSI. Всі ці стеки на нижніх рівнях — фізичному і каналному, — використовують одні і ті ж добре стандартизовані протоколи, наприклад Ethernet, які дозволяють використовувати у всіх мережах одну і ту ж апаратуру. Зате на верхніх рівнях всі стеки працюють за своїми власними протоколами. Ці протоколи часто не відповідають тому, що рекомендується моделлю OSI розбиттю на рівні. Зокрема, *функції* сеансового і показного рівня, як правило, об'єднані з прикладним рівнем. Така невідповідність пов'язана з тим, що модель OSI з'явилася як результат узагальнення тих, що вже існують і реально використовуваних стеків, а не *навпаки*.

Стек TCP/IP. Стек TCP/IP був розроблений за ініціативою Міністерства оборони США більше 20 років тому для зв'язку експериментальної мережі ARPAnet з іншими мережами як набір спільних протоколів для різноманітного обчислювального середовища. Великий вклад в розвиток стека TCP/IP, який отримав свою назву від популярних протоколів IP і TCP, вніс університет Берклі, реалізувавши протоколи стека в своїй версії ОС UNIX. Популярність цієї операційної системи привела до широкого розповсюдження протоколів TCP, IP і інших протоколів стека. Сьогодні цей стек використовується для зв'язку комп'ютерів усесвітньої інформаційної мережі Internet, а також у величезному числі корпоративних мереж.

Стек TCP/IP на нижньому рівні підтримує всі популярні стандарти фізичного і каналного рівнів: для локальних мереж — це Ethernet, Token Ring, FDDI, для глобальних — протоколи роботи на аналогових комутуваних і виділених лініях SLIP, PPP, протоколи територіальних мереж X.25 і ISDN.

Основними протоколами стека, що дали йому назву, є протоколи IP і TCP. Ці протоколи в термінології моделі OSI належать до мережевого і транспортного рівнів відповідно. IP забезпечує просування пакету по складеній мережі, а TCP гарантує надійність його доставки.

За довгі роки використання в мережах різних країн і організацій стек TCP/IP увібрав в себе велику кількість протоколів прикладного рівня. До них

належать такі популярні протоколи, як протокол пересилки файлів FTP, протокол емуляції терміналу telnet, поштовий протокол SMTP, використовуваний в електронній пошті мережі Internet, гіпертекстові сервіси служби WWW і багато інших. Сьогодні стек TCP/IP є одним з найпоширеніших стеків транспортних протоколів обчислювальних мереж. Дійсно, тільки в мережі Internet об'єднано близько 10 мільйонів комп'ютерів по всьому світу, які взаємодіють один з одним за допомогою стека протоколів TCP/IP.

Стрімке зростання популярності Internet привело і до змін в розставлянні сил в світі комунікаційних протоколів — протоколи TCP/IP, на яких побудований Internet, стали швидко тіснити протоколи, що широко використовувалися раніше. Сьогодні в світі переважна більшість комп'ютерів підключаються до мережі, використовуючи стек TCP/IP. Процес становлення стека TCP/IP як стек номер один в будь-яких типах мереж продовжується, і зараз будь-яка промислова операційна система обов'язково включає програмну реалізацію цього стека в своєму комплекті постачання.

Хоча протоколи TCP/IP нерозривно пов'язані з Internet і кожен з багатомільйонної армії комп'ютерів Internet працює на основі цього стека, існує велика кількість локальних, корпоративних і територіальних мереж, що безпосередньо не є частинами Internet, але також використовують протоколи TCP/IP. Щоб відрізнити їх від Internet, ці мережі називають мережами TCP/IP або просто IP-мережами.

Оскільки стек TCP/IP спочатку створювався для глобальної мережі Internet, він має багато особливостей, що дають йому перевагу перед іншими протоколами, коли мова заходить про побудову мереж, що включають глобальні зв'язки. Зокрема, дуже корисною властивістю, що робить можливим застосування цього протоколу у великих мережах, є його здатність фрагментувати пакети. Дійсно, велика складена мережа часто складається з мереж, побудованих на абсолютно різних принципах. У кожній з цих мереж може бути встановлена власна величина максимальної довжини одиниці передаваних даних (кадру). У такому разі при переході з однієї мережі, що має велику максимальну довжину, в мережу з меншою максимальною довжиною може виникнути необхідність ділення передаваного кадру на декілька частин. Протокол IP стека TCP/IP ефективно вирішує цю задачу.

Іншою особливістю технології TCP/IP є гнучка система адресації, що дозволяє простіше в порівнянні з іншими протоколами аналогічного призначення включати в інтермережу мережі інших технологій. Ця властивість також сприяє застосуванню стека TCP/IP для побудови великих гетерогенних мереж.

У стеку TCP/IP дуже економно використовуються можливості ширококомовних (одночасно для декількох користувачів) розсилок. Ця властивість абсолютно необхідна при роботі на повільних каналах зв'язку, характерних для територіальних мереж. Проте, використання технології TCP/IP пред'являє високі вимоги до ресурсів і адміністрування IP мереж. Потужні функціональні можливості протоколів стека TCP/IP вимагають для своєї реалізації високих обчислювальних витрат. Гнучка система адресації і відмова від ширококомовних розсилок приводять до наявності в IP-мережі різних централізованих служб типа DNS, DHCP і тому подібне. Кожна з цих служб має напрям на полегшення адміністрування мережі, у тому числі і на полегшення конфігурації устаткування, але в той же час сама вимагає пильної уваги з боку адміністраторів.

У корпоративній мережі важливу роль грають механізми перетворення фізичних адрес конкретної мережі в міжмережеві (Internet) адреси і навпаки. В рамках кожної окремої мережі робочі станції взаємодіють між собою на каналному рівні, використовуючи для цього свою систему адресації. Так, фізична адреса в мережі Ethernet задається шестибайтовим числовим значенням, кожен байт записується в шістнадцятиричній системі і відділяється двокрапкою, наприклад: 07:01:AO:47:54:C4.

Для забезпечення умови «відвертості» систем міжмережеві адреси, звані *IP-адресами*, є логічними і не залежать від апаратури або конфігурації мережі. IP-адрес складається з чотирьох десяткових цифр (кожен по величині не більше 255), відокремлених один від одного крапками, наприклад 10.18.57.10. Крайнє зліва число позначає базову мережу, подальші числа вказують на дрібніші ділянки усередині цієї мережі — до адреси конкретного комп'ютера. Для полегшення запам'ятовування адрес широко використовується їх іменне позначення, зване *доменним*. Перетворення домена в цифрову адресу здійснюється автоматично при маршрутизації повідомлення. Доменні імена володіють постійною структурою, спираючись на яку, можна зрозуміти, до чого вони належать. *Система доменних імен* (DNS), що описує комп'ютери і організації, в яких вони встановлені, влаштована дзеркально по відношенню до цифрової IP-адресації. Якщо в IP-адресі найбільш спільна інформація вказана ліворуч, то в доменних іменах вона знаходиться праворуч. IP-пакет поміщається у фізичний кадр тієї мережі, по якій він зараз передається. IP-пакет містить міжмережеву адресу вузла-одержувача, мережевий кадр даних, у свою чергу, повинен містити фізичну адресу вузла-одержувача. Особливої актуальності набуває механізм перетворення (відображення) адрес для ширококомовних мереж, таких як Ethernet, Token Ring і подібних до них. Ця процедура реалізується за допомогою протоколу ARP. Перед початком передачі IP-пакета вузол

повинен визначити, яка фізична адреса в мережі відповідає адресі одержувача, заданій в IP-пакеті. Для цього вузол посилає широкомовний пакет ARP, що містить IP-адресу одержувача. Після цього він чекає відповідь від вузла з даною IP-адресою. Одержувач посилає інформаційний кадр з вказівкою своєї фізичної адреси. З метою скорочення часу передачі пакетів і зменшення числа широкомовних запитів, кожен вузол містить кеш-пам'ять, в якій зберігається таблиця дозволу адрес. За допомогою цієї таблиці задається відповідність між фізичними і IP-адресами. Спочатку фізична адреса шукається в таблиці дозволу адрес. Якщо вузол знаходить відповідна фізична адреса для IP-пакета, то він використовує його для звернення до одержувача. Інакше вузол запускає процедуру ARP, після закінчення якої здійснюється відповідна корекція таблиці дозволу адрес.

Можна наводити й інші доводи за і проти стека протоколів Internet, проте факт залишається фактом — сьогодні це найпопулярніший стек протоколів, широко використовуваний як в глобальних, так і локальних мережах.

МЕРЕЖЕВІ ОПЕРАЦІЙНІ СИСТЕМИ ДЛЯ ЛОКАЛЬНИХ МЕРЕЖ.

Основний напрям розвитку сучасних Мережевих Операційних Систем (Network Operation System - NOS) - перенесення обчислювальних операцій на робочі станції, створення систем з розподіленою обробкою даних. Це насамперед пов'язано із зростанням обчислювальних можливостей персональних комп'ютерів і все більш активним впровадженням потужних багатозадачних операційних систем: OS/2, Windows NT, Windows 95. Окрім цього впровадження об'єктно-орієнтованих технологій (Ole, DCE, IDAPI) дозволяє спростити організацію розподіленої обробки даних. У такій ситуації основним завданням NOS стає об'єднання нерівноцінних операційних систем робочих станцій і забезпечення транспортного рівня для широкого круга завдань: обробка баз даних, передача повідомлень, управління розподіленими ресурсами мережі (directory/name service).

У сучасних NOS застосовують три основні підходи до організації управління ресурсами мережі. Перший - це таблиці об'єктів (Bindery). Використовується в мережевих операційних системах NetWare 286 і NetWare v3.1x. Така таблиця знаходиться на кожному файловому сервері мережі. Вона містить інформацію про користувачів, групи, їх права доступу до ресурсів мережі (даним, сервісним послугам і тому подібне). Така організація роботи зручна, якщо в мережі тільки один сервер. В цьому випадку потрібно визначити і контролювати тільки одну інформаційну базу. При розширенні мережі, додаванні нових серверів об'єм завдань по управлінню ресурсами мережі різко зростає. Адміністратор системи вимушений на кожному сервері

мережі визначати і контролювати роботу користувачів. Абоненти мережі, у свою чергу, повинні точно знати, де розташовані ті або інші ресурси мережі, а для діставання доступу до цих ресурсів - реєструватися на вибраному сервері. Звичайно, для інформаційних систем, що складаються з великої кількості серверів, така організація роботи не відповідає вимогам.

Другий підхід використовується в LANServer і LANMahager - структура доменів (Domain). Всі ресурси мережі і користувачі об'єднані в групи. Домен можна розглядувати як аналог таблиць об'єктів (bindery), тільки тут така таблиця є спільною для декількох серверів, при цьому ресурси серверів є спільними для всього домена. Тому користувачеві для того, щоб дістати доступ до мережі, досить підключитися до домена (реєструватися), після цього йому стають доступні всі ресурси домена, ресурси всіх серверів і пристроїв, що входять до складу домена. Проте і з використанням цього підходу також виникають проблеми при побудові інформаційної системи з великою кількістю користувачів, серверів і, відповідно, доменів. Наприклад, мережі для підприємства або великої розгалуженої організації. Тут ці проблеми вже пов'язані з організацією взаємодії і управління декількома доменами, хоча за змістом вони такі ж, як і в першому випадку.

Третій підхід - Служба найменувань директорій або каталогів (Directory Name Services - DNS) позбавлений цих недоліків. Всі ресурси мережі: мережевий друк, зберігання даних, користувачі, сервери і тому подібне розглядаються як окремі гілки або директорії інформаційної системи. Таблиці, визначальні DNS, знаходяться на кожному сервері. Це, по-перше, підвищує надійність і живучість системи, а по-друге, спрощує звернення користувача до ресурсів мережі. При реєстрації на одному сервері користувачеві стають доступні всі ресурси мережі. Управління такою системою також простіше, ніж при використанні доменів, оскільки тут існує одна таблиця, що визначає всі ресурси мережі, тоді як при доменній організації необхідно визначати ресурси, користувачів, їх права доступу для кожного домена окремо.

Питання і завдання для самостійної роботи

1. Перерахуйте основні апаратні компоненти ЛОМ.
2. У чому полягає основна відзнака локальних і глобальних мереж?
3. Які ресурси може надати робоча станція іншим користувачам однорангової локальної мережі?
4. Як складені імена мережевих ресурсів?

5. Який електронний протокол служить для передачі файлів в мережі?
6. Для чого служить протокол НТТР?
7. Який із способів підключення до мережі Інтернет забезпечує найбільші можливості для доступу до інформаційних ресурсів?
8. Для чого призначена служба FTP в Інтернеті?
9. Що використовується для зберігання файлів, призначених для спільного доступу користувачів мережі?
10. Чим характеризується швидкість передачі даних?
11. Дайте визначення комп'ютерним комунікаціям.

Список літератури

1. Буров С. Комп'ютерні мережі. – Львів. – Бак, 2000. – 400 с.
2. Основи комп'ютерних мереж та Інтернету. – К.: Видавнича група ВНУ, 2006. – 256 с.
3. Суприган О.І. Основи проектування корпоративних комп'ютерних мереж. Навчальний посібник. – Вінниця: ВНТУ, 2008. – 137 с.
4. Юринець В. С., Юринець Р. В. Комп'ютерні мережі. Інтернет: Навч. посібник. – Львів: Видавн. центр ЛНУ імені Івана Франка, 2006. – 526 с.

Розділ VIII. Глобальна комп'ютерна мережа Internet

У попередньому розділі були розглянуті принципи і технології побудови, організації і використання локальних комп'ютерних мереж. Історично склалося, що після розробки методології використання локальних комп'ютерних систем, на початку 60-х років ХХ століття виникла необхідність організації обміну даними між різними локальними мережами.

Для вирішення даного завдання ученими був розроблений спеціальний *протокол міжмережевої взаємодії* (англ. *internet protocol* або IP). Завдяки використанню даного протоколу утворилася і почала швидко розростатися особлива комп'ютерна мережа або, точніше, об'єднання мереж, названа по аналогії з використовуваним протоколом, – *Інтернет* (англ. *Internet*).

Розробка теоретичних основ побудови мережі Інтернет виконувалася з використанням сукупності вимог.

1. *Самоорганізація* – відсутність якої-небудь організації, керівної або контролюючої побудови мережі Інтернет, підключення або від'єднання локальних мереж.

2. *Децентралізація* – відсутність єдиного керівника центрального пристрою або групи пристроїв, що управляють, знищення або пошкодження яких може призвести до порушення роботи мережі Інтернет.

3. *Багатозв'язність* – кожен комп'ютер в мережі Інтернет може бути зв'язаний будь-якою кількістю каналів з будь-якою кількістю інших комп'ютерів.

4. *Некомерційний характер побудови* – відсутність плати або ліцензійних відрахувань за передачу даних за допомогою мережі Інтернет.

5. *Відсутність жорстких вимог до архітектури* підключених до Інтернет локальних мереж, а також до використовуваних в них протоколів.

З урахуванням даних вимог при технічній розробці протоколів мережі Інтернет використовувалися наступні принципи.

1. Всі повідомлення, які передаються, розбиваються на окремі частини – *пакети* – кожен з яких, окрім самих даних містить в собі відомості про відправника і одержувача повідомлення.

2. Кожен пакет відправляється і приймається незалежно від інших і лише після отримання дані з окремих пакетів з'єднуються в початкове повідомлення.

3. Маршрут передачі кожного окремого пакету, тобто послідовність каналів і вузлів мережі, по яких передається пакет, визначається безпосередньо в процесі передачі тими вузлами, які його пересилають.

Саме ці ідеї дозволили об'єднати мережі, що базуються на самих різних операційних системах (Windows, Unix, SunOS/Solaris і ін.), що використовують різні мережеві пристрої (Ethernet, Token Ring, FDDI, ISDN, ATM, SDH і так далі) і зробити мережу Інтернет нечутливою до локальних збоїв і відмов апаратури.

8.1. Пошук інформації в Інтернеті

Адресація комп'ютерів в мережі Інтернет. Для ідентифікації комп'ютерів в рамках мережі Інтернет кожному комп'ютеру і мережевому пристрою на основі спеціальних правил привласнюється унікальна 4-х байтова адреса, яка називається *IP-адресою* (адреса інтернет протоколу). Для зручності використання ця адреса представляється у вигляді чотирьох чисел від 0 до 255, розділених крапками, наприклад, 192.168.1.24.

Можна розрахувати, що максимально допустима кількість мережевих адрес для різних пристроїв, комп'ютерів, серверів, підключених до мережі Інтернет, складе 2564 або біля 4-х мільярдів. Зрозуміло, творцям інтернет-протоколу як засобу об'єднання комп'ютерних мереж декількох учбових закладів або військових організацій, така кількість адрес уявлялася свідомо дуже великою. Проте на сьогодні, з урахуванням масового використання мережі Інтернет для цілей ведення бізнесу, а також домашнього використання і розваг, спільна кількість адрес вже практично вичерпана. Це привело до розробки нового розширеного стандарту адресації мережевих пристроїв. Цей стандарт побудований на принципі використання 6-и байтової адреси і названий IPv6 в протилежність попередньому – IPv4. Така система адресації дозволяє привласнити $2,8 \times 10^{14}$ адресів. Всі раніше привласнені 4-х байтові адреси є підмножиною нової безлічі адрес.

Враховуючи, що людська свідомість не дуже добре пристосована до запам'ятовування послідовностей чисел, була розроблена можливість привласнення комп'ютерам в мережі Інтернет унікальних символьних імен, що отримали назву *доменне ім'я*, а також розроблена технологія автоматичного визначення IP-адреси комп'ютера по його доменному імені і навпаки, доменного імені по IP-адресу.

Доменне ім'я і IP-адреса не тотожні — одна IP-адреса може відповідати безлічі імен, що дозволяє підтримувати на одному комп'ютері безліч різних сервісів. Зворотне теж справедливо — одному імені може бути зіставлене безліч IP-адресов: це дозволяє балансувати мережеве навантаження на сервер.

Доменне ім'я комп'ютера окрім імені самого комп'ютера включає ім'я локальної мережі, в яку він входить (*домен*), а також імена тих мереж, в які входить його мережа. Тобто доменне ім'я будується за принципом поштової адреси.

Сукупність доменних імен, правил їх побудови і дозволу в IP-адреса називається *системою доменних імен* або *DNS* (domain name system)

Для зручності використання, при створенні системи доменних імен були створені так звані домени першого (верхнього) рівня для комерційних (.com), учбових (.edu), урядових (.gov), некомерційних (.org) організацій, а також організацій, що забезпечують функціонування мережі Інтернет (.net). Крім того, були створені домени для кожної країни: .us – США, .eu – Європейський Союз, .fr – Франція, .de – Німеччина, .jp – Японія, .ua – Україна, .ru – Росія і так далі У доменах країн також створюються домени другого рівня для комерційних, учбових, урядових організацій, наприклад .gov.ru, com.ua, .org.eu.

Домен, що входить в домен більш високого рівня, називається *субдоменом* або *піддоменом*. Всі домени, що входять в один домен, називаються *зоною* доменів.

Домени верхнього рівня використовуються тільки для реєстрації в них доменів другого рівня і ніякої IP-адреси не визначають.

Для підтримки функціонування, контролю і управління в кожному домені існує як мінімум один спеціальний комп'ютер, званий *DNS-сервером*. DNS-сервер зберігає інформацію про всі комп'ютери цього домена, про всі субдомени, а також про DNS-сервери тієї зони доменних імен, до якої він належить.

Система DNS містить ієрархію *серверів DNS*. Кожен домен або піддомен підтримується як мінімум одним *авторитетним сервером DNS* (від англ. authoritative — авторитетний, заслуговуючий довіри; часто вживають і інші варіанти перекладу: авторизований, авторитативний), на якому розташована інформація про домен. Ієрархія серверів DNS збігається з ієрархією доменів.

Для визначення IP-адреси комп'ютера, який встановлює з'єднання, звертається до DNS-серверу свого домена, той або повідомляє шукану IP-адресу, або у свою чергу звертається із запитом до якогось авторитетного DNS-серверу про піддомен контрольованої ним зони.

Розглянемо на прикладі роботу всієї системи доменних імен. Припустимо, комп'ютеру необхідно визначити IP-адресу комп'ютера з ім'ям ru.wikipedia.org. Він відправляє серверу DNS запит: «яка IP-адреса у ru.wikipedia.org»? Проте, сервер DNS може не мати відомостей не лише про запитане ім'я, але навіть про весь домен wikipedia.org. В цьому випадку має

місце рекурсія: сервер звертається до кореневого сервера — наприклад, 198.41.0.4. Цей сервер повідомляє — «У мене немає інформації про дану адресу, але я знаю, що 204.74.112.1 є авторитетним для зони org.» Тоді сервер DNS направляє свій запит до 204.74.112.1, але отримує відповідь «У мене немає інформації про даний сервер, але я знаю, що 207.142.131.234 є авторитетним для зони wikipedia.org.» Нарешті, той же запит відправляється до третього DNS-серверу і отримує відповідь — IP-адреса, яка і передається комп'ютеру-клієнтові.

Інтернет – засіб обміну даними. Перший спосіб використання комп'ютерної мережі Інтернет, що історично склався, – передача файлів. При цьому могли передаватися як файли з необхідними даними, так і програмні файли, і файли з якимись повідомленнями і новинами.

Спеціальний протокол, що регламентує процедури обміну файлами так і назвали *протокол передачі файлів* або *FTP* (file transfer protocol). Відповідно до правил протоколу FTP створюються спеціальні, особливим чином організовані сховища файлів – файлові сервери. Інші комп'ютери можуть звернутися до FTP-серверу із запитом списку файлів і папок з файлами, що зберігаються на сервері. А потім, за побажанням користувача, завантажити файл або групу файлів з сервера на локальний комп'ютер. Тобто ініціатором процесу передачі файлу є комп'ютер-одержувач.

Програми-засоби завантаження файлів з використанням протоколу FTP як правило вбудовані в усі сучасні операційні системи. Для завантаження файлу зазвичай достатньо знати доменне ім'я сервера FTP.

Звичайно, передача файлів між комп'ютерами – це дуже корисна можливість для цілей обміну науковими даними, учбовою інформацією або для яких-небудь військових цілей (не забуватимемо, що розроблявся Інтернет як інструмент ведення війни), але справжнім поштовхом до розвитку Інтернету послужило усвідомлення того, що глобальна комп'ютерна мережа може бути використана як засіб спілкування між людьми.

Інтернет – засіб спілкування. Першим інструментом спілкування в рамках мережі Інтернет стала так звана «електронна пошта» (англ. *e-mail*), яка практично в незмінному вигляді існує і зараз.

Електронна пошта є засобом передачі повідомлення (листи) від одного користувача мережі до іншого. При цьому як адреса вказуються доменне ім'я поштового сервера та ім'я користувача цього сервера, розділені символом «@». Наприклад, user@example.com.

На відміну від протоколу FTP, при передачі електронної пошти ініціатором пересилки виступає користувач-відправник повідомлення.

Користувач-адресат одержує свої листи тільки після звертання із спеціальним запитом до свого поштового сервера. Тобто як і у випадку із звичайною паперовою поштою, можна сказати, що електронний лист «приходить» в електронну «поштову скриньку» і чекає, поки одержувач його не «забере».

Для відправки і отримання електронних повідомлень використовуються спеціальні програми, наприклад Microsoft Outlook або sendmail.

При створенні нового повідомлення потрібно вказати адресу електронної пошти одержувача, «тему» або заголовок повідомлення, а також текст повідомлення. Крім того, поштові програми дозволяють «приєднати» до тіла листа ще і деяку кількість файлів з іншими даними – документи, зображення, програми і так далі.

Впровадження системи «електронної пошти» відкрило нові можливості використання мережі Інтернет як засоби спілкування. Проте електронне листування не давало відчуття повноцінного спілкування і обміну інформацією.

Інтернет і гіпертекстові сторінки. Сучасний етап повсюдного проникнення мережі Інтернет почався в 1993 році з появою підмережі www – World Wide Web або «усесвітньої павутини».

Усесвітня павутина – глобальний інформаційний простір, заснований на фізичній інфраструктурі мережі Інтернет, гіпертексті і протоколі передачі даних *HTTP*. Усесвітня павутина корінним чином змінила спосіб використання Інтернету і інформаційні технології в цілому. Часто, кажучи про Інтернет, мають на увазі іменну Усесвітню павутину. Для позначення Усесвітньої павутини також використовують слово веб-сервер (англ. web) і аббревіатуру «WWW».

Усесвітню павутину утворюють *веб-сервери* мережі Інтернет, розташовані по всьому світу. *Веб-сервер* є програма, що запускається на підключеному до мережі комп'ютері і використовує протокол *HTTP* для передачі даних. У простому вигляді така програма отримує по мережі *HTTP*-запрос на певний ресурс, знаходить відповідний файл на локальному жорсткому диску і відправляє його по мережі комп'ютеру, що запитав. Складніші веб-сервери здатні динамічно розподіляти ресурси у відповідь на *HTTP*-запит.

HTTP (англ. *HyperText Transfer Protocol* — «протокол передачі гіпертексту») — протокол прикладного рівня передачі даних насамперед у вигляді текстових повідомлень. Основою *HTTP* є технологія «*клієнт-сервер*», тобто передбачається існування споживачів (клієнтів), які ініціюють

з'єднання і посилають запит, і постачальників (серверів), які чекають з'єднання для отримання запиту, проводять необхідні дії і повертають назад повідомлення з результатом.

Для ідентифікації ресурсів (часто файлів або їх частин) у Усесвітній павутині і для визначення місцезнаходження ресурсів в мережі використовуються одноманітні локатори ресурсів *URL* (англ. Uniform Resource Locator).

Такі URL-локатори включають:

- 1) ідентифікатор протоколу, використовуваного для доступу до даних;
- 2) ім'я сервера в системі доменних імен DNS, доменне ім'я (або безпосередньо IP-адреса в числовому записі) входить до складу URL для позначення веб-сервера-сервера;
- 3) адреса запрошуваного документа (веб-сторінки) на вказаному сервері.

Прикладом URL може служити наступний:

<http://www.ccssu.crimea.ua/tnu/image/vid.jpg>

де <http://> - вказівка на використання протоколу HTTP, www.ccssu.crimea.ua - DNS-ім'я сервера в мережі [www](http://www.ccssu.crimea.ua), [tnu/image/](http://www.ccssu.crimea.ua/tnu/image/) - шлях до необхідного файлу [vid.jpg](http://www.ccssu.crimea.ua/tnu/image/vid.jpg).

Усесвітня павутина нерозривно пов'язана з поняттями гіпертексту і гіперпосилання. Велика частина інформації у Вебі вдає із себе саме гіпертекст. Для полегшення створення, зберігання і відображення гіпертексту в Усесвітній павутині традиційно використовується мова *HTML* (англ. HyperText Markup Language), мова розмітки гіпертексту. Робота по розмітці гіпертексту називається *версткою*. Після HTML-розмітки гіпертекст, що вийшов, поміщається в файл, такий HTML-файл є найпоширенішим ресурсом Усесвітньої павутини

http://ru.wikipedia.org/wiki/%D0%92%D0%B8%D0%BA%D0%B8%D0%BF%D0%B5%D0%B4%D0%B8%D1%8F.%D0%A1%D1%81%D1%8B%D0%BB%D0%BA%D0%B8_%D0%BD%D0%B0_%D0%B8%D1%81%D1%82%D0%BE%D1%87%D0%BD%D0%B8%D0%BA%D0%B8.

Після того, як HTML-файл стає доступним веб-серверу-серверу, його починають називати «*веб-сторінкою*».

Набір веб-сторінок утворює *веб-сайт*. У гіпертекст веб-сторінок додаються гіперпосилання. Гіперпосилання допомагають користувачам Усесвітньої павутини легко переміщатися між ресурсами (файлами) незалежно від того, знаходяться ресурси на локальному комп'ютері або на видаленому сервері.

Для проглядання інформації, отриманої від веб-сервера-сервера, на клієнтському комп'ютері застосовується спеціальна програма — *веб-браузер*. Основна функція веб-сервера-браузера — відображення *гіпертексту*.

Одним з найбільш поширених веб-браузерів є браузер Internet Explorer компанії Microsoft.

Використання програми Microsoft Internet Explorer. Програма *Internet Explorer* призначена для проглядання веб-сторінок. Вона підтримує всі нові можливості, що закладаються розробниками веб-сторінок, включаючи звуковий супровід і відеосупровід інформації, що відображується.

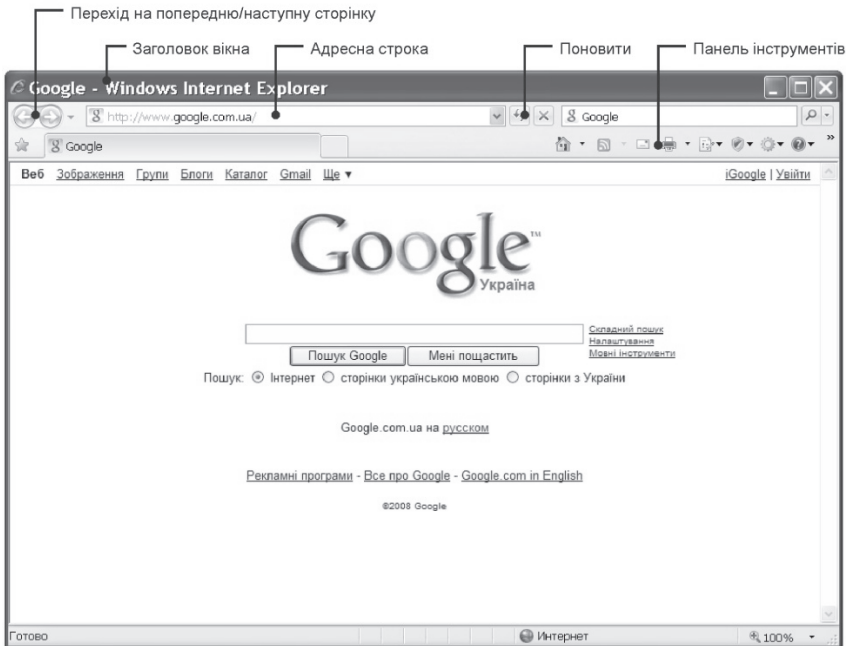


Рис. 8.1.1. Робоче вікно програми *Internet Explorer*

Для запуску програми слід виконати наступні дії.

- Натиснути на кнопку *Пуск* на *Панелі завдань*. Відкриється основне меню *Windows*.







- Виберіть команду *Програми* → *Інтернет (Internet Explorer)*. Після запуску програми на екрані з'явиться вікно навігатора Internet Explorer (рис.8.1.1). Розглянемо назви і призначення всіх елементів.

Заголовок вікна - стандартний заголовок Windows, в якому, окрім назви програми, відображується ще і назва відкритої Web-сторінки.

Під заголовком розташовується *меню*. З його допомогою можна вибрати будь-яку команду Internet Explorer.

Нижче за меню знаходиться *панель інструментів*. На цій панелі розташовані значки, що позначають різні дії, які можна виконати в процесі роботи. Вибір команди здійснюється клацанням миші по відповідному значку. Описи команд, які можуть бути запущені з панелі інструментів, наведені в таблиці 8.1.1.

Таблиця 8.1.1. Опис команд


Кнопка	Назва, призначення
	<i>Назад і вперед</i> - дозволяють переміщатися по проглянутих документах.
	<i>Зупинити (Stop)</i> - перериває завантаження документа. Зупинити завантаження документа можна також натиснувши клавішу <i>ESC</i> .
	<i>Відновити (Refresh)</i> - дає можливість користувачеві повторити отримання відкритого документа.
	<i>Додому (Home)</i> - повертає користувача на Web-сторінку, зареєстровану як стартову при запуску навігатора.
	<i>Пошук (Search)</i> - відкриває панель для пошуку потрібної інформації в Інтернет.
	<i>Вибране (Favorites)</i> - дозволяє перейти до списку адрес, створених користувачем.

Під панеллю інструментів розташовано *поле для введення адреси* сторінки, яку можна проглянути. Замість поля для введення адреси можна відображувати панель з декількома посиланнями на Web-сторінки. *Посилання* - це значок, за яким закріплена адреса якої-небудь Web-сторінки в мережі Інтернет. Для того, щоб відображувати панель посилань, клацніть покажчиком миші по полю *Посилання*. Знов відображувати панель з полем для введення адреси можна, клацнувши покажчиком миші по полю *Адреса*.

Рядок стану призначений для індикації тих дій, які в даний момент виконує програма Internet Explorer. Професійна робота з Internet Explorer обов'язково включає вміння знатися на написах, що з'являються на цьому рядку. В процесі роботи з Інтернет в рядок стану періодично виводяться

повідомлення про адреси джерел інформації, режими очікування, готовності запитаного документа і сукупність інших корисних відомостей.

Основні принципи роботи з програмою Internet Explorer. Для того, щоб спростити перегляд і пошук інформаційних ресурсів мережі Інтернет, потрібно освоїти деякі основні прийоми роботи з навігатором Internet Explorer.

Почати проглядання мережі Інтернет можна з домашньої сторінки, тобто сторінки, що з'являється при запуску Internet Explorer. Для переходу на іншу сторінку виберіть будь-яке посилання на цій сторінці. Щоб дізнатися, чи є даний елемент сторінки посиланням, перемістіть на нього покажчик миші. При вказівці на посилання він перетвориться на значок  (рука), а в рядку стану з'явиться адреса ресурсу, на який указує дане посилання. Посиланням може бути картинка, об'ємне зображення або виділений текст. Web-сторінка, що відображується у вікні навігатора в даний момент часу, називається *активною*.


Проглядання ресурсів Інтернет. Щоб отримати який-небудь ресурс з мережі Інтернет, введіть в адресний рядок його URL (рисунок 8.1.2), наприклад, <http://www.tnu.crimea.ua> або <http://rada.gov.ua>, а потім натисніть кнопку *Перехід* або клавішу *Enter*.



Рис. 8.1.2. Адресний рядок

Необхідні адреси або беруться з довідників, або конструюються користувачем, виходячи з розуміння структури адреси і інтуїції, або знаходяться за допомогою спеціальних засобів пошуку інформації в Інтернет.

Коли користувач почне вводити URL в адресний рядок, з'явиться список схожих адрес, з якого можна вибрати потрібну адресу. Якщо адреса ресурсу виявиться неточною, то Internet Explorer виконає пошук відповідної адреси серед схожих адрес. Після переходу на web-сторінку можна знайти на ній певний текст, викликавши меню кнопкою пошуку і вибрати в меню пункт *Знайти на цій сторінці*.

Відкриття останніх проглянутих Web-сторінок. Щоб повернутися на останню з проглянутих сторінок, натисніть на панелі інструментів кнопку *Назад*, а щоб проглянути сторінку, яку ви проглядали до того, як натискували кнопку *Назад*, натисніть кнопку *Вперед*. Можна також розкрити список декількох проглянутих раніше сторінок. Для цього натисніть невелику направлену вниз стрілку  поряд з кнопкою *Назад* або *Вперед*.

Перехід на домашню Web-сторінку. Щоб повернутися на сторінку, яка з'являється щоразу при запуску оглядача Internet Explorer, натисніть кнопку *Додому*.

Перехід на Web-сторінку із списку вибраних сторінок. Для того, щоб вибрати із списку вибраних сторінок потрібну Web-сторінку, натисніть кнопку *Вибране* (див. докладніше п.3).

Виведення списку проглянутих раніше Web-сторінок. Посилання на сторінки, які були відкриті протягом останнього періоду часу, зберігаються в журналі. Для того, щоб вивести список проглянутих раніше сторінок і швидко перейти до потрібної сторінки, натисніть кнопку *Вибране* і виберіть закладку *Журнал*. У журналі також будуть перераховані файли і папки, відкриті раніше на даному комп'ютері за допомогою навігатора. Для відображення шуканої web-сторінки у вікні навігатора клацніть покажчиком миші по відповідному посиланню.

Дії у випадку, якщо Web-сторінка не відкривається. Наприклад, якщо очікування появи сторінки займає багато часу, натисніть кнопку *Зупинити*. При отриманні повідомлення про неможливість відкрити яку-небудь Web-сторінку або щоб переконатися, що завантажений останній варіант сторінки, натисніть кнопку *Відновити*.

Спрощення доступу до часто відвідуваних сторінок. Після проглядання потрібних web-сторінок можна зберегти посилання на них, що дозволить легко відкрити їх згодом. Існує три способи, що дозволяють зберегти посилання на web-сторінку, що Вас цікавить.

Додавання web-сторінки до списку Вибране. Для того, щоб додати сторінку в папку *Вибране*, перейдіть на сторінку, яку потрібно додати. У меню, викликаному кнопкою *Вибране*, виберіть пункт *Додати в папку Вибране* і вкажіть нову назву для даної сторінки. У міру збільшення кількості вибраних сторінок їх можна упорядкувати по папках за допомогою команди *Упорядкувати вибране*. Щоб згодом відкрити яку-небудь сторінку, занесену в список *Вибране*, на панелі інструментів треба натиснути кнопку *Вибране* і вибрати цю сторінку із списку.


Додавання посилання на web-сторінку в панель посилань. Панель *Посилання* розташована поряд з адресним рядком, завдяки чому на ній зручно розміщувати посилання на часто використовувані web-сторінки. Щоб викликати сторінку, просто виберіть посилання на неї. Додати сторінку на панель посилань можна декількома способами:

- перетягнути значок вибраної сторінки з адресного рядка на панель посилань;
- перетягнути посилання з web-сторінки на панель посилань;
- перетягнути посилання в папку *Посилання* із списку *Вибране*.


Якщо панель посилань відсутня, відкрийте меню *Вигляд*, виберіть команду *Панелі інструментів* і клацніть *Посилання*.

Заміна домашньої сторінки. *Домашня сторінка* - це та сторінка, яка відображується при кожному запуску оглядача Internet Explorer. Бажано, щоб це була та сторінка, яку користувач часто переглядає. В якості домашньої сторінки можна також вибрати сторінку, пристосовану для того, щоб швидко знаходити будь-яку необхідну інформацію, наприклад, www.google.com

Для того, щоб змінити домашню сторінку, потрібно виконати наступні дії:

- перейдіть на сторінку, яку слід відкривати при кожному запуску Internet Explorer;
- натисніть невелику направлену вниз стрілку  поряд з кнопкою *Додому*;
- виберіть пункт *Додавання* або *Зміна домашньої сторінки*;
- натисніть кнопку *Використовувати як єдину домашню сторінку*.

Збереження Web-сторінок. Для того, щоб зберегти web-сторінку на комп'ютері, необхідно виконати наступні дії:

- на панелі інструментів вибрати кнопку  *Сторінка*, в меню, що з'явилося, вибрати пункт *Зберегти як...*;
- вказати папку, в яку хочете помістити сторінку;
- у полі *Ім'я файлу (File Name)* ввести відповідне ім'я.

Щоб зберегти всі файли, необхідні для відображення даної сторінки, включаючи малюнки, кадри і таблиці стилів, виберіть в полі *Тип файлу (File Type)* варіант *Web-сторінка повністю*. В цьому випадку збережеться вся інформація з Web-сторінки. Якщо потрібно зберегти тільки активну Web-сторінку, виберіть *Веб-сторінка, тільки HTML*. Ця команда приведе до збереження інформації, що міститься на web-сторінці, але при цьому не збережуться малюнки, звукові ефекти і інші файли. Щоб зберегти тільки текст, що міститься на активній web-сторінці, виберіть *Тільки текст*. Вибір цього типу файлу приведе до збереження інформації, що міститься на web-сторінці, в звичайному текстовому форматі.

Істотне те, що збережені сторінки можна буде проглядати за допомогою навігатора без підключення до Інтернет.

Збереження малюнків, файлів або тексту з web-сторінок. При прогляданні web-сторінок зустрічається інформація, яку необхідно зберегти на локальному диску свого комп'ютера. Це може бути графічна інформація, вбудована в web-сторінку, текстова інформація, що знаходиться на сторінці, або файли різних форматів, а також інші ресурси, посилання на яких вказані на активній сторінці. Можна зберегти як всю сторінку повністю, так і окрему

її частину: текст, зображення або посилання. Крім того, можна роздрукувати web-сторінки для тих людей, у яких немає доступу в Інтернет або відсутній комп'ютер.

Для того, щоб зберегти малюнок, наведіть на нього покажчик миші і клацніть праву кнопку, а потім виберіть *Зберегти малюнок як...* Для збереження об'єкту (web-сторінки, файлу і тому подібне), на який вказує посилання, розташоване на активній сторінці, наведіть на посилання покажчик миші і клацніть праву кнопку, а потім виберіть *Зберегти об'єкт як...*

НАЛАШТУВАННЯ НАВІГАТОРА

Вибір кодування символів. У більшості web-сторінок міститься інформація, на підставі якої навігатор може встановити необхідне мовне кодування (мова і набір символів).

Якщо на web-сторінці така інформація відсутня, то Internet Explorer зазвичай сам визначає відповідне мовне кодування при включеній функції автоматичного вибору.

Для того, щоб включити функцію автоматичного вибору в меню *Вид (View)* навігатора Internet Explorer, вкажіть пункт *Кодування (Encoding)* і перевірте, щоб був відмічений галочкою пункт меню *Автоматичний вибір (Auto Select)*. Якщо галочки немає, виберіть цей пункт.

Якщо за допомогою функції автоматичного вибору не вдається визначити правильне мовне кодування, а ви знаєте, яке саме кодування потрібне в даному випадку, то його можна вибрати вручну. У меню *Вигляд (View)* виберіть пункт *Кодування (Encoding)*, потім *Додатково (More)* і виберіть відповідну мову.


При підготовці російськомовних web-сторінок найчастіше використовують два кодування: *Кирилиця (Windows-1251) (Cyrillic (Windows-1251))* і *Кирилиця (KOI8-R) (Cyrillic (KOI8-R))*.

Зміна розміру шрифту. Для зміни розміру шрифту символів, що відображаються навігатором, вкажіть в меню *Вигляд (View)* на пункт *Розмір шрифту (Fonts)* і виберіть потрібний розмір.

Відключення мультимедіа. Для зменшення часу завантаження web-сторінок можна заборонити завантаження мультимедіа елементів (графічних зображень, звуків і відео). Для цього виконайте наступні дії:

- у оглядачі Internet Explorer в меню *Сервіс (Tools)* виберіть *Властивості оглядача (Internet Options)*;
- перейдіть на вкладку *Додатково (Advanced)*.
- у групі *Мультимедіа (Multimedia)* зніміть один або декілька з наступних прапорців: *Відображувати малюнки (Show pictures)*,


Відтворювати анімацію (Play animations), Відтворювати звуки (Play sounds), Відтворювати відео (Play videos).

Навіть якщо прапорці *Відображувати малюнки (Show pictures)* і *Відтворювати відео (Play videos)* зняті, можна подивитися окремих малюнок (або анімацію), клацнувши по значку  правою кнопкою миші і вибравши пункт *Показати малюнок (Show picture)*. Якщо малюнок на активній сторінці продовжує залишатися видимим після зняття прапорця *Відображувати малюнки (Show pictures)*, а необхідно його приховати, в меню *Вигляд (View)* виберіть *Відновити (Refresh)*.

Збільшення місця на диску для тимчасового зберігання сторінок Інтернет. Для прискорення роботи Internet Explorer завантажує вже проглянуті web-сторінки не з Інтернет, а із спеціального буфера, який є папкою на диску комп'ютера, куди програма поміщає web-сторінки при їх першому перегляді. Щоб збільшити місце на диску для тимчасового зберігання сторінок Інтернет, виконайте наступні дії:

- у Internet Explorer в меню *Сервіс (Tools)* виберіть *Властивості оглядача (Internet Options)*;
- перейдіть на вкладку *Загальні (General)* і натисніть кнопку *Налаштування (Settings)*;
- для збільшення місця на диску, що виділяється для тимчасового зберігання сторінок, перемістіть покажчик праворуч.

Налаштування часу зберігання адрес проглянутих web-сторінок.

У меню, що викликається кнопкою  *Сервіс (Tools)*, виберіть *Властивості оглядача (Internet Options)* і перейдіть на вкладку *Загальні (General)*. Група елементів управління *Журнал (History)* задає інтервал часу, після закінчення якої адреси проглянутих сторінок будуть видалені. За умовчанням цей час складає двадцять днів. Щоб змінити його, клацніть мишею на полі, де вказаний час і введіть з клавіатури бажане значення. Кнопка *Стерти (Clear History)* служить для видалення всіх адрес з папки.

Способи активного відображення інформації в Усесвітній павутині. Інформація у Вебі може відображатися як пасивно (тобто користувач може тільки прочитувати її), так і активно — тоді користувач може додавати інформацію і редагувати її. До способів активного відображення інформації в Усесвітній павутині належать:

- гостьові книги;
- форуми;
- чати;
- блоги;
- wiki-проекти;
- системи управління контентом.

Слід зазначити, що це ділення вельми умовно. Так, скажімо, блог або гостьову книгу можна розглядувати як окремий випадок форуму, який, у свою чергу, є окремим випадком системи управління контентом. Зазвичай різниця виявляється в призначенні, підході і позиціюванні того або іншого продукту. Розглянемо найбільш поширені серед способів активного відображення інформації.

Чат. Чат (англ. chat — розмова) — засіб спілкування користувачів по мережі в режимі реального часу, а також програмне забезпечення, що дозволяє організовувати таке спілкування. Зазвичай під словом «чат» мається на увазі обмін текстовими повідомленнями. Рідше використовують терміни голосовий чат, відеочат (частіше — відеоконференція) і так далі. Ці технології все ще знаходяться в стані, що розвивається, і відповідна термінологія ще не виробилася.

Існує декілька різновидів програмної реалізації чатів.

- HTTP- або веб-чати. Такий чат виглядає як звичайна веб-сторінка, де можна прочитати останні декілька десятків фраз, написаних учасниками чату і модераторами. Сторінка чату автоматично оновлюється із заданою періодичністю.

- IRC спеціалізований протокол для чатів.
- Програми-чати для спілкування в локальних сетях (наприклад, Vypress Chat Intranet Chat) . Часто є можливість передачі файлів.

По застосуванню чати діляться на:

- all2all групова комунікація (наприклад IRC Yahoo! Chat AVACS Live Chat);
- p2p персональні комунікації (наприклад ICQ Jabber Skype Yahoo! Messenger AOL Instant Messenger) — особисте спілкування;
- b2b ділові — робота в групах;
- b2c споживчі — підтримка клієнтів компанії на корпоративному сайті.

Блоги. Блог (англ. blog, від «Web log» «мережевий журнал або щоденник подій») – це веб-сайт, основний зміст якого — записи, що регулярно додаються, зображення або мультимедіа. Для блогів характерні недовгі записи часової значущості, відсортовані в зворотньому хронологічному порядку (останній запис зверху). Відзнаки блога від традиційного щоденника обумовлюються середовищем: блоги зазвичай публічні і передбачають сторонніх читачів, які можуть вступити в публічну полеміку з автором (у відгуках до блог-запису або своїх блогах).

➤ Блог - це особистий сайт користувача, доступний суспільному перегляду, який складається з регулярно оновлюваних записів, зображень і мультимедіа. Передбачає полеміку читача з автором.

Блогерами називають людей, які ведуть блог. Сукупність всіх блогів мережі прийнято називати блогосферою. По авторському складу блоги можуть бути особистими, груповими (корпоративними, клубними) або суспільними (відкритими). За змістом — тематичними або спільними.

Для блогів характерна можливість публікації відгуків (т.з. «коментарів», коментів) відвідувачами. Вона робить блоги середовищем мережевого спілкування, що має сукупність переваг перед електронною поштою, групами новин, веб-форумами і чатами.

Вікіпедія. Вікіпедія (англ. Wikipedia) — багатомовна загальнодоступна вільно розповсюджена універсальна енциклопедія. Створюється на багатьох мовах світу колективною працею добровільних авторів, що використовують технологію вікі. З моменту зародження на початку 2001 року і понині Вікіпедія неухильно росте і набирає популярності у користувачів Мережі.

Вікіпедія за об'ємом відомостей і тематичним охоптом вважається за найповнішу енциклопедію, що коли-небудь створювалися за всю історію людства. Одним з основних достоїнств Вікіпедії як універсальної енциклопедії – це можливість представити інформацію на рідній мові, зберігаючи її цінність в аспекті культурної приналежності.

Статті Вікіпедії пишуться спільно добровольцями у всьому світі, і майже всі її статті можуть бути змінені всіма, хто має доступ до Інтернету. Популярність її постійно зростає з моменту створення; в даний час Вікіпедія входить в десятку найбільш відвідуваних веб-сайтів світу.

Вікіпедія — це ще і свого роду дискусійний форум. Розбіжності, що виникають серед авторів, обговорюються на спеціально відведених сторінках. Біля кожної статті є своя сторінка обговорення. Тут кожен може приєднатися до дискусії і висловити свою точку зору з приводу представлених матеріалів.

«Віртуальна реальність» в Інтернет. Віртуальна реальність (лат. virtus — потенційний, можливий; лат. realis — дійсний, такий, що існує) — створюваний технічними засобами світ на будь-якому субстраті і передаваний суб'єктові через його звичні для сприйняття матеріального світу відчуття: зір, слух, нюх та інші.

http://ru.wikipedia.org/wiki/%D0%92%D0%B8%D1%80%D1%82%D1%83%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%80%D0%B5%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D1%81%D1%82%D1%8C - cite note-0 Також неідеальна реальність, не відповідна певним критеріям істинності.

Синоніми: штучна реальність, електронна реальність, комп'ютерна модель реальності. Близькі за значенням терміни: потенційна реальність, можливі світи.

Для створення переконливого комплексу відчуттів реальності комп'ютерний синтез властивостей і реакцій віртуальної реальності повинен проводитися в реальному часі.

Об'єкти віртуальної реальності повинні поводитися аналогічно об'єктам матеріальної реальності. Користувач може мати можливість впливати на об'єкти віртуальної реальності. У віртуальних світах створена фізика, подібна реальній (гравітація, властивості води, зіткнення з предметами і т. п.), але часто в розважальних цілях користувачі віртуальних світів можуть більше, ніж можливо в нашому всесвіті, наприклад, літати, створювати будь-які предмети і тому подібне.

8.2. Створення WEB сторінки

Основи мови HTML. HTML (HyperText Markup Language) - мова гіпертекстової розмітки документів. Призначення HTML в тому, щоб зробити документи придатними для читання з екрану монітора.

Для створення HTML документів використовують текстові редактори (наприклад, Блокнот), текстові процесори (Word), редактори тегів HTML і візуальні HTML-редактори. Можна створити HTML документ в простому Блокноті. Дотримуючись певного стандарту і записавши в текстовому файлі HTML код, зберігши на жорсткому диску і змінивши розширення на .html або .htm ви отримаєте повноцінну web сторіночку.

Структура HTML документа. *Теги* - це інструменти розмітки тексту. Теги можуть писатися як рядковими, так і прописними буквами. Теги бувають парними і не парними. Як приклад парного тега можна привести тег `<html></html>`. Цей тег починає і закінчує будь-який HTML документ. Друга частина парного тега відрізняється від першої тільки наявністю символу `"/"`, проте перша частина тега може містити і додаткові параметри. Наприклад, в тегові ``, параметр `size="4"` визначає розмір тексту. Прикладом непарного тега є `<hr>` - тег вставки в HTML документ горизонтальної лінії, такий як в кінці цього абзацу.

Обов'язкові теги мови HTML. Будь-який HTML документ повинен містити наступні теги `<html></html>`, `<head></head>`, `<body></body>`, `<title></title>`. Порядок розташування тегів в HTML документі представлений нижче

```
<html>  
<head>
```

```
<title>Назва вашої сторінки</title>
</head>
<body>
  Тіло документа
</body>
</html>
```

Усередині тега `<head></head>` розташовується назва Вашого HTML документа (найчастіше саме його видно як посилання в результатах пошуку пошуковими машинами), окрім цього тега усередині конструкції `<head></head>` можуть розташовуватися так звані Мета Теги. Їх призначення і опис наводиться в довіднику по Мета Тегам.

Тіло HTML документа. Заголовки в мові HTML виділяються тегами `<h1></h1>`, `<h2></h2>` .., `<h6></h6>`. Таким чином існує 6 рівнів заголовків. Можна поекспериментувати - вставити замість «Тіло документа» в попередньому прикладі `<h1>HTML - це просто</h1>`, зберегти в текстовому форматі, змінити розширення документа .txt на .html і відкрити його в браузері.

Абзаци в мові HTML. Абзац в HTML документі полягає в тег `<P></P>`. Таким чином, конструкція `<p> - Як настрої?</p><p> - Не погано.</p>` виглядатиме таким чином:

- Як настрої?
- Не погано.

Виділення тексту в HTML. Для виділення тексту, або області тексту в HTML використовують теги ``, `<i></i>`, `<u></u>`, таким чином, рядок `жирний <i>курсив</i> <u>підкреслений</u> <u><i>жирний підкреслений курсив</i></u>` виглядатиме

жирний курсив підкреслений жирний підкреслений курсив

Ненумеровані списки в HTML. Ненумеровані списки прописують в HTML коді таким чином `пункт 1пункт 2пункт 3`, що дасть список вигляду

- пункт 1
- пункт 2
- пункт 3

Нумеровані списки в HTML. Нумеровані списки прописують наступними тегами `пункт 1пункт 2пункт 3`, вийде

1. пункт 1
2. пункт 2
3. пункт 3

Вкладені списки в HTML

Приклад вкладеного списку

```
<ul>
<li>Глава 1
</li><li>Розділ 1</li>Розділ 2</ul>
<li>Глава 2
</li><li>Розділ 1</li>Розділ 2</ul>
</ul>
```

Результат запису в HTML коді буде наступним:

- Глава 1
 - Розділ 1
 - Розділ 2
- Глава 2
 - Розділ 1
 - Розділ 2

Спеціальні символи в тексті документа. Спеціальні символи в тексті документа прописуються таким чином:

< - ліва дужка (<)
> - права дужка (>)
& - (&)
" - лапки ("")

Так, рядок вигляду < "Язык HTML" > виглядатиме <"Мова HTML" >

Переривання рядка в тексті HTML документа. Переривання рядка здійснюється тегом
. Наприклад рядок тексту Автор
рік видання
наклад дасть наступний результат:

Автор
Рік видання
Наклад

Гіперпосилання. Гіперпосилання в мові HTML прописується тегом <a>. Наприклад запис вигляду

```
<a href="html.rar">Скачати підручник мови HTML</a>
```

виглядатиме так:

[Скачать учебник языка HTML](#)

Параметр *href* визначає місце документа, на який посилається посилання, в прикладі воно посилається на документ *html.rar*, розташований в тій же директорії, що і сторінка HTML. Абсолютне посилання прописується таким чином:

```
<a href="http://webdesign.net-soft.ru/html.rar">Скачати підручник мови HTML</a>
```

Виглядає воно аналогічно попередньому посиланню, але визначає точне місце розташування документа.

Правила розставлення відносних посилань в мові HTML. Якщо документ, на який посилається посилання розташований в директорії (папці) на рівень нижче, скажемо *dir*, то параметр *href* посилання матиме вид *href="dir/html.rar"*, а якщо документ розташований в директорії на рівень вище, то необхідно записати *href="../html.rar"*.

Вставка малюнків в HTML сторінку. Малюнок в HTML документ вставляється таким чином:

```

```

. Розберемо, яке значення параметрів папки

```
<img>
```

. Як видно, це одиночний тег. Параметр *src* задає шлях до зображення (абсолютний або відносний). Правила вказівки відносного шляху такі ж, як і для посилання. Параметри *width* і *height* визначають ширину і висоту малюнка в пікселях в HTML документі. Параметр *alt* завдає альтернативний текст - той текст, який відображується в браузері, якщо відключено завантаження графіки.

Фрейми в HTML документі. *Фрейм* – це рамка, в яку завантажується інший HTML документ. Багато сайтів мають фреймову структуру. Часто меню сайту завантажується у фрейм. Як це робиться:

```
<iframe name="I2" src="menu.htm" width="200" scrolling="no" height="500" marginwidth="2" marginheight="2">Ваш браузер не підтримує фрейми</iframe>
```

Параметр *name* - ім'я фрейма, *src* - шлях до завантажуваної у фрейм сторінки, *width*, *height* відповідно ширина і висота рамки. Параметр *scrolling* визначає, чи відображуються смуги прокрутки у фреймі, якщо цей параметр не прописаний в HTML коді, то смуги прокрутки відображуються при необхідності, якщо він рівний "no", то смуги прокрутки не відображуються, якщо - "yes", то - відображуються у будь-якому випадку. Параметр *name* використовується для завдання кінцевої рамки за умовчанням – рамки, в яку завантажуватимуться сторінки при переході по гіперпосиланнях HTML документа. Це здійснюється додаванням в HTML код тега

```
<base>
```

 між тегами

```
<head></head>
```

. Для нашої рамки

```
<base target="I2">
```

.

Таблиці в мові HTML. Таблиці в HTML документ вставляються таким чином:

```
<table border="1" style="border-collapse: collapse" bordercolor="#111111">
<tr><td>Ячейка11<td>Ячейка12</tr>
<tr><td>Ячейка21<td>Ячейка22</tr>
```

</table>

Таблицю відкриває і закриває тег <table></table>, тег <tr></tr> - визначає стовпець, одиночний тег <td> визначає клітинку в стовпці, таким чином представлений код прописує таблицю розміром 2x2, яка в HTML документі виглядатиме таким чином

Ячейка11	Ячейка12
Ячейка21	Ячейка22

Теги *style* і *bordercolor* визначають відповідно стиль відображення таблиці і колір межі.

Питання і завдання для самостійної роботи

1. Що є Інтернетом?
 2. Опишіть основні можливості Інтернету.
 3. Охарактеризуйте поняття WWW.
 4. Що таке гіпертекст?
 5. Що таке Web - сторінка?
 6. Що таке електронна пошта? Як формується адреса користувача?
 7. Як будується доменна система імен? Яке ім'я привласнене в світовій комп'ютерній мережі Україні?
 8. Як називається пристрій, за допомогою якого можна вийти в Інтернет, використовуючи телефонну лінію?
 9. Складіть свою першу сторінку Web за наведеним зразком.
- 1) Завантажте Internet Explorer і прогляньте елементи вікна програми.
 - 2) Завантажте сторінку <http://tnu.crimea.ua> і зробіть цю сторінку домашньою.
 - 3) Прогляньте сторінку, перегортуючи її і вибираючи посилання. Для відкриття посилання в новому вікні Internet Explorer клацніть правою кнопкою миші по посиланню і виберіть в контекстному меню *Відкрити посилання в новому вікні (Open in new window)*. Ознайомтеся з компонентами Web-сторінки. При роботі дотримуйтеся наступних правил:
 - для перегортування сторінки використовуйте смуги прокрутки;
 - для переходу на попередню сторінку слід натиснути кнопку *Назад (Back)*, а для повернення - кнопку *Вперед (Forward)*;

- якщо адреса не завантажується або завантажується довго, натисніть кнопку *Зупинити (Stop)* і введіть іншу адресу;
 - якщо сторінка завантажилася частково, натисніть кнопку *Відновити (Refresh)*.
- 4) Збережіть інформацію із сторінки: у форматі Web-сторінка повністю, у форматі тільки *html*, у вигляді тексту. Збережіть малюнок.
- 5) Завантажте друге вікно Internet Explorer і відчиніть в ньому збережені файли (меню *Файл (File)*, команда *Відкрити (Open)*).
- 6) Переключіться в перше вікно Internet Explorer і по посиланню ВХІД перейдіть на іншу Web-сторінку. Коли сторінка завантажиться, клацніть по посиланню *Історія*. Ознайомтеся з історією університету, подорожуючи по посиланнях, розташованих у правій частині вікна оглядача.

Список літератури

1. Мархвида И.В. Создание Web-страниц: HTML, CSS, JavaScript / И.В. Марзвида. – Мн.: Новое знание, 2002. – 352 с.
2. Основи Інтернету. Навч. посібник. / О.М. Левченко, І.О. Завадський, Н.С. Прокопенко. За заг. ред. А.М. Гуржія. - К. : ВНУ, 2008. - 319 с.
3. Рамський Ю.С., Резіна О.В. Вивчення інформаційно-пошукових систем мережі Інтернет: Навч. посіб. – К.: НПУ імені М.П. Драгоманова, 2004. – 60 с.
4. Юринець В. Є., Юринець Р. В. Комп'ютерні мережі. Інтернет: Навч. посібник. – Львів: Видавн. центр ЛНУ імені Івана Франка, 2006. – 526 с.

Розділ IX. ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ В ЕКОНОМІЦІ І МЕНЕДЖМЕНТІ

В цьому розділі вивчаються основи комп'ютерних інформаційних технологій та інтегрованої економічної інформаційної системи управління підприємством (ERP-система).

9.1. Комп'ютерні інформаційні технології

В даний час однією з найважливіших і життєво значущих для керівників будь-якого рівня проблем є використання інформаційних технологій в розробці управлінських рішень. Сучасний рівень розвитку бізнесу пред'являє принципово нові вимоги до інформаційного обслуговування, у тому числі забезпечення *швидкості* передачі інформації, її *актуальності*, *достовірності* і *вчасності* надання кінцевому користувачеві. У гострій конкурентній боротьбі виживають і добиваються успіху ті організації, в яких керівництво володіє умінням ухвалювати ефективні рішення, використовуючи додаткові можливості, які дають сучасні інформаційні технології.

Попит на інформацію і інформаційні послуги забезпечує розвиток, розповсюдження і все більш ефективно використання інформаційних технологій. Під впливом нових інформаційних технологій відбуваються корінні зміни в технології управління, а саме, автоматизуються процеси обґрунтування і ухвалення рішень, автоматизуються організація їх виконання, підвищується кваліфікація і професіоналізм фахівців, зайнятих управлінською діяльністю. Проникнення у всі сфери життя інформаційних технологій не залишило в стороні від цієї тенденції і сферу економіки і управління підприємством.

Сучасні технології організації інформаційних процесів пов'язані, перш за все, з використанням комп'ютерних технологій.

➤ Під *комп'ютерними інформаційними технологіями* розуміється процес, що використовує сукупність методів і засобів реалізації операцій *збору, реєстрації, передачі, накопичення і обробки інформації* за допомогою комп'ютерів і комп'ютерних мереж для вирішення управлінських завдань економічного об'єкту.

Поняття «Інформаційні технології» стосується всього перетворення інформації, у тому числі і на паперовій основі, поняття «Комп'ютерні інформаційні технології» належить до обробки інформації на основі використання засобів комп'ютерної техніки.

Комп'ютеризація управління, розвиток економічної, технічної і соціальної сфери викликає необхідність прискорення інформаційних процесів. У багатьох країнах настала ера «золотих комірців». Вона характеризується оснащенням службовців сучасними технічними засобами управління і обробки інформації на базі персональних комп'ютерів і сучасного програмного забезпечення.

У сучасних умовах комп'ютерні інформаційні технології є основою управлінської діяльності фірми. Зростання об'ємів інформації в контурі управління, потреба в прискоренні і складніших способах її переробки приводить до необхідності автоматизованої обробки інформації, тобто впровадження комп'ютерних інформаційних технологій. Керівникам щодня доводиться ухвалювати рішення різної складності в умовах великої невизначеності, пов'язаної із змінною ситуацією і недостатністю інформації. Широке застосування персональних комп'ютерів, що забезпечують полегшений доступ до баз даних і баз знань, використання інтелектуальних технологій і систем дають фахівцеві реальні можливості для виконання аналітичних, прогнозних функцій підготовки управлінських рішень в сучасному технологічному режимі обробки інформації.

Використання сучасних досягнень в області комп'ютерних технологій у сфері управління забезпечує підвищення якості економічної інформації, її точності, об'єктивності, оперативності і, як наслідок цього, можливості ухвалення вчасних управлінських рішень на основі *моделювання, аналізу і прогнозування*. Таким чином, одна з головних переваг використання сучасних комп'ютерних технологій - можливість оперативно і оптимально управляти підприємством.

Історія виникнення інформаційних технологій йде своїм корінням в глибину старовину. Появу простих інформаційних технологій можна віднести до виникнення писемності (5-6 тисячоліть тому), яка дозволила реалізувати повний набір процесів циркуляції і переробки інформації: її збір, передачу, переробку, зберігання і доведення. Ці можливості відкрила фіксація інформації на матеріальних носіях (на камені, кістках, дереві, глині, папірусі, пергаменті, шовку, папері). Подальший розвиток інформаційних технологій пов'язаний, головним чином, з появою нових технічних засобів переробки інформації, які визначають рівень розвитку інформаційної технології. Поліпшення управління є найважливішим чинником підвищення ефективності. Вдосконалення форм і методів управління відбувається на

основі досягнень науково-технічного прогресу і вивчення законів, методів і способів накопичення, обробки і передачі інформації.

Існують різні точки зору на періодизацію історії інформаційних технологій. Так, виділяють наступні етапи в історії розвитку інформаційних технологій. Перший етап – це ручна технологія збору і обробки інформації, що панувала до другої половини XIX ст. Основними інструментами у той час були перо, чорнила і прості рахункові апарати, а для прискорення передачі інформації використовувалися вогнища, кур'єри, поштовий зв'язок. Продуктивність інформаційної обробки була украй низькою. Потужним поштовхом до розвитку інформаційних технологій на цьому етапі з'явилося книгодрукування (середина XV ст.), що дозволило тиражувати інформацію і відкрило еру паперової інформаційної технології, що займає і в даний час значне місце.

Наприкінці XIX ст. розпочався етап механічної технології, пов'язаний з технічною революцією. До цього часу відноситься створення друкувальної машинки, телеграфу, телефону, радіо, модернізація системи суспільної пошти, які істотно змінили технологію обробки і передачі інформації, організаційну структуру підприємств. Інформація перетворилася на рушійну силу технічного, соціального і економічного прогресу.

З появою електронних друкувальних машинок, копіювальних машин і диктофонів в 40-60-х рр. XX ст. пов'язується етап електронної технології. Інструменти цієї технології поліпшили установчу діяльність за рахунок підвищення якості, кількості і швидкості обробки документів. Багато сучасних установ базуються на подібній технології.

В середині XX ст. був винайдений комп'ютер – універсальна електронно-обчислювальна машина, що увібрала в себе все краще, що було створене багатьма поколіннями учених, теоретиків і практиків. Вперше був створений спосіб запису і довготривалого зберігання формалізованих знань, при якому ці знання могли безпосередньо впливати на роботу виробничого устаткування. Масове виробництво електронно-обчислювальних машин в 60-х рр. і їх широке проникнення у всі сфери діяльності є початком зародження комп'ютерної інформаційної технології, направленої на збільшення ступеня автоматизації всіх інформаційних операцій і, отже, прискорення науково-технічного прогресу.

В ході еволюції комп'ютерів і пов'язаних з ними технологій їх використання переміщається від виконання елементарних операцій в область ухвалення рішень, інтелектуальних і наукомістких проблем.

У першій бухгалтерській ері (з кінця 50-х до початку 60-х років XX ст.) експлуатувалися комп'ютери першого (20 тисяч операцій в секунду) і другого (до 500 тисяч операцій в секунду) покоління для вирішення окремих

розрахункових (інженерних) і найбільш простих, але трудомістких економічних завдань (наприклад, матеріального обліку), тобто застосовувалася часткова електронна обробка даних.

У другій експлуатаційній ері (до середини 70-х років ХХ ст.) використовується електронна обробка даних на всіх етапах управління діяльністю підприємства, тобто здійснюється перехід до автоматизованих систем управління (матеріально-технічного постачання, руху товару, контролю запасів і транспортних перевезень, обліку реалізації готової продукції і т. п.).

Особливо успішно комп'ютерна інформаційна технологія почала упроваджуватися з середини 70-х рр. ХХ ст, коли були створені перші персональні комп'ютери, що дозволили формалізувати і зробити широкодоступними для автоматизації багато процесів людської діяльності, що важко формалізуються. Поява персонального комп'ютера з достатньо широкими для непрофесійного користувача можливостями провела інформаційну революцію і знаменувала початок безпаперової фази розвитку інформаційних технологій. Абсолютно неймовірно і виключно швидко його розповсюдження і розвиток саме як інструментального засобу накопичення, перетворення і передачі інформації дозволили комп'ютерним інформаційним технологіям упровадитися практично у всі області людської діяльності.

З появою персональних комп'ютерів пов'язується третя інформаційна ера (з кінця 70-х до початку 80-х років ХХ століття), прикладним акцентом в якій було використання інформації *для ухвалення рішень*. До 80-х років комп'ютерні інформаційні технології розвинулися в *системи підтримки ухвалення рішень*, що виготовлялися за замовленням, і ранні стратегічні *системи планування*.

З 80-х років ХХ століття істотним чином почала змінюватися концепція побудови технологічних процесів переробки інформації. Від централізованої технології обробки інформації, яка була характерною при застосуванні великих ЕОМ, почали переходити до децентралізованої і розподіленої технології з використанням персональних комп'ютерів на робочих місцях користувачів.

У четвертій ері, що розпочалася в середині 80-х років ХХ ст., у взаємозв'язаному суспільстві величезну увагу почали приділяти можливостям отримання стратегічних і конкурентних переваг за рахунок використання комп'ютерних інформаційних технологій як елементу ділової стратегії.

Зараз настала п'ята ера – глобальне взаємозв'язане суспільство, в якому фірми можуть використовувати комп'ютерні інформаційні технології, щоб працювати зі своїми діловими партнерами і клієнтами у світовому масштабі. Інформація *стає ресурсом нарівні з матеріалами, енергією і*

капіталом. Вона перетворюється на один з найбільш цінних за змістом і масових за формою продуктів цивілізації, споживачем якої стає все людство. Машинна інтуїція (експертні системи) перетворюється на продуктивну силу, а штучний інтелект дозволяє вирішувати якісно нові завдання технічного прогресу.

Історія розвитку механізму інформаційної взаємодії між людьми і між людиною і машиною дає підставу для розуміння інформаційних технологій як єдиної інтеграційної системи розвитку всіх галузей знань, етапи якої в основному збігаються з періодами становлення природознавства і з ранішими за часом періодами накопичення знань в суспільстві.

Сьогодні комп'ютери і відповідне програмне забезпечення радикально змінюють методи і технологію обробки інформації. Комп'ютери забезпечують пристрої для зберігання і обробки інформації. Комп'ютерні програми, або програмне забезпечення, є наборами керівництва по обслуговуванню, яке управляє роботою комп'ютерів. Комп'ютери і програми для них – це інструментальні засоби і матеріали сучасних інформаційних технологій, але вони самі по собі *не можуть проводити* потрібну для організації *інформацію*.

Основна мета комп'ютерної інформаційної технології – отримувати за допомогою переробки первинних даних інформацію нової якості, на основі якої виробляються оптимальні управлінські рішення.

Це досягається за рахунок інтеграції інформації, забезпечення її актуальності і несуперечності, використання сучасних технічних засобів для впровадження і функціонування якісно нових форм інформаційної підтримки діяльності апарату управління. Комп'ютерна інформаційна технологія дозволяє менеджерів справлятися з істотним збільшенням об'ємів інформації, що переробляється, і веде до скорочення термінів її переробки.

Можливі різні схеми класифікації комп'ютерних інформаційних технологій. Кожна з них будується на певних класифікаційних ознаках.

Основними класифікаційними ознаками комп'ютерних інформаційних технологій є:

- ступінь централізації технологічного процесу;
- тип наочної області;
- ступінь обхвату завдань управління;
- клас технологічних операцій, що реалізуються;
- тип призначеного для користувача інтерфейсу;
- спосіб побудови мережі.

За ступенем централізації технологічного процесу комп'ютерні інформаційні технології в системах управління поділяються на централізовані, децентралізовані і комбіновані.

Централізовані технології характеризуються тим, що обробка інформації і вирішення основних функціональних завдань економічного об'єкту проводяться в інформаційному Центрі Обробки Даних (ЦОД), організованому на підприємстві обчислювальної мережі або в галузевому або в територіальному інформаційно-обчислювальному центрі.

Децентралізовані технології ґрунтуються на локальному застосуванні засобів обчислювальної техніки, встановлених на робочих місцях користувачів для вирішення конкретного завдання фахівця. Вони не мають централізованого автоматизованого сховища даних, але забезпечують користувачів засобами комунікації для обміну даними між вузлами мережі.

Комбіновані технології характеризуються інтеграцією процесів вирішення функціональних завдань на місцях з використанням спільних баз даних і концентрацією всієї інформації системи в автоматизованому банку даних.

Тип наочної області виділяє функціональні класи завдань відповідних підприємств і організацій, вирішення яких проводиться з використанням сучасної комп'ютерної інформаційної технології. До них відносяться завдання бухгалтерського обліку і аудиту, банківської сфери, страхової і фінансової діяльності і ін.

У сучасних умовах інформатизації і комп'ютеризації швидкий розвиток економічних систем вимагає нового підходу до обробки інформації і процесу ухвалення рішень. Діяльність організаційних систем, що супроводиться процесом генерації великого об'єму інформації і вимагає оперативної обробки для ухвалення рішень, може ускладнюватися сукупністю чинників, таких як передача неповної, неточної або помилкової інформації, непостійність характеристик і умов функціонування самих систем, наявність людського чинника, тобто участь людей, що володіють свободою дії.

Що ж до такого важливого напрямку у використанні сучасних комп'ютерних технологій, як аналіз діяльності фірми, сприяння в ухваленні управлінських рішень, то слід зазначити, що системи такого роду є, безумовно, актуальними, і саме в їх розробці, впровадженні і постійному розвитку вбачається сьогодні головний стимул розвитку економіки.

Застосування комп'ютерної технології в економічному процесі покращує керованість (прискорюються цикли управління), забезпечує зростання інтелектуальних можливостей всієї системи управління, покращує якість управління за рахунок системи використання банків даних, експертних систем і прогнозу ухвалюваних рішень.

Основним напрямом діяльності будь-якої організації є процес розробки і просування свого продукту на ринок. Процес розробки нового

продукту є найбільш відповідальним, тому саме цей етап потребує створення програмного продукту, що забезпечує інформаційну підтримку процесу ухвалення рішення. Для ухвалення рішення найбільш важливим є: по-перше, швидкість процесу ухвалення рішення; по-друге, обґрунтованість вибору рішення.

Найбільший вплив сучасні комп'ютерні технології мають на просування продукту (розповсюдження і продажі). Перш за все, це стосується можливості формування нових маркетингових каналів просування і збуту продукту. Так, в області реклами широкого поширення набула пряма розсилка інформаційних повідомлень електронною поштою (direct-mail). Останніми роками кожне підприємство прагне створити свій власний сайт в Інтернет.

В області менеджменту сталися також кардинальні зміни. Сучасний рівень розвитку бізнесу і жорстка конкуренція в цій області додають особливу важливість інформаційним системам управління підприємством.

Сучасні комп'ютерні інформаційні технології здатні кардинально змінювати методичну, інформаційну і технологічну складові управлінських процесів і здійснювати їх якісно новому, ефективнішому рівні. Проте, в даний час все ще існує сукупність об'єктивних чинників, що надають стримуючу дію на темпи їх впровадження в нашій країні, до яких можна віднести наступні: економічну нестабільність, «пропуски» в законодавчому забезпеченні, недостатність освіти управлінських кадрів у сфері інформаційних технологій, дефіцит фахівців в області інформації, недостатнє державне фінансування науково-дослідних і практичних розробок, пов'язаних з новими інформаційними технологіями (НІТ). Разом з перерахованими проблемами існує ще маса інших проблем, таких як недостатня компетентність як керівництва всіх рівнів управління підприємством, так і рядових працівників управлінської сфери відносно питань автоматизації (впровадження нових інформаційних систем і технологій); прихильність до традиційного підходу у сфері управління. І хоча багато керівників і фахівців розуміють, що час вимагає нових підходів до реалізації більшості завдань, але втілювати їх на практиці не поспішають. Це стосується як типових завдань, так і принципово нових завдань. Ще одна проблема – аналіз існуючої системи управління на підприємстві. Не дивлячись на те, що дане питання достатньо детально висвітлене в літературі і необхідність такого роду досліджень вже неодноразово доводилася вітчизняними і зарубіжними ученими, підприємства вельми неохоче погоджуються на проведення подібної роботи, причому тільки тоді, коли уникнути її вже не можна. Наприклад, здійснення таких робіт є обов'язковою умовою впровадження корпоративних систем управління. І, як наслідок, необхідність організаційної перебудови

підприємства, як у виробничому, так і в управлінському секторі. Хоча більшість пакетів програм передбачають настроювання на існуючу організаційну структуру, проте не можна стверджувати, що «приспособлення» пакету під потреби існуючої організації є раціональним.

Поступовість впровадження передбачає реалізацію первинної, базової системи, розширення круга користувачів, збільшення числа прикладних завдань, інтеграцію.

Результат впровадження комп'ютерних інформаційних технологій – заощадження часу фахівців, позбавлення їх від рутинних операцій, переведення їх діяльності в творче русло. Єдиний спосіб отримання відчутного економічного ефекту від заощадження часу – переорієнтація цього часу на досягнення конкретних цілей даного підрозділу, визначуваних спільною стратегією фірми.

Комп'ютерні інформаційні технології нестримно еволюціонуватимуть і далі, даючи поштовх в розвитку науки економічних і управлінських інформаційних технологій і набуваючи все більшої значущості як найважливішого інструменту науково-технічного і соціально-економічного розвитку суспільства.

9.2. Інтегровані системи управління підприємством (ERP-системи)

➤ *Інтегрована економічна інформаційна система управління підприємством (ERP-система)* – людино-машинна система, в якій за допомогою економіко-математичних методів сучасних засобів збору, передачі і обробки, економічної інформації вирішуються завдання з управління виробничими процесами.

Елементами цієї системи є людина, технічні засоби, економіко-математичні методи, сукупність завдань і функцій управління представлених у формалізованому вигляді, програмні засоби.

Проектування ERP-системи базується на сукупності наукових положень, до яких належить:

- використання системного підходу;
- використання принципів по створенню систем, які сформулював академік В.М. Глушков.

Системний підхід – метод, який базується на дотриманні сукупності правил при проектуванні всієї системи та її окремих елементів.

Згідно цьому методу весь процес проектування розбивають на два етапи:

- макропроекування;
- мікропроекування.

Перший етап передбачає розробку принципових положень по структурі і функціональному забезпеченню системи.

До них належать:

- формулювання проблеми, яка розбивається на певну мету системи, опис чинників, що діють на систему, опис зв'язків, що діють з іншими системами;
- визначення варіантів побудови системи;
- аналіз цих варіантів;
- побудова адекватного математичного опису;
- синтез оптимального варіанту системи.

Мета, способи, адекватний математичний опис, програмні засоби, кожен з елементів підкоряється спільній меті.

Принципи, яким доцільно дотримуватися для розробки систем:

- принципи нових завдань (ІС залучає до себе нові завдання, розширення системи);
- принцип безперервного розвитку системи;
- принцип однократного введення інформації і багатократного використання (введення інформації трудомісткий процес, одні і ті ж дані повинні вводиться один раз, але можуть по-різному використовуватися);
- проєктовані системи повинні передбачати не лише обробку інформації, але і організацію збору, передачі даних по каналах зв'язку;
- принцип першого керівника (всією системою повинна керувати одна людина, яка відповідає за її функціонування в майбутньому);
- завдання розробки бази даних, призначеної для зберігання інформації;
- завдання розробки графічного інтерфейсу користувача клієнтських застосувань.

Основні поняття виробничого менеджменту (у тому числі і термін «ERP») можна вважати цілком усталеними. У цій області визнаним стандартом служить термінологія Американської асоціації по управлінню запасами і виробництвом (American Production and Inventory Control Society, APICS). Основні терміни і визначення наводяться в Словнику APICS, який регулярно оновлюється по мірі розвитку теорії і практики управління. Саме у цьому виданні міститься якнайповніше і точніше визначення ERP-системи.

Відповідно до Словника APICS, термін «ERP-система» (Enterprise Resource Planning — Управління ресурсами підприємства) може вживатися в двох значеннях.

По-перше, це — інформаційна система для ідентифікації і планування всіх ресурсів підприємства, які необхідні для здійснення продажів,

виробництва, закупівель і обліку в процесі виконання клієнтських замовлень.

По-друге (у більш загальному контексті), це — *методологія ефективного планування і управління всіма ресурсами підприємства*, які необхідні для здійснення продажів, виробництва, закупівель і обліку при виконання замовлень клієнтів в сферах виробництва, дистрибуції і надання послуг.

Таким чином, термін ERP може означати не лише інформаційну систему, але і відповідну методологію управління, що реалізовується і підтримується цією інформаційною системою.

Планування потреб в матеріалах (MRP). У 60-і роки XX століття був розроблений метод MRP (Material Requirements Planning — Планування потреб в матеріалах), що дозволяє усунути недоліки простих систем управління запасами. MRP базується на даних основного виробничого плану, при складанні якого за вихідну точку береться очікуваний попит на готову продукцію або інші виникаючі потреби в матеріалах. Таким чином, прогресивність методу MRP пояснювалася його орієнтацією на майбутні потреби і можливістю формування замовлень на поповнення запасів саме тоді, коли це дійсно необхідно і лише на реально необхідні об'єми.

Недолік методу MRP полягає в тому, що він виходить з принципу необмеженого завантаження, тобто ігнорує обмеженість виробничих потужностей. Але, як відомо, далеко не всі ресурси підприємства можуть розглядатися як необмежені. Тому, не дивлячись на свою прогресивність, метод MRP опинився не здатний вирішити завдання взаємозв'язку планування матеріальних ресурсів і потужностей.

Планування виробничих ресурсів (MRP II). Концепцію MRP II (Manufacturing Resource Planning — Планування виробничих ресурсів) часто називають методологічною основою ERP-систем. Це метод планування всіх ресурсів виробничого підприємства, включаючи планування в натуральних одиницях, фінансове планування у вартісному вираженні, а також елементи моделювання виробничих ситуацій. MRP II є результатом розвитку методу MRP.

На відміну від MRP, що розглядає виробничі потужності як необмежені, MRP II містить спеціальну функцію, що дозволяє погоджувати потреби в матеріалах з можливостями виробництва. Ця функція отримала назву CRP (Capacity Requirements Planning — Планування виробничих потужностей). Таким чином, MRP II є поєднанням планування по MRP з функцією CRP, включаючи управління складами, постачанням, продажами і виробництвом.

Вдосконалене планування (APS). Методологія APS (Advanced Planning and Scheduling — Вдосконалене планування) з'явилася в середині 90-х років і тому може вважатися однією з останніх розробок в теорії управління виробництвом. Цей метод включає дві частини: планування виробництва і постачання і диспетчеризацію виробництва.

Перша частина методу APS схожа на алгоритм MRP II. Істотна відзнака полягає в тому, що в системі APS узгодження матеріалів і потужностей відбувається не ітеративно, а синхронно, що різко скорочує час перепланування. Це особливо актуально для позаказного виробництва, а також у випадках жорсткої конкуренції в термінах виконання замовлення і необхідності точного дотримання цих термінів. Друга частина методу APS — диспетчеризація виробництва, з можливістю обліку різного роду обмежень, з елементами оптимізації.

Більшість сучасних ERP-систем побудовані за модульним принципом, що дає замовникові можливість вибору і впровадження лише тих модулів, які йому дійсно необхідні. Модулі різних ERP-систем можуть відрізнятися як по назвах, так і за змістом. Проте, є деякий набір функцій, який може вважатися за типовий для програмних продуктів класу ERP. Такими типовими функціями є:

- ведення конструкторських і технологічних специфікацій. Такі специфікації визначають склад кінцевого виробу, а також матеріальні ресурси і операції, необхідні для його виготовлення (включаючи маршрутизацію);
- управління попитом і формування планів продажів і виробництва. Ці функції призначені для прогнозу попиту і планування випуску продукції;
- планування потреб в матеріалах. Дозволяють визначити об'єми різних видів матеріальних ресурсів (сировини, матеріалів, комплектуючих), необхідних для виконання виробничого плану, а також терміни постачань, розміри партій і т.д.;
- управління запасами і закупівельною діяльністю. Дозволяють організувати ведення договорів, реалізувати схему централізованих закупівель, забезпечити облік і оптимізацію складських запасів і т.п.;
- планування виробничих потужностей. Ця функція дозволяє контролювати наявність доступних потужностей і планувати їх завантаження. Включає укрупнене планування потужностей (для оцінки реалістичності виробничих планів) і детальніше планування, аж до окремих робочих центрів;
- фінансові функції. До цієї групи входять функції фінансового обліку, управлінського обліку, а також оперативного управління фінансами;
- функції управління проектами. Забезпечують планування завдань проекту і ресурсів, необхідних для їх реалізації.

➤ *ERP-системи* – набір інтегрованих застосувань, які комплексно, в

єдиному інформаційному просторі підтримують всі основні аспекти управлінської діяльності підприємств – планування ресурсів (фінансових, людських, матеріальних) для виробництва товарів (послуг), оперативне управління виконанням планів (включаючи постачання, збут, ведення договорів), всі види обліку, аналіз результатів господарської діяльності.

Серед вимог, що пред'являються до ERP-систем: централізація даних в єдиній базі, близький до реального часу режим роботи, збереження спільної моделі управління для підприємств будь-яких галузей, підтримка територіально-розподілених структур, робота на широкому колі апаратно-програмних платформ і СУБД.

Є три основні причини застосування ERP.

- *Інтеграція фінансових даних.* Намагаючись оцінити роботу компанії в цілому, керівник стикається з різними уявленнями співробітників про реальне положення справ. Відділ фінансів має в своєму розпорядженні одні дані про прибуток, відділ збуту - інші. Є декілька версій про те, який вклад кожного підрозділу і працівника в цей прибуток. ERP створює єдину картину реальності, яка незаперечна, тому що всі використовують одну систему.

- *Стандартизація процесів виробництва.* У компаніях - виробниках (особливо в тих, хто захоплюється злиттям і поглинаннями) нерідко декілька підрозділів виконують одні і ті ж функції, використовуючи різні методи і комп'ютерні системи. Стандартизація цих процесів і застосування єдиної системи збереже час, підвищить продуктивність і понизить чисельність працівників.

- *Стандартизація кадрової інформації.* У фірмах з великою кількістю підрозділів відділ кадрів часто не має простого способу знаходити дані про відпрацьований час, допомоги і послуги для працівників. ERP дасть такий спосіб.

За спостереженнями фахівців, є три основні способи впровадження ERP.

- *«Великий вибух».* Найважчий спосіб: компанія разом відмовляється від колишніх систем і упроваджує єдине ERP-рішення. Зараз на цей крок йдуть не багато компаній, оскільки «великий вибух» вимагає повної мобілізації і змін в компанії в цілому.

Щоб спонукати весь персонал співробітничати і прийняти нову систему, потрібне колосальне зусилля, оскільки у неї поки немає прихильників. Ніхто в компанії не вмів нею користуватися, і ніхто не упевнений, чи буде вона працювати. У багатьох відділах є комп'ютерні системи, відладжені для певного виду діяльності. В більшості випадків ERP не запропонує ні нових функцій, ні звичної зручності, яку давала колишня

система. Часто знижується швидкість роботи, оскільки система обслуговує всю компанію, а не один відділ.

- *Стратегія «франчайзингу».* Цей метод личить для крупних або багатопрофільних компаній, підрозділи яких використовують не пересічні процеси. Незалежні ERP-системи встановлюють в кожному підрозділі, сполучаючи спільні процеси (наприклад, фінансовий облік) по всьому підприємству.

Цей метод впровадження став загальноприйнятим. Кожен підрозділ отримує свій «екземпляр» ERP: окрему систему і базу даних. Системи зв'язані один з одним тільки для обміну інформацією, необхідного, якщо компанії потрібний звіт по всіх підрозділах або по найбільш спільних процесах.

Як правило, таке впровадження починається з демонстрації або «експериментальної» інсталяції в найбільш відкритому підрозділі, невдалий досвід якого не знищить компанію. Коли команда запусить програмне забезпечення і усуне недоліки, вона розпочне «просувати» систему в інші підрозділи, використовуючи перше впровадження як довідковий посібник.

- *Впровадження «Частинами».* При використанні цього методу увага приділяється небагатьом основним процесам, які містяться, наприклад, у фінансовому модулі ERP-системи. Цей метод зазвичай використовують малі компанії, які хочуть ознайомитися і протестувати ERP.

Мета такого впровадження - щонайшвидше запустити систему, уникнувши складнощів реінжиніринга. Небагато хто з компаній, які встановили ERP в такий спосіб, може похвалитися великою віддачею. Провести реінжиніринг процесів після такого впровадження складніше, оскільки мало хто в компанії зміг відчути які-небудь переваги.

ERP-системи – останнє досягнення в еволюції автоматизованих інтегрованих систем управління підприємством. Їх безпосередніми попередниками були MRP II-системи (Manufacture Resources Planning - планування виробничих ресурсів). Спочатку робота MRP II-систем зводилася до планування матеріальних, потужносних і фінансових ресурсів, необхідних для здійснення виробничої діяльності підприємства. Ці системи розвивалися, в них вводилися нові функціональні можливості. Поступово виникло розуміння того, як пристосувати подібні системи для планування і управління бізнесом всього підприємства і навіть багатопрофільної корпорації. Системи, що реалізували ці ідеї, отримали назву ERP-систем. Відмінності між ERP- і MRP II-системами видно вже з назв: з одного боку, планування ресурсів промислового підприємства/корпорації (Enterprise Resources Planning), з іншого – планування ресурсів виробництва (Manufacture Resources Planning).

Відмінності між двома типами систем (MRP II і ERP) не формалізовані, але існують.

Впровадження ERP-системи або сучасної MRPII-системи надає підприємству наступні переваги.

1. Системами класу MRPII/ERP є інтегровані інформаційні системи управління. Це означає, що:

- системи не пов'язані з виробничим процесом безпосередньо, вони не є автоматизованими системами управління технологічними процесами, але мають справу з моделлю технологічного процесу;

- їх робота полягає в поліпшенні діяльності підприємства, оптимізації матеріальних і фінансових потоків на основі необхідної інформації, що вводить на робочих місцях;

- у одній системі охоплюється планування і управління всією діяльністю виробничого підприємства, починаючи від закупівлі сировини і закінчуючи відвантаженням товару споживачеві;

- інформація вводить в систему тільки один раз в тому підрозділі, де вона виникає, зберігається в одному місці, і багато разів використовується всіма зацікавленими підрозділами.

Як наслідок, MRP II / ERP-системи дозволяють досягти узгодженості роботи різних підрозділів, знижуючи при цьому адміністративні витрати і усуваючи проблему інтеграції даних для різних застосувань. Ці системи є інструментом підвищення ефективності управління, ухвалення правильних стратегічних і тактичних рішень на основі своєчасної і достовірної інформації, що видається комп'ютером.

2. Використання MRP II / ERP-систем дозволяє досягти конкурентних переваг за рахунок оптимізації бізнес-процесів підприємства і зниження витрат.

Ці системи створювалися якраз для управління собівартістю продукції і досягнення за рахунок цього конкурентних вигод. Тому системи реалізують методи планування і управління, що дозволяють:

- регулювати кількість запасів, усуваючи їх дефіцит і залежування, і тим самим значно понизити омертвлені в запасах витрати і складські витрати;

- скоротити незавершене виробництво, оскільки виробництво планується тільки на основі попиту на кінцеву продукцію, при цьому виробничі роботи ініціюються, виходячи з терміну, до якого має бути виконане клієнтське замовлення;

- оцінювати здійснимість поданих замовлень з погляду потужностей, що є на підприємстві;

- скоротити витрати і час, що витрачаються на виготовлення продукції, за рахунок оптимізації бізнес-процесів;

- відстежувати фактичну продуктивність кожної виробничої одиниці і, порівнюючи її з плановою продуктивністю, оперативно вносити коректування до виробничих планів;
- в результаті зменшення циклу виробництва і циклу виконання замовлення гнучкіше реагувати на попит;
- поліпшити обслуговування клієнтів і замовників за рахунок своєчасного виконання поставок.

Реалізовані в MRP II / ERP-системах можливості гнучкого управління собівартістю продукції дозволяють отримувати вищий прибуток. Окрім цього, зниження собівартості означає можливість варіювати ринкову ціну продукції (у бік пониження), що є потужною перевагою в конкурентній боротьбі.

Залишається відзначити, що впровадження MRP II / ERP-системи конкурентом - це сигнал для ухвалення симетричного рішення, оскільки автоматизовані системи управління є дійсно потужним інструментом бізнесу і засобом виживання в нелегких ринкових умовах.

3. Упроваджена MRP II / ERP-система може допомогти компанії привабити інвестиції. MRP II / ERP-системи роблять бізнес компанії прозорішим, що підвищує довіру до нього з боку інвесторів. Це особливо актуально в умовах ринку, що стрімко розвивається.

Слід додати, що існують українські і російські розробки MRP II-систем («Галактика», «ІС», «Вітрило», «БЕСТ»), проте часто вибір вітчизняних підприємств падає на аналогічні західні системи. Мотивом такого рішення, як правило, є бажання привабити західні інвестиції: зазвичай іноземні акціонери, інвестори, консультанти вимагають, щоб на підприємстві була встановлена знайома ним система.

4. Останнім часом відмічене формуванням нової економіки, основним інфраструктурним елементом якої є мережа Інтернет. Виводячи частку свого бізнесу в Інтернет, компанії переслідують одночасно декілька цілей: від скорочення витрат до поліпшення обслуговування клієнтів і організації нового онлайн-каналу збуту. Існує багато схем електронної комерції як в секторі B2C (наприклад, створення Web-вітрини або Інтернет-магазину), так і в секторі B2B (наприклад, організація корпоративного порталу або участь у роботі віртуального торговельного майданчика - e-marketplaces).

Важливим чинником для ERP-систем є дотримання міжнародних стандартів технологічної безпеки.

Міжнародні стандарти, що підтримують випробування технологічної безпеки ІС. Основою розвитку процесу стандартизації забезпечення технологічної безпеки ERP-систем є формування раціонального по складу, структурі і рівням вимог комплексу нормативно-технічної

документації (НТД), що забезпечує нормативну основу створення і застосування ІС. Рівень введення і застосування міжнародних норм, правил і вимог в тих, де діють вітчизняні НТД, на сьогоднішній день дуже низький. У цих умовах реалізація проектів інформатизації без врахування вимог і рекомендацій міжнародних стандартів призводить до великих втрат технічних і фінансових ресурсів із-за несумісності технічних і програмних засобів.

Таблиця 1. Міжнародні стандарти, направлені на забезпечення технологічної безпеки

ISO 09126:1991.ИТ.	Оцінка програмного продукту. Характеристики якості і керівництво по їх застосуванню.
DOD-STD-2168.	Програма забезпечення якості оборотних програмних коштів.
ISO 09000-3:1991.	Спільне керівництво якістю і стандарти по забезпеченню якості. Ч.3: Керівні вказівки по застосуванню ISO 09001 при розробці, постачанні і обслуговуванні програмного забезпечення.
ISO 12207:1995.	Процеси життєвого циклу програмних засобів.
DOD-SND 2167 A:1988.	Розробка програмних засобів для систем військового призначення.
ISO 09646-1-6:1991.ИТ.ВОС.	Методологія і основи атестаційного тестування ВОС.
ANSI/IEEE 829-1983.	Документація при тестуванні програм.
ANSI/IEEE 1008-1986.	Тестування програмних модулів і компонент ІС.
ANSI/IEEE 1012-1986	Планування оцінки (verification) і підтвердження достовірності (validation) програмних засобів.

Ці важливі групи міжнародних стандартів регламентують:

- Показники якості програмних засобів;
- Життєвий цикл і технологічний процес створення критичних комплексів програм, що запобігають дефектам;
- Тестування програмних засобів для виявлення і усунення дефектів програм і даних;
- Випробування і сертифікацію програм для посвідчення досягнутої якості і безпеки їх функціонування.

Технологічна безпека функціонування ERP-систем при неумисних погрозах побічно підтримується ще багатьма десятками стандартів, які, в тому або іншому ступені, забезпечують життєвий цикл і технологію розробки

і супроводу, якість, тестування, випробування і сертифікацію ERP-системи, а також уніфікацію їх інтерфейсів з операційним і зовнішнім середовищем.

РОЗРАХУНОК ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ РОЗРОБКИ І ВПРОВАДЖЕННЯ АВТОМАТИЗОВАНОЇ ЕКОНОМІЧНОЇ СИСТЕМИ УПРАВЛІННЯ.

Мета розрахунку економічної ефективності – визначення економічної доцільності розробки і впровадження ERP-системи.

В процесі розрахунку зіставляються дані за витратами на створення системи і дані, що відображають зростання ефективності інформаційної системи. Система буде економічно ефективною, якщо показники ефективності (термін окупності, розрахунковий коефіцієнт ефективності капітальних вкладень) відповідають галузевим нормам ефективності капіталовкладень.

Основні показники ефективності автоматизованої обробки інформації:

- Зниження витрат на обробку інформації, річний приріст прибутку (річна економія);
- Річна економічна ефективність;
- Термін окупності;
- Розрахунковий коефіцієнт економічної ефективності.

Проектована система зазвичай дозволяє понизити трудомісткість обробки інформації. Тому річна економія (зростання прибутку) розраховується за рахунок економії праці. Інші показники ефективності (підвищення точності, оперативності розрахунків) враховуються на якісному рівні.

Методика розрахунку показників економічної ефективності.

Кількісний розрахунок економічної ефективності впровадження ІС виконується по наступній схемі:

Визначення річної економії

$$E = Z_{p.o} - Z_{EOM}$$

Де $Z_{p.o}$ – витрати ручної обробки інформації, грн.;

Z_{EOM} – витрати на автоматизовану обробку інформації, грн..

$$Z_{p.o} = M * I2(1 + K1 + K2)$$

де M – середньомісячна заробітна плата (на автоматизованій посаді), грн.;

$K1$ – коефіцієнт, що визначає розмір додаткової заробітної плати ($k1=0.45$);

$K2$ – коефіцієнт, що визначає розмір накладних витрат ($k2=0.4-0.7$)

$$Z_{EOM} = C_{EOM} * T_{EOM}$$

де C_{EOM} – вартість однієї машинної години роботи EOM, грн./година;

T_{EOM} – річний машинний час рішення задачі, година

$$C_{EOM} = (Z_{пчас} + A_{м.відр.} + Z_{ел.ен} + Z_{рем} + Z_{матер}) / F_{дійсн.},$$

де $Z_{\text{час}}$ - зарплата посадової особи при рішенні задачі на ЕОМ;

$A_{\text{м.відр.}}$ – амортизаційні відрахування основних і допоміжних технічних засобів, що беруть участь у вирішенні завдань, грн.;

$Z_{\text{ел.ен}}$ – витрати на електроенергію, грн.;

$Z_{\text{рем}}$ – витрати на ремонт устаткування, грн.;

$Z_{\text{матер}}$ – витрати на матеріали (картридж, папір), грн.;

$F_{\text{дійсн}}$ – дійсний фонд часу роботи ЕОМ, година

$A_{\text{м.відр.}} = (C_{\text{т.еом}} + C_{\text{т.доп.об.}}) * 0.25 + (C_{\text{т1}} \text{ кв.м} * \text{Пл}) * 0.07$

де $C_{\text{т.евм}}$ - вартість ЕМВ;

$C_{\text{т.доп.об.}}$ – вартість додаткового устаткування, грн.;

$C_{\text{т1}}$ кв.м – вартість одного квадратного метра майдану, зайнятого під обчислювальну техніку, грн.;

Пл – майдан, займаний ЕОМ і додатковим устаткуванням, м².

$(Z_{\text{ел.ен}} + Z_{\text{рем}} + Z_{\text{матер}}) = 0.03 (C_{\text{т.еом}} + C_{\text{т.доп.об.}})$

$F_{\text{дійсн}} = F_{\text{ном}} * (1 - I / 100\%)$

де $F_{\text{ном}}$ - номінальний річний фонд часу роботи ЕОМ, година;

I – коефіцієнт відображає простої устаткування у зв'язку з профілактикою і ремонтом ($I = 10 - 15\%$).

Одноразові витрати визначають по формулі:

$T_{\text{кап.вл.}} = (C_{\text{т.еом}} + C_{\text{т.доп.об.}}) + C_{\text{т.проект.роб.}} + (C_{\text{т.ПЗ}}) * 0.3$

де $C_{\text{т.еом}}$ - вартість ЕОМ, грн.;

$C_{\text{т.доп.об.}}$ - вартість додаткового устаткування, грн.;

$C_{\text{т.проект.роб.}}$ - вартість проектних робіт, грн.;

$C_{\text{т.ПЗ}}$ – вартість програмного забезпечення, грн..

Розрахунок економічної ефективності виконується по формулі:

$E_{\text{г}} = e - e_{\text{н}} * t_{\text{кап.вл.}}$,

де e – річна економія, грн.;

$e_{\text{н}}$ – нормативний коефіцієнт ефективності вкладення на обчислювальну техніку ($e_{\text{н}} = 0.32$);

$T_{\text{кап.вл.}}$ – одноразові витрати, грн.

Термін окупності одноразових витрат визначаються по формулі:

$T = t_{\text{кап.вл.}} / e$

Коефіцієнт економічної ефективності обчислюється за формулою:

$E_{\text{р}} = e / t_{\text{кап.вл.}}$

Система буде економічно ефективною, якщо показники ефективності (термін окупності, розрахунковий коефіцієнт ефективності капітальних вкладень) відповідають галузевим нормам ефективності капіталовкладень.

Ефективність кожної окремо узяті системи, як видно з формули, є функцією від річної економії засобів завдяки використанню ERP-системи. Для оцінки ефективності системи отриманий коефіцієнт економічної

ефективності порівнюють з аналогічними коефіцієнтами для інших систем або з нормативним значенням коефіцієнта по галузі.

На підставі результатів порівняння показників ефективності керівництво підприємства приймає рішення про початок процесу впровадження ERP-системи або про вибір упроваджуваної системи з можливих варіантів.

Питання і завдання для самостійної роботи

1. Що розуміється під комп'ютерними інформаційними технологіями?
2. Перерахуйте, що входить в комплекс технічних засобів, що забезпечують роботу системи.
3. Як називається програмно-апаратний комплекс, призначений для обробки документів і автоматизації роботи користувачів в системах управління?
4. Що є обов'язковими елементами проектного технологічного забезпечення ІТ?
5. Назвіть основні класифікаційні ознаки комп'ютерних інформаційних технологій.
6. Перерахуйте етапи формування рішень.
7. Що є інформаційним забезпеченням?
8. Як класифікується управлінська інформація?
9. Перерахуйте вимоги, що пред'являються до ERP-систем.
10. Назвіть принципи створення ІС і ІТ управління.
11. Перерахуйте форми інформаційних систем в управлінні підприємствами.
12. Що дозволяє досягти використання MRP II / ERP-систем?
13. Як розраховуються показники економічної ефективності впровадження ІС?

Список літератури

1. Беспалов В. М., Вакула А. Ю., Гострик А. М., Діордіца С. Г., Таракановський С. М., Тихонович Є. В. Інформатика для економістів: Навчальний посібник для студентів вищих навчальних закладів економічних спеціальностей. – К.: ЦУЛ, 2003. – 788 с.
2. Вовчак І. С. Інформаційні системи та комп'ютерні технології в менеджменті: Навч. посібн. –Тернопіль: Карт-бланш, 2001. – 354 с.
3. Інформаційні системи і технології в економіці / під ред. Пономаренка В. С. – К.: Академія, 2002. – 542 с.
4. Юринець В. Є., Крупка М. І., Юринець З. В. Автоматизовані інформаційні системи у фінансах: Навч. посібник. – Львів: ЛНУ імені Івана Франка, 2004. – 328 с.

Розділ X. ІНТЕЛЕКТУАЛЬНІ КОМП'ЮТЕРНІ СИСТЕМИ

Останніми роками методи і системи штучного інтелекту активно використовуються в різних сферах діяльності людини від інтелектуальних ігор (шахи) до інтелектуального аналізу даних в економіці. Більш того, проєкт комп'ютерів 5 покоління передбачає можливість створення єдиного інформаційного простору на основі онтологій – баз знань, що однозначно розуміються. Формування онтологій неможливе без використання методів штучного інтелекту, що пояснюють поняття «знання».

Учені і філософи з давніх часів намагалися зрозуміти, що таке інтелект людини і пояснити, на яких принципах засновано навчання, сприйняття, запам'ятовування і міркування. Багато алгоритмів і методів штучного інтелекту засновано на спостереженнях за природними процесами і спробах їх відтворення (копіювання). Наприклад, модель нейрона і нейронна мережа з'явилися в результаті спостереження за природними процесами, що протікають в нервовій системі живих організмів. Генетичні алгоритми з'явилися як результат спостереження за еволюцією популяцій живих організмів. Поява комп'ютерів відкрила нові можливості для дослідників, забезпечивши можливості перевірки ідей в області штучного інтелекту за допомогою проведення експериментів. Наприклад, описана Розенблаттом модель сприйняття була перевірена за допомогою комп'ютерного моделювання. Традиційно вважається, що поява штучного інтелекту як наукової дисципліни відноситься до 1956 р.

10.1.1. Природний і штучний інтелекти

Сучасні комп'ютери мають в своєму розпорядженні такі потужні ресурси, що можуть вирішувати складні задачі, які до недавнього часу були по плечу тільки людині. Основою людської діяльності є мислення. Для вирішення якої-небудь задачі людина, перш за все, ставить мету. Якщо мета достатньо складна, вона розбивається на підцілі, для досягнення яких вирішуються невеликі задачі. Для вирішення задачі використовуються факти і правила. Інтелект можна представити як сукупність фактів і способів їх застосування для досягнення мети.

У літературі зустрічається велике число визначень штучного інтелекту. Наведемо лише деякі з них.

➤ *Штучний інтелект* (визначення Е. Charniak і D. McDermot, 1985 р.) – дослідження розумових здібностей за допомогою використання обчислювальних моделей.

➤ *Штучний інтелект* (визначення Г.С. Поспелова, 1986 р.) – теорія евристичного пошуку і питання створення вирішувачів задач, що належать до розряду творчих або інтелектуальних.

➤ *Штучний інтелект* (визначення R.J. Schalkoff, 1990 р.) – область дослідження, яка намагається зрозуміти і моделювати розумну поведінку в термінах обчислювальних процесів.

Таким чином, *штучний інтелект* – програмна система, що імітує на комп'ютері мислення людини

При рішенні задачі людині необхідно звернутися до необхідних фактів і правил, так само треба зробити і програмі. Отримуючи результат рішення задачі, людина отримує нові факти. Для отримання результату в системі штучного інтелекту застосовується механізм логічного висновку.

Системи штучного інтелекту діляться на три класи:

- 1) *інтелектуальні інформаційні системи*, що дозволяють отримати відповідь на питання з деякої галузі знань на професійній мові користувача;
- 2) *розрахунково-логічні системи*, що розв'язують складні задачі в діалозі;
- 3) *експертні системи*, що дозволяють представляти знання в описовій формі і розв'язувати задачі з обґрунтуванням отриманого результату.

При реалізації цих систем вирішуються три основні проблеми штучного інтелекту: представлення знань, використання природної мови і моделювання на комп'ютері людських міркувань. Всі перераховані види систем засновані на знаннях, тому представлення знань є центральною проблемою штучного інтелекту. Знання можуть відображати деякий стан або деякий процес.

При роботі систем штучного інтелекту моделюються процеси, здійснювані природним інтелектом людини при розв'язанні задач. У основі людської діяльності лежить мислення. В процесі рішення задачі людиною можна виділити наступні етапи.

1. *Постановка мети. Мета* – це кінцевий результат, на який направлені розумові процеси людини. Головна мета розбивається на підцілі - локальні цілі. Мозок завжди зосереджений на меті, незалежно від того, виконує людина фізичну роботу чи розв'язує інтелектуальну задачу. Цілі досягаються за допомогою правил використання всіх відомих фактів.

Факт – це вислів про об'єкт, що містить його характеристики або зв'язки з іншими об'єктами.

Інтелект можна представити як сукупність фактів і способів їх застосування для досягнення мети.

2. *Спрощення*. Це вибір правильної реакції на конкретну ситуацію. Коли людський мозок приступає до вирішення навіть найпростішої задачі, для вибору необхідних дій в його розпорядженні є величезна кількість

інформації. У мозку існує складна система, що керується вибором необхідних для даного завдання відомостей. Без спрощення життя не лише б сповільнилося, а стало б неможливим.

3. *Логічний висновок.* Частина інтелекту, яка дозволяє витягувати нові факти, називається *механізмом висновку*.

Досягаючи мети, людина не лише приходиться до рішення поставленої перед ним задачі, але і набуває нових знань.

Процес рішення задачі людиною реалізований в *експертних системах*.

Під *експертною системою* розуміється комп'ютерна програма, що об'єднує можливості комп'ютера із знаннями і досвідом експерта в такій формі, що комп'ютер зможе запропонувати розумну раду або здійснити розумне розв'язання поставленої задачі. Система повинна на вимогу пояснювати хід своїх міркувань в зрозумілій для того, хто запитусь, формі.

Базова структура експертної системи представлена на рисунку 10.1.1.

База даних зберігається на диску, вона є безліччю значень для кожного факту. *База знань* – це сукупність правил для обробки фактів, вона створюється для представлення знань в конкретній наочній області (науковій або практичній сфері людської діяльності). *Користувальницький інтерфейс* призначений для правильної передачі відповідей користувачеві і введення від нього запитів. *Модуль набуття знань* використовується для отримання знань від експерта, він включає організацію діалогу, програму занесення правил в базу знань і програму занесення фактів в базу даних. *Модуль порад* включає також різні коментарі, що додаються до висновку, пояснює мотиви висновку і показує вірогідність правильності висновку або очікуваної події.

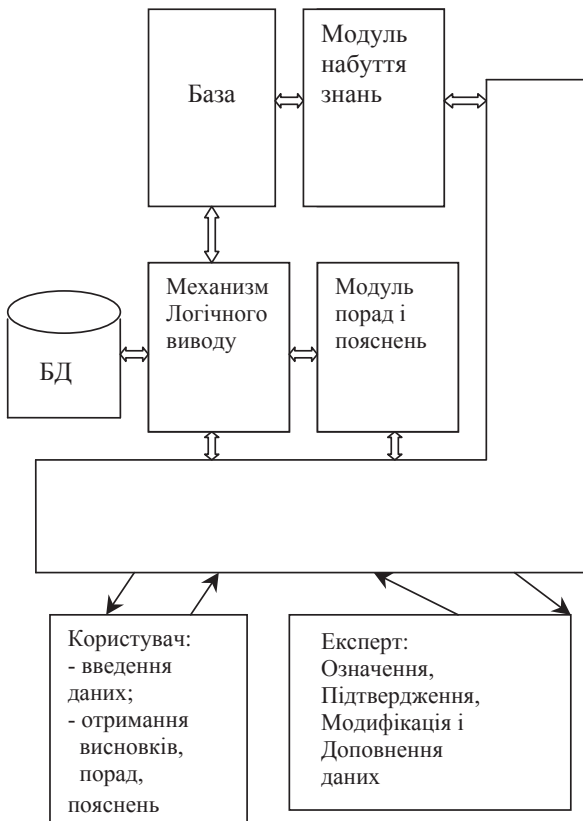


Рис. 10.1.1. Базова структура експертної системи

10.1.2. Способи представлення знань

За визначенням японського фахівця в області обробки знань С.Осугі: «Знання – це інформація, що зберігається в комп'ютері, формалізована відповідно до певних структурних правил, яку комп'ютер може автономно використовувати при логічних висновках».

При обробці знань найбільш фундаментальною і важливою проблемою є опис смислового вмісту завдань найширшого діапазону, а також наявність такої форми опису знань, яка гарантує, що обробка їх вмісту формальними правилами перетворення здійснюватиметься правильно.

Іншими словами, знання, представлені в комп'ютері, – це інформація з обмеженою семантикою (вузького значення).

Найбільш простим способом для представлення знань є система **продукційних правил (евристик)**, коли знання мають вид пропозиції «ЯКЩО ... ТО ...». Частина правила «ЯКЩО» називається посилкою, а частина «ТО» - висновком або дією. Посилка може складатися з декількох фактів або умов, висновок - це факт або дія. Тому правила можуть розумітися двояко:

ЯКЩО достеменні всі перераховані факти, ТО має місце даний факт або

ЯКЩО виконуються деякі умови, ТО виконати вказану дію.

У такому вигляді можна представити знання з найрізніших областей. Наприклад: ЯКЩО промінь виходить з вершини кута **I** ділить кут навпіл, ТО промінь - бісектриса;

ЯКЩО учень їде в школу **I** автобус зупинився біля школи, ТО треба вийти;

ЯКЩО точки **A**, **B** і **C** не лежать на одній прямій, ТО фігура **ABC** - трикутник;

ЯКЩО фрукт м'який **I** синій **I** має одну кісточку, ТО фрукт - слива.

Перше, друге і останнє правила містять декілька фактів в частині ЯКЩО, сполучених союзом “**I**”. Для отримання правильного висновку всі ці факти повинні мати місце, тобто бути присутніми одночасно.

Для представлення знань з деякої області наукової або практичної діяльності людини використовується також *семантична мережа* - мережева структура, що дозволяє зв'язати різні поняття, встановивши зв'язки між ними.

Головне призначення семантичної мережі - правильно відобразити *сенси* кожного поняття. Залежно від призначення, семантична мережа може містити різні зв'язки між об'єктами, а також об'єктами та їх атрибутами. Переходячи від одного елементу семантичної мережі до іншої, можна встановити зв'язок, на перший погляд, між не зв'язаними поняттями. Фрагменти семантичної мережі можуть бути представлені у вигляді евристик, але в правилі тільки констатується зв'язок між фактами, але не указується тип цього зв'язку. Фрагмент семантичної мережі для поняття «файл» представлений на рисунку 10.1.2. Зв'язки мають напрями. Так, зв'язок «поле має ім'я», відображений направленою стрілкою від «поля» до «ім'я» і називається «характеризується». При прогляданні семантичної мережі спрямованість зв'язку грає істотну роль, оскільки, якщо не вказати напрям, то виникає неоднозначність і незрозуміло, що чим характеризується: чи то поле запису своїм ім'ям, чи то ім'я - якимось полем.

Семантичні мережі бувають двох типів: інтенціональні і екстенціональні.

Інтенціональна семантична мережа є спільною структурою модельованої наочної області на основі абстрактних об'єктів і стосунків.

Екстенціональна семантична мережа описує конкретні знання про модельовані об'єкти і явища, вона є як би «фотографією» їх поточного стану.

У семантичній мережі використовуються три типи об'єктів: *поняття, події, властивості*.

Поняття – це відомості про абстрактні або фізичні об'єкти наочної області. Спільні поняття ідентифікуються як безліч значень атрибутів (параметрів або констант). Так, поняття «алгоритм» характеризується набором його властивостей; «працівник» - відомостями про посаду, заробітну плату, місце роботи; «студент» – назвою факультету, спеціальності, групи, оцінками по складених іспитах і тому подібне.

Властивості використовуються для уточнення понять, подій або інших властивостей. Стосовно понять властивості описують їх особливості або характеристики – колір, розміри, якість, а стосовно подій – тривалість, місце, час.

Події – це дії, які можуть внести зміни до наочної області. Результатом події є деякий новий стан наочної області.

При побудові інтелектуальних банків даних на основі семантичних мереж розділяються інтенціональні і екстенціональні знання: інтенціональні є основою бази знань, а екстенціональні – бази даних.

Занит до семантичної мережі – це набір фактів (або деяка ситуація), при описі яких використовуються змінні в позиціях значень атрибутів і стосунків. Пошук відповіді полягає в рішенні задачі ізоморфного вкладення

графа (або його підграфа) в семантичну мережу. Це можна представити як накладення одного дротяного каркаса на інший і пошук збігів структур. Якщо збігаються форми каркасів і дані у вузлах, то вважається, що в порожніх вузлах запиту необхідно помістити дані з вузлів основної мережі, з якою шукається збіг.

Третій спосіб представлення знань - створення *фреймів* (рамок, структур). У центр фрейма поміщається об'єкт, навколо нього, з відповідними зв'язками - його атрибути. Об'єднуючись через однотипні атрибути, фрейми також утворюють мережу. *Відзнакою фрейма від семантичної мережі* є те, що фрейм може також містити процедури (звернення до програм) обробки елементів фрейма. Теорія фреймів стосується розуміння того, що ми бачимо і чуємо. Фрейм, з погляду людського мислення, є одиницею представлення знань. Це як би заповнена у минулому рамка, деталі якої при необхідності уточнюються або замінюються. Коли людина отримує нові відомості, вона зіставляє їх з тим, що вже є, порівнюючи з деякими шаблонами, які існують в пам'яті. При цьому нові відомості класифікуються, включаються в ієрархічні структури. Фрейми дозволяють зберігати відомості про один об'єкт спільно, в той же час дозволяючи пов'язати їх з іншими об'єктами.

Фрейм – це структура даних для представлення стереотипних ситуацій. Фрейми відповідають поняттям, що відображають об'єкти, явища, характеристики предметної області. Фрейм містить як інформаційні, так і процедуральні елементи, а також може мати незаповнені частини (слоти). Процедуральні елементи дозволяють обробляти вузли фрейма. Наприклад, якщо представити фрейм поняття «байт», то процедуральними його елементами можуть бути процедури кодування символів і перетворення чисел з десяткової систему числення в двійкову. Таким чином, фрейм – це декларативно-процедуральна структура, сукупність описів і пов'язаних з ними процедур, доступ до яких здійснюється прямо з фрейма.

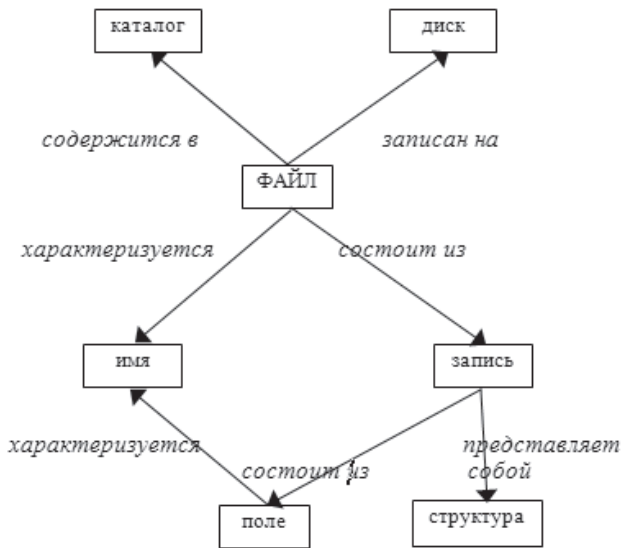


Рис.10.1.2. Фрагмент семантической сети

10.1.3. Механизмы логического вывода

Для отримання нових знань в системі продукційних правил, або обґрунтування деякого факту використовуються прямий і зворотний логічні висновки.

Прямий висновок служить для прогнозу і називається *індуктивним*. Він відповідає на питання «Що буде, якщо ... ?». Припустимо, система містить наступні правила.

- 1) ЯКЩО купити машину, ТО поїхати до моря;
- 2) ЯКЩО поїхати до моря, ТО загоряти;
- 3) ЯКЩО загоряти, ТО отримати сонячний удар.

При відповіді на питання такої системи «Що буде, якщо купити машину?», вона відповідає, що можна отримати сонячний удар. Така відповідь здивує будь-якого користувача, тому при розробці систем із

зворотним висновком необхідно ретельно складати правила. При прямому висновку поставлена мета (покупка машини) шукається в частині ЯКЩО правил системи. Коли факт знайдений, то розглядається частина ТО цього правила, в даному випадку - правила 1. Висновок цього правила (поїздка до моря) шукається в частині ЯКЩО іншого правила системи. Якщо такого правила немає, то як відповідь системи видається висновок першого правила (поїздка до моря). Якщо правило знайдене, то його висновок шукається знову в частині ЯКЩО наявних правил. Оскільки більше правил немає, кінцевим є висновок з третього правила. У даному прикладі перше правило містить маловірогідний висновок, тому результат роботи даної системи явно незадовільний. Щоб уникнути подібних висновків, в частині ТО указують дробове число, що показує вірогідність (або правдоподібність) цього висновку.

Зворотний висновок називається *дедуктивним* і обґрунтовує або пояснює наявні факти. Такий висновок використовує відомий детектив Шерлок Холмс при розкритті злочинів. Одна з перших експертних систем MYCIN, розроблена в Станфордському університеті (США) для діагностики бактерій в крові, використовує 500 евристичних правил. По ходу роботи програма запрошує нові дані і обґрунтовує свої міркування. При зворотному висновку наявний факт шукається в частині ТО правил. Якщо він знайдений, то факти з частини ЯКЩО цього правила використовуються для пошуку в частинах ТО нових правил. Якщо таких правил немає, то отримані факти розглядаються як причина початкового. Якщо вони знайдені, то пошук продовжується і останні отримані факти видаються як першопричина, що привела до появи факту-запиту. Якщо в прикладі, що розглянуто, з покупкою машини систему запитати «Чому стався сонячний удар?», то відповіддю буде «Тому що купили машину». Як обґрунтування буде виданий зворотний ланцюжок міркувань: «Сонячний удар стався через те, що ви загоряли, а загоряли тому, що поїхали до моря, а поїхали до моря тому, що купили машину».

Питання і завдання для самостійної роботи

1. Що таке інтелект людини?
2. Які існують основні класи систем штучного інтелекту?
3. Що роблять експертні системи?
4. Які проблеми вирішує штучний інтелект як науковий напрям?
5. Яка основна проблема штучного інтелекту?
6. Що таке евристика?

7. Як можуть розумітися проекційні правила?
8. Як здійснюється прямий логічний висновок?
9. Що дозволяє зробити і як проводиться зворотний висновок?
10. Для чого використовується семантична мережа?
11. Чим відрізняється фреймова мережа від семантичної?
12. Складіть продукційні правила, що дозволяють відрізнити птаха від літака.
13. Пригадайте детективну історію і складіть правила, які можна використовувати в системі із зворотним висновком.
14. Використовуючи рисунок 10.1.2, намалюйте фрейм поняття “байт”.
15. Напишіть правила класифікації компонентів комп'ютера.

Список літератури

1. Александров Е.А. Основы теории эвристических решений. Подход к изучению естественного и искусственного интеллекта. – М.: Сов. Радио, 1975. – 256 с.
2. Девятков В.В. Системы искусственного интеллекта: Учеб. пособие для вузов. – М.:Изд-во МГТУ им. Н.Э.Баумана, 2001. – 352 с.
3. Кричевский М.Л. Интеллектуальный анализ данных в менеджменте: учеб. пособие / СПбГУАП. СПб., 2005. 208 с.
4. Лорьер Ж.-Л. Системы искусственного интеллекта: Пер. с фр. М.: Мир, 1991.
5. Напалков А. В., Прагина Л. Л. Мозг человека и искусственный интеллект. — М.: Изд-во Моск. ун-та, 1985, 120 с.
6. Попов Э.В. Экспертные системы. Решение неформализованных задач в диалоге с ЭВМ. М.: Наука, 1987, 288 с.
7. Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы: Пер. с польск. И.Д. Рудинского. – М.: горячая линия – Телеком, 2006. – 452 с.

Розділ XI. Пакети прикладних програм

У цьому розділі представлений короткий опис математичного пакету Maple, що дозволяє проводити математичні обчислення різного ступеня складності і видавничої системи LATEX, використовуваної при підготовці науково-технічних друкарських видань високої якості, що містять велику кількість математичних формул і ілюстрацій

11.1. Математичний пакет Maple

Сьогодні неможливо уявити собі кваліфікованого ученого, інженера, конструктора, що не використовує при проведенні досліджень можливостей математичних пакетів, таких як Maple, MathCad, Matematica, MatLab, Statistica, MUPAD, Eureka, Derive і т.п. Яку систему серед такого різноманіття вибрати для розв'язання своїх задач? Без серйозного вивчення вбудованих спеціальних математичних функцій, графічних можливостей і функціональності мови програмування кожній з цих систем відповісти на поставлене питання неможливо.

Maple – потужний пакет для символьних, чисельних і графічних обчислень. Опис пакету проводиться на прикладі версій 5 і 9.01, але більшість описів, що наводяться, вірні і для раніших версій програми.

Математичний пакет Maple дозволяє розв'язувати велике число задач з математики, фізики і економіки різного рівня складності.

Основне вікно програми складається з рядка меню команд, рядків піктограм (графічних об'єктів, при натисненні на яких виконуються деякі обумовлені дії), нижче розташоване робоче вікно (лист), саме в ньому вводяться рядкові команди, що визначають необхідні математичні дії і, нарешті, в самому низу знаходиться рядок стану, що відображує різну допоміжну інформацію.

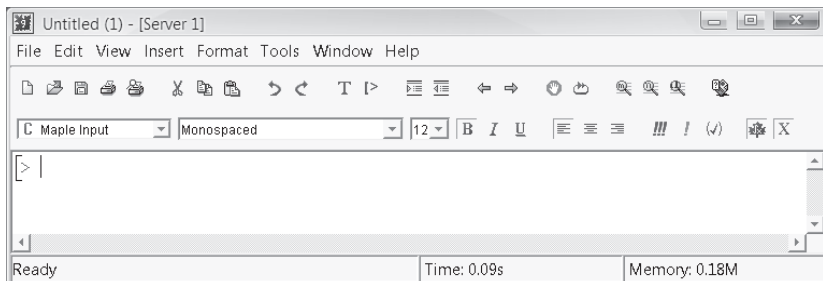


Рис.11.1.1. Користувальницький інтерфейс програми Maple 9.01

Синтаксис будь-якої команди Maple має вигляд:

> **Ім'я_команди (параметр1, параметр2,...);**

Кожну команду слід завершувати знаком “;”

Наведемо приклад деяких команд обчислення похідної, інтеграла і межі:

```

> diff(x^2+x^4-3*x+2, x);
                                     2 x + 4 x3 - 3
-----
> int(sin(x), x);
                                     -cos(x)
-----
> limit(sin(x)/x, x=0);
                                     1

```

Наведемо приклад команди виведення математичного вираження без обчислення:

```

> Int( exp(-x^3)/(x^2+1), x);
                                     ∫ (-x3)
                                     e
                                     ----- dx
                                     x2 + 1

```


11.1.1. Найпростіші обчислення і побудова графіків в пакеті Maple

Обчислення меж. Для обчислення межі функції f в точці $x=a$ використовуються наступні команди системи Maple:

```
> limit(f, x=a);  
> limit(f, x=a, dir);
```

Тут f – функція, x – ім'я змінної (наприклад, $x=\pi/2$ ($x = \pi/2$), $x=-\text{infinity}$ ($x = -\infty$)), dir – опція, що вказує на напрям пошуку межі (left – зліва, right – справа, real – в області дійсних значень, complex – на комплексній площині).

Функції $\text{Limit}(f, x=a)$; $\text{Limit}(f, x=a, \text{dir})$; - це інертні функції системи Maple, тобто призначені для запису меж в тексті. На рисунку 11.1.2 наведені приклади обчислення меж.

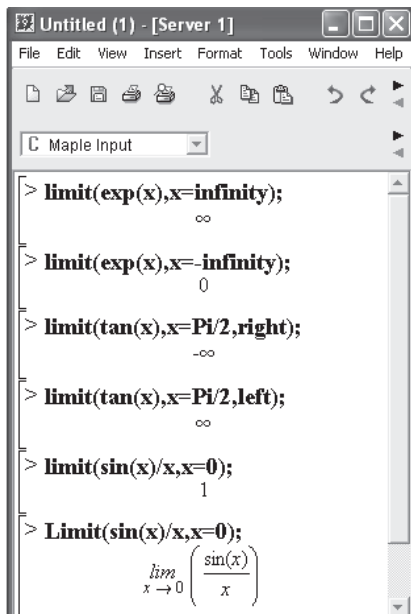


Рис.11.1.2. Приклади обчислення меж в пакеті Maple

Обчислення похідних. Для обчислення похідних в Maple слід використовувати наступні функції:

> **diff**($f(x_1, \dots, x_n), x_1, \dots, x_n$) - обчислення приватних похідних функції $f(x_1, \dots, x_n)$ по змінних x_1, \dots, x_n ;

Оператор \$ використовується для обчислення похідних вищих порядків. Для цього після імені відповідної змінної ставиться цей оператор і вказується порядок похідної.

> **Diff**($f(x_1, \dots, x_n), x_1, \dots, x_n$) – інертна форма використовується для запису похідних в тексті.

Приклади обчислення похідних наведені на рисунку 11.1.3.

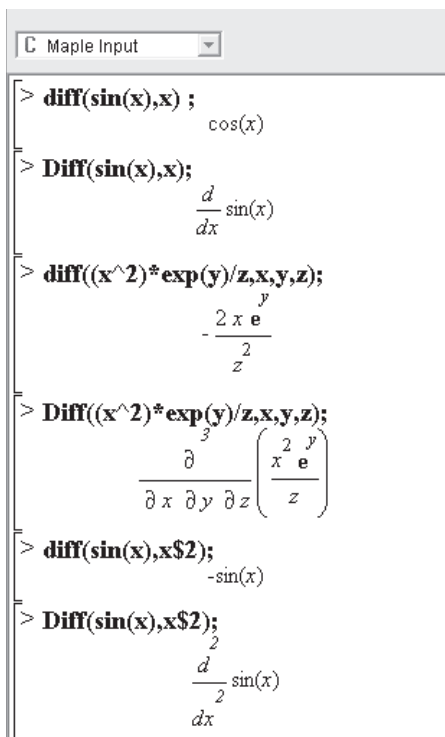


Рис.11.1.3. Приклади обчислення похідних в Maple

Побудова графіків функцій. Для побудови двовірних графіків використовується команда (функція) `plot`, синтаксис якої задається таким чином:

```
> plot(f,h,v);  
> plot(f,h,v,o);
```

Тут **f** – функція або функції, **h** – змінна з вказівкою області її зміни по горизонталі, **v** – опціонально задана змінна з вказівкою області її зміни по вертикалі, **o** – опція, що задає стиль оформлення графіка, наприклад, колір і товщину кривих.

Для побудови графіків поверхонь використовується команда (функція) `plot3d`, синтаксис якої задається таким чином:

```
> plot3d(f, x=a..b, y=c..d, p);  
> plot3d([f,g,h], x=a..b, y=c..d, p);
```

Тут **f, g, h** – функції, **a, b, c, d** – числові дійсні константи, **p** – опції. Приклад побудови поверхні наведений на рисунку 11.1.5.

```
> plot3d(x^2/18-y^2/8, x=-5..5, y=-2..2);
```

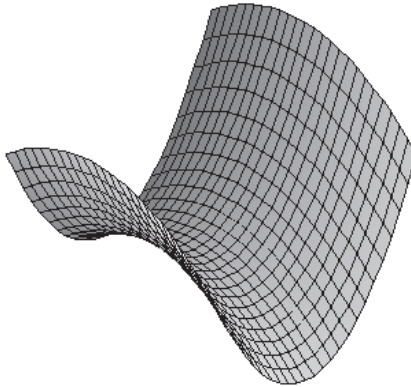


Рис.11.1.5. Приклад побудови поверхні в Maple

11.1.2. Використання спеціалізованих пакетів в Maple

При виклику тієї або іншої команди недостатньо знати ім'я команди, в деяких випадках слід завантажити бібліотеку або пакет, в якому описана команда, що нас цікавить. Зробити це можна таким чином:

> **with** (ім'я_пакета)

Опишемо деякі функціональні пакети і їх основні функції.

1. Пакет **linalg** – пакет лінійної алгебри. Матриця в Maple представляється у вигляді двовимірного масиву з рядками і стовпцями,

індексація яких починається з 1. Матриця (наприклад $\begin{pmatrix} -3 & 2 \\ 4 & 7 \end{pmatrix}$) може бути

введена двома способами, використовуючи або команду **matrix** з перерахуванням в квадратних дужках порядково елементів матриці (наприклад, `matrix([[-3,2], [4,7]])`) або команду **array** з перерахуванням діапазонів зміни індексів по рядках і стовпцях і з перерахуванням в квадратних дужках порядково елементів матриці (наприклад, `array(1..2,1..2,[[-3,2], [4,7]])`). У системі Maple є можливість інтерактивного введення елементів матриці за допомогою команди **entermatrix** (індексація рядків, індексація стовпців).

Деякі функції бібліотеки **linalg**:

addcol – формує лінійні комбінації із стовпців матриці;

addrow – формує лінійні комбінації з рядків матриці;

add(A,B,c,d) – повертає $c*A+d*B$, де A,B – матриці, c,d – константи;

adj(A) – повертає сполучену з A матрицю;

angle – обчислює кут між векторами;

augment – об'єднує дві або більше число матриць разом горизонтально;

band – створює стрічкову матрицю;

basis – знаходить базис для векторного простору;

BlockDiagonal – створює блок-діагональну матрицю;

charpoly(A,lambda) – повертає характеристичний поліном матриці A відносно змінної lambda;

col(A,i) – повертає i-тий стовпець матриці A;

coldim(A) – повертає число стовпців матриці A;

crossprod(U,V) – обчислює векторний добуток векторів U і V;

curl – обчислює ротор вектора;

delcols(A,i..j) – видаляє стовпці матриці A від i-го до j-го;

delrows(A,i..j) – видаляє рядки матриці A i-ой до j-ой;

diag – створює блок-діагональну матрицю;

dotprod(U,V) – обчислює скалярний добуток векторів U і V;

eigenvals – обчислює власні значення матриці;

eigenvects – обчислює власні вектори матриці;
equal – визначає, чи є дві матриці рівними;
exponential – створює експоненціальну матрицю;
fibonacci – матриця Фібоначчі.

Операції з матрицями

inverse (квадратна матриця) – створює зворотню матрицю;
transpose (матриця) – транспонування матриці;
det (квадратна матриця) – обчислює детермінант матриці;
 Приклад використання цих функцій наведено на рисунку 11.1.7.

matadd (матриця 1, матриця 2) – складання матриць або векторів;
multiply (матриця 1, матриця 2) – множення матриць;
 Приклад використання цих функцій наведено на рисунку 11.1.6.

```

C Maple Input Monospaced 14 B I
> with(linalg);
> A:=array(1..2,1..2,[[3,-17],[23,5]]);
      A := [ 3  -17 ]
           [23   5 ]
> B:=array(1..2,1..2,[[ -1,3],[-5,-17]]);
      B := [ -1   3 ]
           [-5 -17 ]
> matadd(A,B);
      [ 2  -14 ]
      [18  -12 ]
> multiply(A,B);
      [ 82  298 ]
      [-48 -16 ]
  
```

Рис.11.1.6. Приклади використання спеціалізованного пакету linalg в Maple

Хай A і B - матриці однакового розміру, тоді
 - складання цих двох матриць можна виконати з використанням команди

- > **evalm**(A+B);
- віднімання матриць
- > **evalm**(A-B);
- множення матриць
- > **evalm**(A&*B);
- ділення матриць
- > **evalm**(A/B);

```

> with(linalg);

> A:=array(1..2,1..2,[[3,-17],[23,5]]);
      A :=  $\begin{bmatrix} 3 & -17 \\ 23 & 5 \end{bmatrix}$ 

> Вычисление определителя матрицы A
> det(A);
      406

> Вычисление обратной матрицы
> inverse(A);
       $\begin{bmatrix} \frac{5}{406} & \frac{17}{406} \\ \frac{-23}{406} & \frac{3}{406} \end{bmatrix}$ 

> Транспонирование
> transpose(A);
       $\begin{bmatrix} 3 & 23 \\ -17 & 5 \end{bmatrix}$ 

```

Рис.11.1.7. Приклад використання функцій спеціалізованого пакета linalg в Maple

hermite(A) – Ермітова нормальна форма матриці A;
innerprod – обчислює векторний добуток;

jordan(A) – повертає матрицю A у формі Жордана;
Ludecomp – здійснює LU-композицію;
minor(A,i,j) – повертає мінор матриці A для елемента з індексами i і j;
norm(A) – норма матриці A або вектора A;
rank(A) – повертає ранг матриці A;
row(A,i) – повертає i-тий рядок матриці A;
rowdim(A) – повертає число рядків матриці A;
singval – обчислює сингулярне значення матриці;
singularvals – обчислює сингулярні значення матриці;
subvector – витягує вказаний вектор з матриці;
swapcol – міняє два стовпці в матриці;
swaprow – міняє два рядки в матриці;
trace(A) – повертає слід матриці A;
vectdim(A) – повертає розмірність вектора A;
vector(n,list) – завдання вектора з n елементами, де list – це перераховані в квадратних дужках через кому елементи вектора (наприклад, vector(3 [-17,2,15])).

Вирішення рівнянь і нерівностей

leastsqrs – вирішення рівнянь по методу найменших квадратів;
linsolve – вирішення систем лінійних рівнянь;
solve (рівняння, невідоме) – вирішення рівняння або нерівності
solve (система рівнянь, невідомі) – вирішення системи рівнянь або нерівностей
subs (вираження, значення) – підстановка значень у вираження

2. **Пакет simplex** – пакет лінійної оптимізації. Розглянемо приклад рішення оптимізаційної задачі симплексним методом (рисунком 11.1.8):

$$F = 2x_1 + 3x_2 \rightarrow \max$$

$$\begin{cases} x_1 + 3x_2 \leq 18, \\ 2x_1 + x_2 \leq 16, \\ x_2 \leq 5, \\ 3x_1 \leq 21. \end{cases}$$

$$x_1 \geq 0, x_2 \geq 0.$$

Для обчислення максимального значення цільової функції використовується функція `maximize`, для обчислення мінімального значення – функція `minimize`. Перший параметр цих функцій – опис цільової функції, другий – обмеження задачі.


```

> with(simplex);
> maximize(2*x1+3*x2, {x1+3*x2<=18, 2*x1+x2<=16, x2<=5, 3*x1<=21}
union{x1>=0, x2>=0});
{x2 = 4, x1 = 6}

```

Рис.11.1.8. Приклад розв'язання задачі лінійного програмування

Перерахуємо основні функції пакету `simplex`.

NONNEGATIVE – ключове слово, що вказує на невід'ємний тип даних;

basis – повернення списку основних змінних для безлічі лінійних рівнянь;

convexhull – обчислення опуклої оболонки для набору точок;

cterm – завдання констант для системи рівнянь або нерівностей;

define_zero – визначення найменшого значення, що приймається за нуль;

display – вивід в матричній формі системи рівнянь і нерівностей;

dual – видача зв'язаних виразів;

equality – опція для функції `convert`, що вказує на еквівалентність;

maximize – обчислення максимуму функції;

minimize – обчислення мінімуму функції;

standardize – приведення заданої системи рівнянь або нерівностей до стандартної форми.

3. Пакет **stats** – пакет статистики містить команди для обробки і проведення статистичного аналізу. Найважливішими підпакетами цього пакету є:

anova – підпакет дисперсійного аналізу;

describe – підпакет описових статистик;

fit – підпакет лінійної регресії;

random – підпакет генерації випадкових чисел;

statevalf – підпакет чисельного обчислення розподілів.

4. **Пакет finance** – пакет **фінансової математики** містить функції для виконання фінансових розрахунків. Перерахуємо деякі основні функції, призначені для обчислення поточної вартості різних фінансових об'єктів:

annuity – для обчислення ренти з постійними виплатами;

cashflows – для обчислення виплат, змінних від періоду до періоду;

growingannuity – для обчислення ренти із зростаючими виплатами;

growingperpatuity – для обчислення безстрокової ренти із зростаючими виплатами;

levelcoupon – для обчислення вартості облигації;

perpetuity – для обчислення поточного значення довічної ренти;

yieldtomaturity – для обчислення доходу від цінного паперу при його погашенні.

11.2. Оформлення наукових робіт за допомогою видавничої системи LATEX

Випускна робота бакалавра (магістра) є виконаним студентом закінченим дослідженням в рамках сучасних наукових напрямів фундаментальної і прикладної математики, інформатики, економіки, а також методики викладання економіки і математики.

По суті, випускна робота студента вузу є спробою написання студентом першої наукової роботи, тому вона має бути оформлена і структурована у відповідності з вимогами, що пред'являються до наукових видань. Сьогодні редакції провідних наукових журналів однією з вимог написання наукової статті вважають використання видавничої системи LATEX. І це не випадково. Вся потужність видавничої системи LATEX яскраво виявляється при підготовці науково-технічних друкарських видань високої якості, що містять велику кількість математичних формул і ілюстрацій.

Почати оформлення випускної роботи слід із запуску кращого на сьогоднішній день спеціалізованого редактора документів LATEX для Windows, яким є WinEdt, що дозволяє здійснювати вставку шаблонів таблиць, формул і т.п. натисненням всього лише однієї кнопки. Окрема панель для вставки практично будь-яких TeX-символів не гірша, ніж у традиційного редактора формул Equation Editor в Word'і. Компіляцію документа можна також здійснити натисненням однієї кнопки <TeXify> з автоматичним викликом всіх корисних утиліт, а також побачити результат за допомогою зручного превьюера для переглядання DVI-файлів.

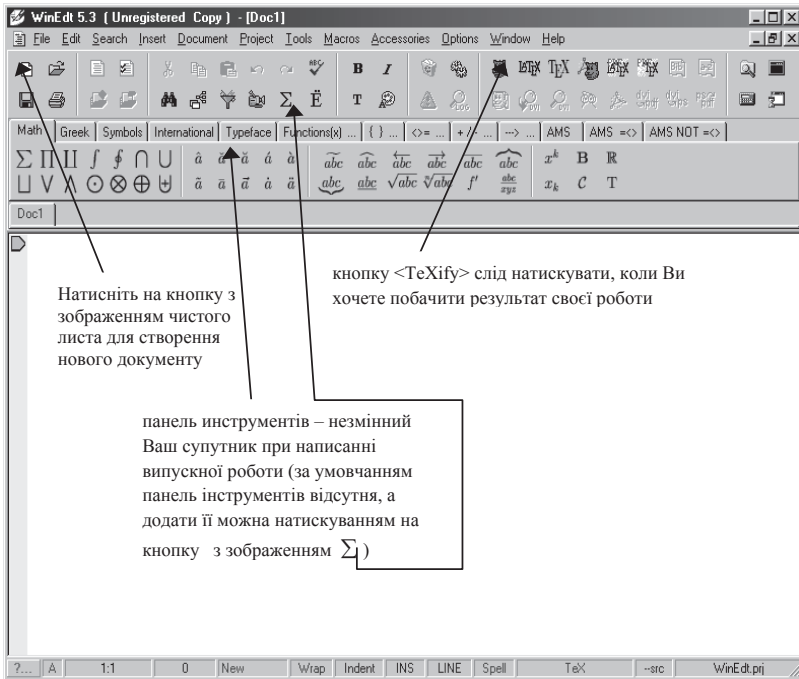


Рис. 11.1.9. Призначений для користувача інтерфейс WinEdt 5.3

Попередню підготовку до роботи виконаємо за допомогою наступних дій:

1. Натисніть кнопку із зображенням чистого листа для створення нового документа, або виконайте команду File| New, або натисніть комбінацію клавіш Ctrl+N;
2. Збережіть новий документ, натиснувши кнопку із зображенням дискети, або виконайте команду File| Save as..., або натисніть комбінацію клавіш Ctrl+S, після чого введіть ім'я файлу, що зберігається, наприклад, Diplom.tex

Оформлення випускної роботи почнемо з написання преамбули. Преамбула дозволяє встановити клас документа, розмір шрифту (кегля) тексту, встановити кодування Windows, підключити спеціальні пакети за

допомогою команди `\usepackage{}`, що розширяє функціональні можливості LATEX'a. Ім'я пакету, що підключається, записується у фігурних дужках.

Перші рядки при написанні випускної роботи в редакторі WinEdt мають вигляд (після знаку % можна записувати коментар):

```
\documentclass[12pt]{article} % установка класу документа article і
                               % розміру шрифту 12 пунктів
\usepackage[cp1251]{inputenc} % встановлюється кодування Windows

\usepackage[russian]{babel}   % основною мовою випускної роботи
                               % є російська мова
```

Додамо в преамбулу підключення графіки:

```
\usepackage[dvips]{graphicx}
\DeclareGraphicsRule{.wmf}{bmp}
\DeclareGraphicsRule{.emf}{bmp}
```

А також команди підключення основних пакетів:

```
\usepackage[T2A]{fontenc}
\usepackage{amsfonts,amsmath,amsxtra,amsthm,amssymb,latexsym
mathrsfs,srcltx,dsfont} % підключення додаткових шрифтів
\usepackage{eufrak, euscript, eucal} % підключення додаткових
                                   шрифтів
```

Команда	Результат	Пакет, що підключається
<code>\mathrm{ABCDEF}</code>	ABCDEF	
<code>\mathit{ABCDEF}</code>	<i>ABCDEF</i>	
<code>\mathnormal{ABCDEF}</code>	<i>ABCDEF</i>	
<code>\mathcal{ABCDEF}</code>	<i>ABCDEF</i>	eucal
<code>\mathfrak{ABCDEF}</code>	ℒℑℒℑ	eufrak
<code>\mathbb{ABCDEF}</code>	ABCDEF	amsfonts або amssymb

```
\usepackage{moreverb}
\usepackage[noend]{algorithmic}
```

`\usepackage{multicol}`

Встановимо параметри сторінки, додамо в преамбулу

<code>\textheight=24cm</code>	<i>% установка висоти тексту</i>
<code>\textwidth=16cm</code>	<i>% установка висоти тексту</i>
<code>\oddsidemargin=0mm</code>	<i>% величина лівого поля при однобічному наборі</i>
<code>\topmargin=-1cm</code>	<i>% розмір верхнього поля</i>
<code>\parindent=24pt</code>	<i>% установка величини абзацного відступу</i>
<code>\footnotesep=3ex</code>	
<code>\flushbottom</code>	
<code>\tolerance=3000</code>	
<code>\binoppenalty=10000</code>	
<code>\relpenalty=10000</code>	<i>% абсолютна заборона розриву рядків після бінарних відношень</i>
<code>\mathsurround=1pt</code>	<i>% оточення кожної формули усередині тексту додатковими пропусками по 1 пункту з обох боків</i>
<code>\frenchspacing</code>	<i>% заборона на установку різних проміжків між пропозиціями (тільки для російськомовних текстів)</i>

Весь текст випускної роботи слід укласти усередині оточення `\begin{document}` `\end{document}`, розміщеного безпосередньо після преамбули.

11.2.1. Верстка математичних формул і спеціальних символів в системі LATEX

Розрізнятимемо формули 3 типів: формули усередині тексту, формули, розміщені в окремому рядку, що мають номер і формули, що розміщені в окремому рядку і не мають номера. Формула усередині тексту оформляється за допомогою оточення `$....$`, усередині якого замість трикрапки записується сама формула. При розміщенні формули в окремому рядку без нумерації використовується оточення `$$...$$`, а з нумерацією – оточення `\begin{equation}... \end{equation}`.

При використанні оточення $\begin{equation}...\end{equation}$ формулі можна привласнити мітку за допомогою команди $\label{\text{ім'я мітки}}$. Ця можливість дозволяє посилатися на формулу в тексті за допомогою команди $\ref{\text{ім'я мітки}}$, пов'язане з командою $\label{\text{ім'я мітки}}$.

Формулі можна приписати будь-який номер (виключаючи автоматичну генерацію номерів формул) за допомогою команди \leqno (праворуч від формули) або команди \leqquo (зліва від формули).

Наведемо приклади:

$$\begin{aligned} & \begin{array}{l} \S\S \\ K_{\{U,D\}}(f)=\max_{\{x \in D\}} K_{\{U\}}(f|x) \leqno (*) \\ \S\S \end{array} & K_{U,D}(f)=\max_{x \in D} K_U(f|x) (*) \\ & \begin{array}{l} \S\S \\ K_{\{U,D\}}(f)=\max_{\{x \in D\}} K_{\{U\}}(f|x) \leqquo (*) \\ \S\S \end{array} & (*) K_{U,D}(f)=\max_{x \in D} K_U(f|x) \end{aligned}$$

Робота з текстом і установка пропусків у формулі, розташованій в окремому рядку. Фрагмент тексту можна включити у формулу за допомогою команди $\mbox{\}$ або $\textit{\}$, уклавши потрібний текст у фігурних дужках, а пропуски встановлюються за допомогою команд \quad («одинарний» пропуск) або \qquad («подвійний» пропуск). Втім, є можливість і витонченішої корекції пропусків усередині формул (команда $\,$ дає можливість встановлювати «тонкий» пропуск; команда $\:$ – «середній» пропуск; команда $\;$ - «товстий» пропуск; команда $\!$ дозволяє зменшити відстань між символами «розміром» в «тонкий» пропуск).

$$\begin{array}{l} \S\S \\ |x_n - a| < \varepsilon \quad \quad \quad \mbox{\text{при}} \quad n > N(\varepsilon) \\ \S\S \end{array}$$

$$|x_n - a| < \varepsilon \quad \text{при} \quad n > N(\varepsilon)$$

Розглянемо функціональні можливості, що надаються панеллю інструментів редактора WinEdt (вкладка *Math*). Результат досягається

натисненням однієї кнопки з вказівкою у фігурних дужках потрібних параметрів:

Math	Greek	Symbols	International	Typeface	Functions(x) ...	{ } ...	<>= ...	+ / · ...	→ ...	AMS	AMS									
Σ	Π	Π	\int	\oint	\cap	\cup	\hat{a}	\tilde{a}	\check{a}	\acute{a}	\grave{a}	\overline{abc}	\widehat{abc}	\overleftarrow{abc}	\overrightarrow{abc}	\overleftarrow{abc}	\overrightarrow{abc}	x^k	\mathbf{B}	\mathbb{R}
\sqcup	\vee	\wedge	\odot	\otimes	\oplus	\oplus	\bar{a}	\vec{a}	\dot{a}	\ddot{a}	\underbrace{abc}	\underbrace{abc}	\sqrt{abc}	$\sqrt[3]{abc}$	f'	$\frac{abc}{xyz}$	x_k	\mathcal{C}	\mathcal{T}	



1 Шаблони символів операцій ($\Sigma, \Pi, \cap, \cup, \wedge, \vee, \oplus$)



$\int y(x)dx$

$\int y(x)dx$

$\int_a^b y(x)dx$

$\int_a^b y(x)dx$



$\sum_{i=1}^n u_i(x)$

2 Шаблони надрядкових знаків для одного символу (нижче перелічені найбільш часто використовувані)



\hat{x}

\hat{x}



\vec{x}

\vec{x}



\tilde{x}

\tilde{x}



\dot{x}

\dot{x}



\bar{x}







\bar{x}



\ddot{x}

\ddot{x}

3 Шаблони надстрокових і підстрокових знаків для груп символів, шаблони кореней, дробів і т.п. (нижче перелічені найбільш часто використовувані).

	<code>\overrightarrow{AB}</code>	\overrightarrow{AB}
	<code>\widetilde{AB}</code>	\widetilde{AB}
	<code>\overbrace{a+a+\ldots}</code>	$\overbrace{a+a+\dots+a}$
	<code>\underbrace{a+a+\ldots}</code>	$\underbrace{a+a+\dots+a}$
	<code>\sqrt{x}</code>	\sqrt{x}
	<code>\sqrt[n+1]{x}</code>	$n+1\sqrt{x}$

Верстка дробів часом вимагає від користувача дійсного мистецтва. Розглянемо можливості, так званого «виключного» стилю `\displaystyle`:

\$\$

`\frac{x}{2}` **displaystyle** $x^2 + \frac{2}{2}$ **displaystyle**

$x^2 + \frac{4}{8}$ **displaystyle** $x^2 + \frac{8}{8}$ **displaystyle**



x^2

\$\$



$$x^2 + \frac{\frac{x}{2}}{x^2 + \frac{4}{x^2 + \frac{8}{x^2}}}$$

4 Шаблони верхніх і нижніх індексів і т.п.

x^k	$\$x^{\{n\}}\$$	x^n
x_k	$\$x_{\{n\}}\$$	x_n
\mathbb{R}	$\$\mathbb{N}\$$	\mathbb{N}


Наведемо приклад комбінування команд вкладки Math. Натиснемо кнопку  і введемо $\underbrace{a+a+\dots+a}$, а потім кнопку , установивши попередньо курсор в формулі перед другим знаком \$, і введемо в фігурних дужках n . Остаточню сформулюємо команду $\underbrace{a+a+\dots+a}_{n}$, результатом виконання якої буде

$$\underbrace{a + a + \dots + a}_n$$

Для верстки позначення числа A_n^k всіляких розміщень із n по k слід ввести $\$A\$$ і, установивши курсор після символу A, натиснути на кнопку , потім в фігурних дужках набрати верхній індекс k , отримаємо: $A^{\{k\}}$. Далі установити курсор перед другим знаком \$, натиснути на кнопку  і в фігурних дужках указати аргумент n . Остаточню отримаємо $A^{\{k\}_{n}}$.

Якщо потрібно, щоб верхні і нижні індекси розташовувалися на різних відстанях від вираження, до якого вони відносяться, то слід частину індексів оформити як індекси до «порожньої формули» (під «порожньою формулою» зазвичай розуміють $\{\}$). Розглянемо приклад:

$$\mathbb{R}_j^i \ell k$$





Введення грецьких букв можна здійснити натисненням однієї кнопки на вкладці *Greek* панелі інструментов. Єдине, що не слід забувати – обрамляти кожен грецьку букву знаками $\$ \dots \$$. Наприклад, введення першої букви грецького алфавіту здійснюється за допомогою команди $\$ \alpha \$$ шляхом натиснення на кнопку , коли курсор знаходиться між знаками $\$$ і $\$$.



Введення спеціальних символів можна здійснити натисненням однієї кнопки на вкладці *Symbols* панелі інструментів, неодмінно обрамлюючи кожен із змальованих на цій вкладці символів знаками $\$$ і $\$$.



Вкажемо найбільш часто використовувані команди вкладки *Symbols*:

-  $\$ \infty \$$ ∞
-  $\$ \exists \$$ \exists
-  $\$ \forall \$$ \forall
-  $\$ \emptyset \$$ \emptyset

У російськомовних математичних текстах для введення порожньої множини традиційно використовують команду:

$$\varnothing$$

Для введення *математичних функцій* використовується вкладка *Functions(x)*... панелі інструментів.

Math	Greek	Symbols	International	Typeface	Functions(x) ...	{ } ...	<>= ...	+ / - ...	--> ...	AMS	AMS =<>	◀ ▶		
arccos	arcsin	arctan	arg	cos	cosh	cot	coth	csc	det	dim	exp	gcd	hom	inf
lim	lim inf	lim sup	ker	lg	ln	log	max	min	sec	sin	sinh	sup	tan	tanh

Наведемо приклади використання команд цієї вкладки:

$$\lim_{n \rightarrow \infty} x_n$$

$$\lim_{n \rightarrow \infty} x_n$$

$$\lim_{n \rightarrow \infty} \frac{1^p + 2^p + \dots + n^p}{n^{p+1}}$$

$$\lim_{n \rightarrow \infty} \frac{1^p + 2^p + \dots + n^p}{n^{p+1}}$$

$$\overline{\lim_{n \rightarrow \infty} (x_n + y_n)} = \overline{\lim_{n \rightarrow \infty} x_n} + \overline{\lim_{n \rightarrow \infty} y_n}$$

$$\lim_{n \rightarrow \infty} (x_n + y_n) = \lim_{n \rightarrow \infty} x_n + \lim_{n \rightarrow \infty} y_n$$

$$\lim_{n \rightarrow \infty} \int_0^1 f_n(x) dx$$

$$\lim_{n \rightarrow \infty} \int_0^1 f_n(x) dx$$

Щоб межі інтегрування стояли безпосередньо над і під знаком інтегралу, слід після команди `\int` вказати команду `\limits`, а після неї – значення меж інтегрування.

$$\lim_{n \rightarrow \infty} \int_0^1 f_n(x) dx$$

$$\lim_{n \rightarrow \infty} \int_0^1 f_n(x) dx$$

За допомогою команди `\limits` подібного ефекту можна добитися і при записі формул усередині тексту. Наприклад

Довести, що якщо ряд $\sum_{n=1}^{\infty} a_n$ з позитивними і монотонно зменшуваними членами збігається, то $\lim_{n \rightarrow \infty} n a_n = 0$.

Довести, що якщо ряд $\sum_{n=1}^{\infty} a_n$ з позитивними і монотонно

зменшуваними членами збігається, то $\lim_{n \rightarrow \infty} n a_n = 0$.

Для верстки деяких математичних функцій і інших символів краще використовувати не команди вкладки `Functions(x)...`, а наступні прийняті для російськомовних математичних текстів команди:

<code>\tg x</code>	<i>tg x</i>
<code>\ctg x</code>	<i>ctg x</i>
<code>\arctg x</code>	<i>arctg x</i>
<code>\arcctg x</code>	<i>arcctg x</i>
<code>\sh x</code>	<i>sh x</i>

$\$th x\$$	$th x$
$\$cth x\$$	$cth x$
$\$\mathop{\text{Im}} z\$$	$\text{Im } z$
$\$\mathop{\text{Re}} z\$$	$\text{Re } z$

Для введення символів відношень слід скористатися вкладкою $\langle \rangle = \dots$ панелі інструментів:



Наприклад, \approx $\$ \pi \approx 3,14 \$$ $\pi \approx 3,14$

Розбиття довгої формули з вирівнюванням частин можна виконати за допомогою оточення $\begin{split} \end{split}$.

```
\begin{equation*}
\begin{split}
(1+x)^m=1+mx+\frac{m(m-1)}{2!}x^2+\dots \backslash
\ldots +\frac{m(m-1)\dots(m-n+1)}{n!} x^n+\dots \backslash, (-1<x<1)
\end{split}
\end{equation*}
```

$$(1+x)^m = 1 + mx + \frac{m(m-1)}{2!}x^2 + \dots$$

$$\dots + \frac{m(m-1)\dots(m-n+1)}{n!}x^n + \dots \quad (-1 < x < 1)$$

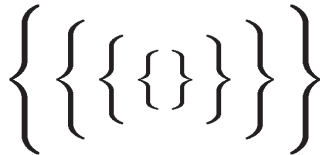
Для відступу усередині формули була використана команда, що складається із зворотної косої межі і точки (\backslash), щоб не привласнювати формулі номер, було використано оточення з «*» - `\begin{equation*}` `\end{equation*}`.

Установка правильного розміру дужок у формулах. Команда `\left` перед відкриваючою дужкою в сукупності з командою `\right` перед відповідною їй закриваючою дужкою дозволяє автоматично встановити потрібний розмір дужок:

$$\begin{equation*} f(x) = x \cdot \left(2 + \sin \frac{1}{x} \right) \end{equation*}$$

Розміром дужок можна керувати на свій розсуд за допомогою команд `\Bigg`, `\bigg`, `\Big`, `\big`:

$$\begin{equation*} \Bigg\{ \bigg\{ \Big\{ \big\{ \Big\} \big\} \Big\} \bigg\} \end{equation*}$$



За допомогою команд `\left` з крапкою в кінці і `\right` можна керувати розміром і числом інших обмежувачів (у даному прикладі лівий обмежувач відсутній за рахунок команди `\left`):

```

$$
\int \limits_{a}^{b} \frac{1}{2} (1+x)^{-3/2} dx =
\left. -\frac{1}{\sqrt{1+x}} \right|_{a}^{b}
$$

```

$$\int_a^b \frac{1}{2} (1+x)^{-3/2} dx = -\frac{1}{\sqrt{1+x}} \Big|_a^b$$

При верстці систем рівнянь або нерівностей ставиться лівий обмежувач `(` командою `\left`, а правий обмежувач елімінується за допомогою додавання крапки в кінці команди `\right`. (див. пункт «Верстка систем рівнянь, нерівностей та ін.»).

11.2.2. Створення найпростіших переліків і нумерованих списків в системі LATEX

Ознайомимося з можливостями пункту *Insert* головного меню і його підпункту *Lists*.

Для створення шаблону нумерованого списку виберіть пункти і відповідні ним підпункти *Insert* | *Lists* | *Enumerate* (з'явиться запит про число елементів в списку):

```

\begin{enumerate}
\item *
\item *
\item *
\item *
\end{enumerate}

```

Змінюючи символи «*» потрібними даними, отримаємо нумерований список:

<code>\begin{enumerate}</code>	1. $x + y = y + x$
<code>\item $x+y=y+x$</code>	
<code>\item $(x+y)+z=x+(y+z)$</code>	2. $(x + y) + z = x + (y + z)$
<code>\item $x+e=e+x=x$</code>	
<code>\item $x+(-x)=e$</code>	3. $x + e = e + x = x$
<code>\end{enumerate}</code>	
	4. $x + (-x) = e$

Для створення шаблону простого переліку виберіть пункти і відповідні ним підпункти *Insert* | *Lists* | *Itemize* (з'явиться запит про число елементів в переліку):

```
\begin{itemize}
  \item *
  \item *
  \item *
  \item *
\end{itemize}
```

Змінюючи символи «*» потрібними даними, отримаємо найпростіший перелік:

<code>\begin{itemize}</code>	• $x + y = y + x$
<code>\item $x+y=y+x$</code>	
<code>\item $(x+y)+z=x+(y+z)$</code>	• $(x + y) + z = x + (y + z)$
<code>\item $x+e=e+x=x$</code>	
<code>\item $x+(-x)=e$</code>	• $x + e = e + x = x$
<code>\end{itemize}</code>	
	• $x + (-x) = e$

Для створення шаблону переліку спеціального вигляду із заголовками виберіть пункти і відповідні ним підпункти *Insert* | *Lists* | *Description* (з'явиться запит про число елементів в переліку):

```
\begin{description}
  \item[*]*
  \item[*]*
  \item[*]*
  \item[*]*
\end{description}
```


`\end{description}`

Перелік спеціального вигляду із заголовками дозволяє задати в квадратних дужках замість символу «*» вид заголовка:

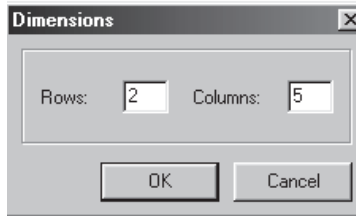
```
\begin{description}
  \item[1]  $x+y=y+x$ 
  \item[2]  $(x+y)+z=x+(y+z)$ 
  \item[3]  $x+e=e+x=x$ 
  \item[4]  $x+(-x)=e$ 
\end{description}
```

Оточення *description* зазвичай використовується для вирівнювання заголовків елементів переліку по лівому краю. Відзначимо, що оточення *enumerate* і *itemize* допускають зміни заголовків переліків за допомогою необов'язкового параметра `l` команди `\item`.

```
\begin{itemize}
  \item[-]  $x+y=y+x$ 
  \item[-]  $(x+y)+z=x+(y+z)$ 
  \item[-]  $x+e=e+x=x$ 
  \item[-]  $x+(-x)=e$ 
\end{itemize}
```

11.2.3. Верстка таблиць в системі LATEX

Верстка таблиць виконується за допомогою оточення `\begin{tabular}` `\end{tabular}`, можливо, з горизонтальними і вертикальними відокремлюючими лініями. Для автоматичного створення оточення таблиці при роботі в редакторі WinEdt використовуйте пункт головного меню *Insert*, в якому треба вибрати підпункт *Tabular (n × m)*. З'являється запит про число рядків і стовпців створюваної таблиці:



Після вказівки числа рядків і стовпців і натиснення на кнопку <OK> автоматично створюється шаблон вигляду:

```
\begin{tabular}|c|c|c|c|c|
\hline
% after \: \hline or \cline{col1-col2} \cline{col3-col4} ...
* & * & * & * & * & * \
\hline
* & * & * & * & * & * \
\hline
\end{tabular}
```

Замінивши символ «*» даними, отримаємо:

```
\begin{tabular}|c|c|c|c|c|
\hline
$x$ & -2 & 0 & 4 & 5 \
\hline
$y$ & 5 & 1 & -3 & 1 \
\hline
\end{tabular}
```

x	-2	0	4	5
y	5	1	-3	1

Преамбула (аргумент, специфікація) команди-оточення `\begin{tabular}` полягає у фігурні дужки і визначає формат таблиці. Так, вище `\begin{tabular}{|c|c|c|c|c|}` преамбула використовувалася для центрування тексту в елементах таблиці; для вирівнювання по лівому краю використовується преамбула `l \begin{tabular}{|l|l|l|l|l|}`; управо – преамбула `r`, а оточення таблиці записується як `\begin{tabular}{|r|r|r|r|r|}`.

Наявність в преамбулі символу «|» приводить до появи в таблиці вертикальних розділових ліній. Наведемо приклад таблиці, в якій відсутні вертикальні розділові лінії:

```
\begin{tabular}{cccc}
\hline
$x$ & -2 & 0 & 4 & 5 \
\hline
$y$ & 5 & 1 & -3 & 1 \
\hline
\end{tabular}
```

x	-2	0	4	5
y	5	1	-3	1

Наявність горизонтальних розділових ліній в таблиці забезпечується командою `\hline`. Візьмемо таблицю з попереднього прикладу і видалимо горизонтальні розділові лінії:

```
\begin{tabular}{cccc}
$x$ & -2 & 0 & 4 & 5 \
$y$ & 5 & 1 & -3 & 1 \
\end{tabular}
```

x	-2	0	4	5
y	5	1	-3	1

За допомогою команди `\multicolumn` можна об'єднати декілька стовпців в межах одного рядка таблиці.

```
\begin{tabular}{|c|c|c|c|c|}
\hline
$x$ & \multicolumn{3}{c}{-2 0 4}
& 5 \
\hline
$y$ & 5 & 1 & -3 & 1 \
\hline
\end{tabular}
```

x	-2	0	4	5
y	5	1	-3	1

11.2.4. Верстка матриць, визначників і систем рівнянь в системі LATEX

Познайомимось з деякими функціональними можливостями, що надаються пунктом *Insert* головного меню редактора WinEdt.

Для верстки систем рівнянь або нерівностей можна скористатися пунктом *Insert* головного меню, в якому слід вибрати підпункти *Lists* і *Cases*. Після чого послідує запит про число рядків, що входять в систему, вказавши які, користувач отримує шаблон наступного вигляду:

```

$$
\left%
\begin{array}{l}
*, & \hbox{*;} \backslash
*, & \hbox{*;} \backslash
\end{array} %
\right.
$$

```

Замінивши символи «*» необхідними даними, користувач вже через декілька секунд може милуватися результатом своєї роботи:

```

$$
\left%
\begin{array}{l}
ax & \hbox{якщо } \$\pi < x < 0\$; \backslash
bx & \hbox{якщо } \$0 < x < \pi\$; \backslash
\end{array} %
\right.
$$

```

$$\begin{cases} A, & \text{если } 0 < x < \ell; \\ 0, & \text{если } \ell < x < 2\ell. \end{cases}$$

За умовчанням виконано вирівнювання по лівому краю. Про це свідчить преамбула l , вказана після \begin{array} у фігурних дужках. Як легко

зрозуміти, вид вирівнювання можна змінити, вказавши в преамбулі r (вирівнювання по правому краю) або c (вирівнювання по центру).

```

 $f(x)=$ 
\left%
\begin{array}{rr}
x & \hbox{ якщо } 0\leq x \leq 1; \setminus
1 & \hbox{ якщо } 1<x<2; \setminus
3-x & \hbox{ якщо } 2\leq x \leq 3. \setminus
\end{array}%
\right.
 $$ 
```

$$f(x) = \begin{cases} x, & \text{если } 0 \leq x \leq 1; \\ 1, & \text{если } 1 < x < 2; \\ 3 - x, & \text{если } 2 \leq x \leq 3. \end{cases}$$

У російськомовних математичних текстах прийнято замість команди нестрогої нерівності \leq використовувати команду \leqslant .

```

 $f(x)=$ 
\left%
\begin{array}{rr}
x & \hbox{ якщо } 0\leqslant x \leqslant 1; \setminus
1 & \hbox{ якщо } 1<x<2; \setminus
3-x & \hbox{ якщо } 2\leqslant x \leqslant 3. \setminus
\end{array}%
\right.
 $$ 
```

$$f(x) = \begin{cases} x, & \text{если } 0 \leqslant x \leqslant 1; \\ 1, & \text{если } 1 < x < 2; \\ 3 - x, & \text{если } 2 \leqslant x \leqslant 3. \end{cases}$$

Якщо необхідно, щоб окремі рівняння в системі були пронумеровані, слід скористатися пунктом *Insert* головного меню, потім *Environments*, потім *Eqnarray*. Послідовність таких дій приводить до появи запиту про число рівнянь в системі з подальшим створенням шаблону:

```

\begin{eqnarray}
% \nonumber to remove numbering (before each equation)
* &=& * \setminus

```

```
* &= & * \
* &= & *
\end{eqnarray}
```

Команда `\` відокремлює одне рівняння системи від іншого. Для відміни нумерації якого-небудь рівняння системи слід використовувати команду `\nonumber` відразу після команди `\` і перед тим рівнянням системи, біля якого не буде номера. Замінюючи символи «*» необхідними даними, отримаємо:

```
\begin{eqnarray}
(5-\lambda)x_1-2x_2-x_3&=& 1 \ \nonumber
-2x_1+(2-\lambda)x_2-2x_3 \ &=& 2 \
-x_1-2x_2+(5-\lambda)x_3&=& 1.
\end{eqnarray}
```

$$(5 - \lambda)x_1 - 2x_2 - x_3 = 1, \tag{1}$$

$$-2x_1 + (2 - \lambda)x_2 - 2x_3 = 2,$$

$$-x_1 - 2x_2 + (5 - \lambda)x_3 = 1. \tag{2}$$

Вибір пункту *Insert* головного меню, а потім підпункту *Array* ($n \times m$) приводить до появи запиту про число рядків і стовпців таблиці чисел, після вказівки яких в текст автоматично буде вбудований шаблон таблиці чисел:

```
$$
\begin{array}{ccc}
* & * & * \
* & * & * \
* & * & * \
\end{array}
$$
```

Замінивши в шаблоні символи «*» необхідними даними, отримаємо:

```
$$
\begin{array}{ccc}
1 & 0 & 0 \
0 & 1 & 0 \
0 & 0 & 1 \
\end{array}
0 & 0 & 1
$$
```

Оточення *array* можна використовувати для створення таблиць чисел довільної форми. Наприклад, для того, щоб надрукувати трикутник Паскаля створимо за допомогою шаблону (*Insert | Array (n × m)*) таблицю, що містить 5 рядків і 11 стовпців. Шаблон такої таблиці має вигляд:

```

$$
\begin{array}{ccccccccc}
* & * & * & * & * & * & * & * & * & * & * & * \\
* & * & * & * & * & * & * & * & * & * & * & * \\
* & * & * & * & * & * & * & * & * & * & * & * \\
* & * & * & * & * & * & * & * & * & * & * & * \\
* & * & * & * & * & * & * & * & * & * & * & * \\
\end{array}
$$

```

Якщо якийсь елемент таблиці чисел повинен залишатися порожнім, то символ «*» слід просто видалити:

```

$$
\begin{array}{ccccccccc}
& & & 1 & & 1 & & & & & & \\
& & & 1 & & 2 & & 1 & & & & \\
& & 1 & & 3 & & 3 & & 1 & & & \\
& 1 & & 4 & & 6 & & 4 & & 1 & & \\
1 & & 5 & & 10 & & 10 & & 5 & & 1 & \\
\end{array}
$$

```

			1		1						
			1		2		1				
		1		3		3		1			
	1		4		6		4		1		
1		5		10		10		5		1	

Скористаємося пунктом *Insert* головного меню, а також його підпунктом *AMSTeX* для створення таблиць чисел з різними типами дужок.

Вибір *Insert | AMSTeX | (Matrix) (n × m)* приводить до появи запиту про число рядків і стовпців таблиці чисел, після вказівки яких в текст автоматично буде вбудований шаблон вигляду:

```

$$
\begin{pmatrix}
* & * \ \\
* & * \
\end{pmatrix}
$$

```

Заміна символів «*» даними приводить до створення таблиці чисел (наприклад, матриці) з круглими дужками:

```

$$
\begin{pmatrix}
15 & 13 \ \\
17 & 25 \
\end{pmatrix}
$$

```

Вибір *Insert | AMSTeX | [Matrix] (n × m)* приводить до появи запиту про число рядків і стовпців таблиці чисел, після вказівки яких в текст автоматично буде вбудований шаблон вигляду:

```

$$
\begin{bmatrix}
* & * \ \\
* & * \
\end{bmatrix}
$$

```

Заміна символів «*» даними приводить до створення таблиці чисел з квадратними дужками:

```

$$
\begin{bmatrix}
15 & 13 \ \\
17 & 25 \
\end{bmatrix}
$$

```


Вибір *Insert | AMSTeX | |Matrix|* ($n \times m$) приводить до появи запиту про число рядків і стовпців визначника, після вказівки яких в текст автоматично буде вбудований шаблон вигляду:

```

$$
\begin{vmatrix}
* & * \backslash \\
* & * \backslash
\end{vmatrix}
$$

```

Заміна символів «*» даними приводить до створення визначника в оточенні $\begin{vmatrix}$:

```

$$
\begin{vmatrix}
15 & 13 \backslash \\
17 & 25 \backslash
\end{vmatrix}
$$

```

Вибір *Insert | AMSTeX | ||Matrix||* ($n \times m$) приводить до появи запиту про число рядків і стовпців таблиці чисел, після вказівки яких в текст автоматично буде вбудований шаблон вигляду:

```

$$
\begin{Vmatrix}
* & * \backslash \\
* & * \backslash
\end{Vmatrix}
$$

```

Заміна символів «*» даними приводить до створення таблиці чисел в оточенні $\begin{Vmatrix}$:

```

$$
\begin{Vmatrix}
15 & 13 \backslash \\
17 & 25 \backslash
\end{Vmatrix}
$$

```

11.2.5. Оформлення визначень, теорем, лем і тверджень в системі LATEX

За допомогою команди `\newtheorem{theoremrus}{Теорема}` створимо власне оточення `\begin{theoremrus} \end{theoremrus}` для автоматично нумерованих теорем. У текст випускної роботи необхідно ввести послідовність наступних команд:

```
\newtheorem{theoremrus} {Теорема}
\begin{theoremrus}
Вірогідність того, що \ldots задовольняє нерівності \ldots
\end{theoremrus}
```

Надалі в тексті роботи при оформленні чергової теореми використовуйте оточення `\begin{theoremrus} \end{theoremrus}` (при цьому, команду `\newtheorem{theoremrus} {Теорема}` використовувати вже не треба). Команду `\newtheorem{theoremrus} {Теорема}` і подібні до неї краще розміщувати в преамбулі документа перед оточенням `\begin{document}`. Якщо написати команду `\newtheorem{theoremrus} {Теорема}[section]`, то номери теорем будуть пов'язані з номерами розділів, в яких ці теореми розглядаються (відаватимемо перевагу саме цьому варіанту).

```
\newtheorem{theoremrus} Теорема [section]
\begin{theoremrus}
Вірогідність того, що \ldots задовольняє нерівності \ldots
\end{theoremrus}
```

Оточення для визначень:

```
\newtheorem{definitionrus}Визначення[section]
\begin{definitionrus}
.....
\end{definitionrus}
```

Оточення для тверджень:

```
\newtheorem{statementrus}Твердження[section]
\begin{statementrus}
.....
\end{statementrus}
```

Ясно, що список створюваних оточень можна продовжувати на свій розсуд.

11.2.6. Вставка графіки в системі LATEX

Ознайомимося з можливостями пакету *graphics* (підключення пакету записується в преамбулі командою `\usepackage{graphics}`). Для вставки в текст малюнка з графічного файлу використовується команда:

```
\includegraphics[список значень ключів]{ ім'я графічного файлу}
```

Попередня підготовка перед вставкою малюнка з графічного файлу полягає в створенні *bmp*- або *eps*-файла, який слід зберегти в тій же папці, що і робочий файл з текстом.

Виберіть підпункт *Enviroments* пункту *Insert* головного меню редактора WinEdt, виконавши послідовність дій *Insert* | *Enviroments* | *Figure*. В результаті таких дій в тексті автоматично буде створений шаблон вставки ілюстрацій:

```
\begin{figure}  
% Потрібний \usepackage{graphicx}  
\includegraphics[width=*]{*}  
\caption{*}label{*}  
\end{figure}
```

Розглянемо, які параметри слід вказати замість символів «*»:
Спочатку за допомогою команди `\includegraphics[width=*]{*}` треба вказати після знаку «=» ширину розміщуваної ілюстрації (параметр в квадратних дужках), при цьому для *bmp*-файла треба додати через кому ще один параметр «height=», потім вказати ім'я *bmp*- або *eps*-файла (параметр у фігурних дужках). За допомогою команди `\caption{*}` можна задати підпис до малюнка (який слід вказати у фігурних дужках замість символу «*»).

Приклад вставки малюнка:

```
\begin{figure}  
% Потрібний \usepackage{graphicx}  
\includegraphics[width=25 mm,height=23 mm]{emblem.bmp}  
\caption{ТНУ}label{emblem1}  
\end{figure}
```



Рис.1: ТНУ

За умовчанням малюнок буде вирівняний по лівому краю, а підпис до малюнка по центру. Для розміщення малюнка по центру додамо команду

заборони абзацного відступу `\noindent` і команду центрування ілюстрації `\centering`:

Приклад вставки малюнка:

```
\begin{figure}
% Потрібний \usepackage{graphicx}
\noindent \centering{
\includegraphics[width=25 mm, height=23 mm]{emblem.bmp}
}
\caption{ТНУ}label{emblem1}
\end{figure}
```



Рис.1: ТНУ

Оточення `figure` має необов'язковий аргумент, що вказується в дужках `[*]`. Замість символу «*» можна вказати одну з чотирьох букв:

`[t]` – розмістити ілюстрацію у верхній частині сторінки;

`[b]` – розмістити ілюстрацію в нижній частині сторінки;

`[p]` – розмістити ілюстрацію на окремій сторінці, що цілком складається з «плаваючих» ілюстрацій.

Вставка двох ілюстрацій поруч. Заздалегідь створіть два графічні файли і збережіть їх в тій же папці, що і `tex`-файл. Розміщення двох ілюстрацій поруч здійснимо за допомогою оточення `\begin{multicols} {2} ... \end{multicols}` (параметр `{2}` вказує число створюваних колонок). Заздалегідь в преамбулу документа треба додати підключення відповідного пакету `\usepackage{multicol}`.

```
\begin{figure}
\begin{multicols} {2}
\hfill
\includegraphics[width=25 mm,height=23 mm]{emblem1.bmp}
\hfill
\caption{герб зліва}label{emblemleft}
\hfill
\includegraphics[width=25 mm,height=23 mm]{emblem2.bmp}
\hfill
\caption{герб справа}label{emblemright}
\end{multicols}
\end{figure}
```

\end{figure}



Рис.1: герб зліва



Рис.2: герб справа

Поєднання ілюстрацій з текстом. Для поєднання ілюстрацій з текстом використовуватимемо команду `\parbox`.

Приклад поєднання ілюстрації з текстом:

```
\noindent
\parbox[b][3 cm]{10 mm}
\includegraphics[width=25 mm,height=23 mm]{emblem.bmp}
\hfill
\parbox [b] [3 cm]{125 mm}{Таврійський національний університет імені
В.\,И.\,Вернадського є наступником першого вищого учбового закладу в
Криму~--- Таврійського університету, створеного в 1918 році.}
```



Таврійський національний університет імені В.И. Вернадського є наступником першого вищого учбового закладу в Криму – Таврійського університету, створеного в 1918 році.

11.2.7. Створення псевдомалюнків в системі LATEX

За допомогою оточення `picture` можна створювати псевдомалюнки. Безпосередньо після `\begin{picture}` (*ширина, висота*) у круглих дужках через кому указуються розміри створюваного малюнка. Заздалегідь вкажемо одиницю виміру розмірів псевдомалюнка за допомогою команди `\unitlength=1mm`.

Для розміщення псевдомалюнка використовується команда `\put(абсциса, ордината){указується об'єкт, який слід розмістити}`.

Для створення відрізка використовується команда `\line{пара чисел, що визначає нахил відрізка}{довжина проєкції відрізка на горизонтальну вісь}`.

```
\begin{picture}(100,50)
\put(60,50) {\line(1,-2){30}}
\end{picture}
```



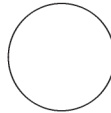
Стрілу (вектор) можна змалювати за допомогою команди `\vector`, що має ті ж параметри, що і команда `\line`.

```
\begin{picture}(100,50)
\put(60,50) {\vector(1,-2){30}}
\end{picture}
```



Окружність можна задати за допомогою команди `\circle{}`, а круг - `\circle*{}` (в фігурних дужках вказується діаметр окружності або круга відповідно).

```
\begin{picture}(100,100)
\put(40,40) {\circle{40}}
\end{picture}
```



```
\begin{picture}(100,100)
\put(40,40) {\circle*{40}}
\end{picture}
```



11.2.8. Верстка лістингів програм в системі LATEX

Для буквального відтворення лістингів програм слід використовувати оточення `\begin{verbatim} ... \end{verbatim}`, шаблон якого з'являється, якщо виконати послідовність дій *Insert* | *Enviroments* | *Verbatim*. Наприклад

```
\begin{verbatim}  
Uses Crt;  
Begin ClrScr;  
WriteLn('Hello, world!');  
ReadLn;  
End.  
\end{verbatim}
```

```
Uses Crt;  
Begin ClrScr;  
WriteLn('Hello, world!');  
ReadLn;  
End.
```

*Вставку лістингу програми в текст наукової роботи можна істотно спростити. Скористаємося для цієї мети можливостями пакету *moreverb* (заздалегідь в преамбулу слід додати команду підключення пакету `\usepackage{moreverb}`). Для того, щоб вивести у вигляді лістингу вміст файлу `first.pas` треба виконати команду*

`\listinginput[1]{1}{first.pas}`

```
1 Uses Crt;  
2 Begin ClrScr;  
3 WriteLn('Hello, world!');  
4 ReadLn;  
5 End.
```

Список літератури

1. Матросов А.В. Maple 6. Решение задач высшей математики и механики. – СПб.:БХВ-Петербург, 2001, 528 с.
2. Дьяконов В. Maple 7: учебный курс. – СПб: Питер, 2002, 672 с.
3. Котельников И.А., Чеботаев П.З. Издательская система LATEX2ε. Новосибирск : сибирский хронограф, 1998.
4. Львовский С.М. Набор и верстка в пакете LATEX, 2-е издание. М.: Космосинформ, 1995.

Зміст

Розділ I. ОСНОВНІ ПОНЯТТЯ ІНФОРМАТИКИ	3
Розділ II. ТЕОРЕТИЧНІ ОСНОВИ ІНФОРМАТИКИ	22
Розділ III. ОРГАНІЗАЦІЯ І ПРИНЦИПИ ФУНКЦІОНУВАННЯ КОМП'ЮТЕРІВ.....	66
Розділ IV. ОПЕРАЦІЙНІ СИСТЕМИ	80
Розділ V. БАЗИ ДАНИХ.....	112
Розділ VI. Офісний пакет програм	
Microsoft Office 2007	145
Глава VII. КОМП'ЮТЕРНІ МЕРЕЖІ	
I ТЕЛЕКОМУНІКАЦІЇ.....	347
Розділ VIII. Глобальна комп'ютерна мережа Internet.....	365
Розділ IX. ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ В ЕКОНОМІЦІ І МЕНЕДЖМЕНТІ.....	386
Розділ X. ІНТЕЛЕКТУАЛЬНІ КОМП'ЮТЕРНІ СИСТЕМИ.....	405
Розділ XI. ПАКЕТИ ПРИКЛАДНИХ ПРОГРАМ.....	415

НАВЧАЛЬНЕ ВИДАННЯ

Наталія Володимирівна АПАТОВА

Оксана Миколаївна ГОНЧАРОВА

Юлія Юріївна ДЮЛІЧЕВА

ІНФОРМАТИКА ДЛЯ ЕКОНОМІСТІВ

ПІДРУЧНИК

Керівник видавничих проєктів – Сладкевич Б. А.

Оригінал-макет підготовлено
ТОВ «Центр учбової літератури»

Підписано до друку 08.11.2010. Формат 60x84^{1/16}
Друк офсетний. Папір офсетний. Гарнітура PetersburgCTT.
Умовн. друк. арк. 25,65. Наклад – 700 прим.

Видавництво «Центр учбової літератури»
вул. Електриків, 23 м. Київ 04176
тел./факс 044-425-01-34
тел.: 044-425-20-63; 425-04-47; 451-65-95
800-501-68-00 (безкоштовно в межах України)
e-mail: office@uabook.com
сайт: www.cul.com.ua

Свідоцтво суб'єкта видавничої справи ДК № 2458 від 30.03.2006