

ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ»
МІНІСТЕРСТВА ОСВІТИ І НАУКИ УКРАЇНИ

Н.В.Матвіїшина

ОРГАНІЗАЦІЯ ТА ОБРОБКА ЕЛЕКТРОННОЇ ІНФОРМАЦІЇ

Методичні рекомендації до лабораторних занять
для студентів освітнього ступеня «бакалавр»
напряму підготовки «Інформатика»

Затверджено
вченою радою ЗНУ
Протокол № 11 від 26.05.2015

Запоріжжя
2015

УДК 004:005.57(075.8)
ББК 3937.202я73
МЗ38

Матвіїшина Н.В.

Організація та обробка електронної інформації: методичні рекомендації до лабораторних занять для студентів освітнього ступеня «бакалавр» напряму підготовки «Інформатика». – Запоріжжя : Запорізький національний університет, 2015. – 60 с.

Методичні рекомендації спрямовані на закріплення студентами теоретичних знань з дисципліни «Організація та обробка електронної інформації» за такими темами: робота з офісними додатками (Word, Excel), створення Web-сайту засобами HTML, офісне програмування на VBA.

Відповідно до навчальної програми дисципліни, в методичних рекомендаціях надано необхідну теоретичну інформацію, а також розглянуто питання щодо практичного використання набутих знань і навичок. До кожної теми, що розглядається в методичних рекомендаціях, надано контрольні запитання.

Засвоєнню теоретичного матеріалу сприятиме розроблений термінологічний словник ключових понять.

Методичні рекомендації з дисципліни «Організація та обробка електронної інформації» призначені для студентів напряму підготовки «Інформатика», але представлене видання може бути корисним студентам різних спеціальностей і напрямів підготовки.

Рецензент *Д.П.Бувайло*

Відповідальний за випуск *С. Ю. Борю*

ВСТУП

Основними завданнями вивчення дисципліни «Організація та обробка електронної інформації» є вивчення основних методів та засобів технологій, що використовуються для обробки електронної інформації, у тому числі на програмному рівні; вивчення засобів підготовки та реалізації електронних ресурсів.

Метою дисципліни є вивчення теоретичних основ і принципів роботи із сучасними програмними офісними додатками та засобами обробки електронної інформації на програмному рівні.

У результаті вивчення дисципліни студенти повинні:

- знати принципи обробки інформації на рівні офісу; основні види та формати електронних документів;
- знати принципи створення та редагування електронних ресурсів за допомогою HTML;
- знати принципи обробки електронної інформації за допомогою мови інтеграції офісних застосунків VBA.
- використовувати прикладне програмне забезпечення, зокрема пакети прикладних програм, офісні, навчальні, мультимедійні, графічні системи, системи керування вмістом;
- володіти методами та сучасними програмними засобами для налагодження програм та програмних комплексів.

Вивчення дисципліни дає підґрунтя для подальшого засвоєння можливостей сучасних інформаційних технологій для організації та обробки електронної інформації.

Структура методичних рекомендацій

Методичні рекомендації складаються зі вступу, 4 теоретичних тем, що вивчаються студентами напряму підготовки «Інформатика» в рамках дисципліни «Організація та обробка електронної інформації» та завдань лабораторних робіт. Теоретичний матеріал структурований та проілюстрований скріншотами, рисунками і схемами, що надає наочності представленому матеріалу. Наприкінці кожного теоретичного розділу є контрольні запитання.

Друга частина рекомендацій містить завдання до 8 лабораторних робіт за такими темами:

1. Редагування тексту документа у MS Word.
2. Створення комплексного документу у MS Word.
3. Злиття документів у MS Word.
4. Створення таблиці та побудова діаграм у MS Excel.
5. Розробка Web-сайту засобами мови HTML.
6. Розробка Web-сайту: створення фреймової структури, гіперпосилань.
7. Мова програмування VBA: функції користувача у MS Excel.
8. Мова програмування VBA: створення процедури.

Після виконання лабораторної роботи студентам рекомендується відповісти на запитання для самоперевірки.

1 ІНСТРУМЕНТИ ТА ЗАСОБИ РОБОТИ З КОМПЛЕКСНИМИ ТЕКСТОВИМИ ДОКУМЕНТАМИ. ТЕКСТОВИЙ ПРОЦЕСОР

1.1 Основні можливості текстового процесора

Повнофункціональний текстовий процесор, наприклад MS Word, повинен володіти багатьма можливостями.

1.1.1 Базові задачі текстового процесора

Введення та редагування тексту. Введення та редагування тексту – це базові задачі, для вирішення яких призначений текстовий процесор. Після вводу тексту може виникнути необхідність редагувати його: ввести додатковий текст, видалити, перемістити або скопіювати частину вже введеного тексту, виправити слова, у яких є орфографічні помилки або пропущені букви.

Для виконання операцій з фрагментами тексту у текстовому процесорі MS Word використовується меню **«Правка»**, **«Вставка»**:

- **«Правка»/«Выделить все»** – виділення вмісту всього документу;
- **«Правка»/«Копировать»** (кнопка ) , **«Правка»/«Вырезать»** (кнопка ) , **«Правка»/«Вставить»** (кнопка ) – копіювання та переміщення фрагментів тексту за допомогою буферу обміну;
- **«Вставка»/«Файл»** – розміщення в документі вмісту іншого документу;
- **«Правка»/«Отменить»** (кнопка ) – скасування помилково виконаних дій.

Пошук і заміна. Текстовий процесор MS Word має механізм, що забезпечує пошук заданого слова, словосполучення, поєднання символів або тексту у заданому форматі. Текст, що знайдено, може бути автоматично замінений запропонованим фрагментом, або може бути виділений для подальших дій користувача.

Для виконання операцій пошуку і заміни використовуються команди **«Правка»/«Найти...»** і **«Правка»/«Заменить...»** (Рис. 1).

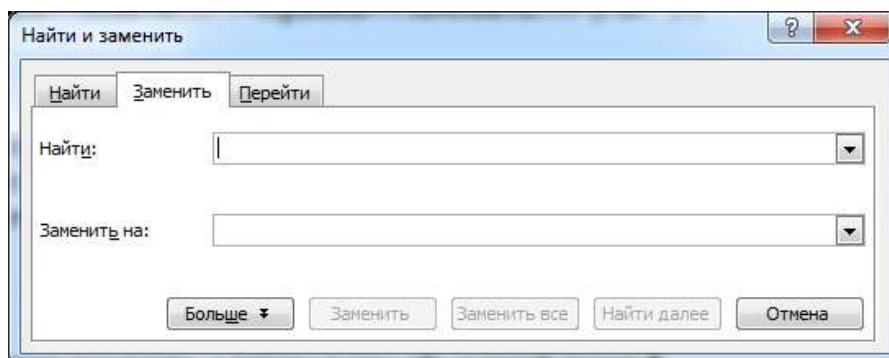


Рис. 1 Діалогове вікно «Найти и заменить»

Створення, відкриття та збереження документів. Можливість створення, відкриття та збереження документу в текстовому процесорі MS Word забезпечується меню «Файл»:

- «Файл»/«Создать» – створення нового документу;
- «Файл»/«Открыть» – відкриття документу;
- «Файл»/«Сохранить...», «Файл»/«Сохранить как...» – збереження документу (можливо, з іншим ім'ям або в іншій папці).

Можливість збереження документу в інших форматах – одна з важливих функцій сучасного текстового процесору. Один і той же документ може бути потрібен користувачу у різних представленнях, наприклад:

- у вигляді тексту – збереження тільки текстового вмісту;
- у вигляді документу іншого формату – для відкриття у текстовому процесорі іншого типу;
- у форматі PDF – для подальшого друку і читання в іншій операційній системі;
- у форматі HTML – для публікації на web-сторінці.

Кожний документ базується на певному шаблоні. **Шаблон** визначає основну структуру документа і містить набір елементів форматування. Існують загальні шаблони і шаблони документів. До загальних належить, наприклад, шаблон *Normal.dot*, що містить налаштування, доступні для всіх документів.

Шаблон документа доступний тільки для документу, створеного на основі цього шаблону, наприклад, шаблони записок або факсів, можна обрати командою «Файл»/«Создать»/«На моем компьютере» (Рис. 2).



Рис. 2 Діалогове вікно «Шаблоны»

Документ Word можна відображати в *чотирьох* режимах (меню «Вид», або кнопки , що розташовані у лівому нижньому куті вікна документа):

1) *звичайний режим* – представляється тільки змістовна частина документа без реквізитних елементів оформлення, що відносяться не до тексту, а до друкарських сторінок. Режим зручний на ранніх етапах розробки документа;

2) *режим розмітки* – екранне представлення документа повністю відповідає друкарському. Режим зручний для більшості робіт, пов'язаних з форматуванням тексту, призначеного для друку;

3) *режим Web-документа* – екранне представлення документа не відповідає друкарському. Режим характерний для електронних публікацій;

4) *режим структури* – корисний, коли розробка документа починається зі створення плану змісту. При активізації цього режиму автоматично відкривається панель інструментів «**Структура**», кнопки якої дозволяють змінювати структуру документа.

1.1.2 Форматування тексту

Основними операціями форматування у MS Word є: форматування символів, форматування абзаців, створення списків.

Форматування символів. Форматування символів здійснюється за допомогою команди «**Формат**»/«**Шрифт**» (Рис. 3), або панелі інструментів «**Форматирование**»:

- вибір гарнітури шрифту (Times New Roman);
- вибір розміру шрифту (12);
- вибір накреслення шрифту (Ж К Ч).

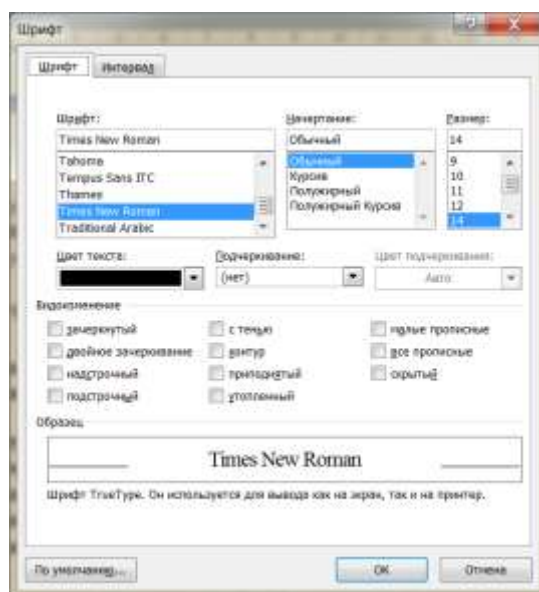


Рис. 3 Діалогове вікно «Шрифт»

Форматування абзацу – це зміна його параметрів як структурного елементу тексту. За допомогою команди «**Формат**»/«**Абзац**» (Рис. 4), або панелі інструментів «**Форматирование**» можна змінити:

- тип вирівнювання: по правому краю, по лівому краю, по центру, по ширині ;
- відступи від полів сторінки: зліва, справа
- відступи першого рядка абзацу;
- міжрядковий інтервал: одинарний, полуторний і т.д.;
- відстань між абзацами;
- контроль положення абзацу на сторінці.

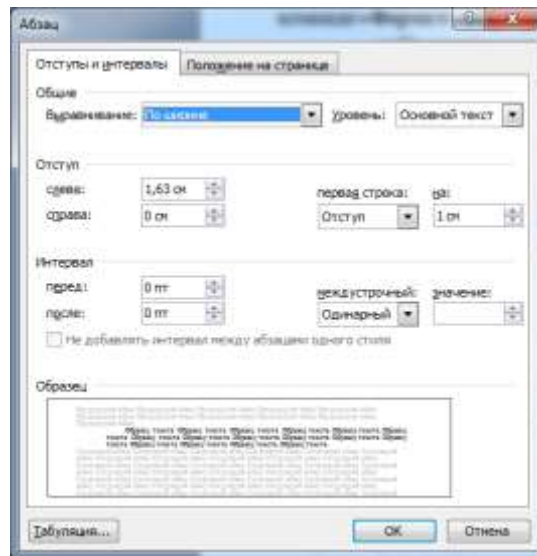
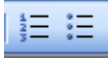


Рис. 4 Діалогове вікно «Абзац»

Побудова списків. Декілька послідовно розташованих абзаців можна оформити у вигляді списку: маркірованого, нумерованого, багаторівневого. Створення списку здійснюється за допомогою команди «Формат»/«Список» (Рис. 5), або панелі інструментів «Форматирование» .

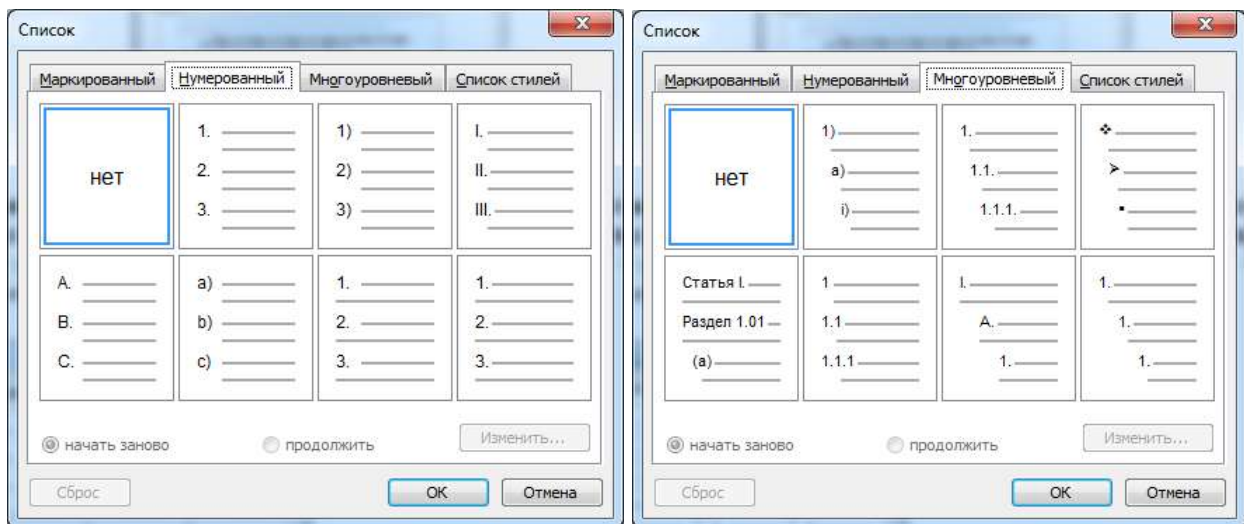


Рис. 5 Діалогове вікно «Список»

1.2 Засоби автоматизації розробки текстового документу

Інструментами, що автоматизують підготовку документу, є засоби для автоматичного створення змісту, покажчиків, списків ілюстрацій та використаної літератури, а також автоматичного форматування тексту. Однак, для того, щоб використати ці засоби, текст, що вводиться необхідно оформити відповідним стилем.

1.2.1 Стили

Стилем називають іменований набір параметрів форматування, який застосовується до певних структурних одиниць документа. Стили дозволяють за одну дію застосувати деякий набір елементів форматування.

Будь-який документ створюється на основі одного із стилів, що зберігається в шаблоні цього документа. За замовчуванням використовується шаблон *Normal* і стиль *Обычный*.

За допомогою команди «**Формат**»/«**Стили и форматирование**» або кнопки  панелі інструментів «**Форматирование**» можна переглянути, застосувати і створити стилі: символів, абзацу, таблиці, списків. Для зміни вигляду тексту потрібно його виділити і застосувати існуючий (вбудований) стиль, вибираючи його зі списку у вікні «**Выберите форматирование для применения**» області задач «**Стили и форматирование**» (Рис. 6).

Якщо потрібний стиль відсутній, його можна створити, а потім застосувати

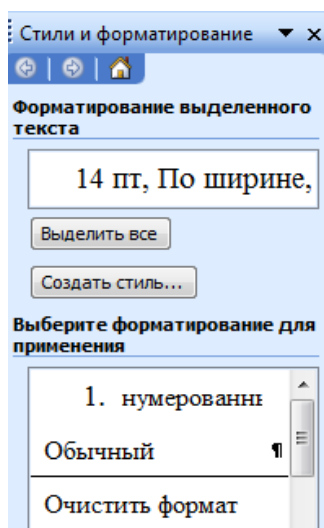


Рис. 6 Область задач «Стили и форматирование»

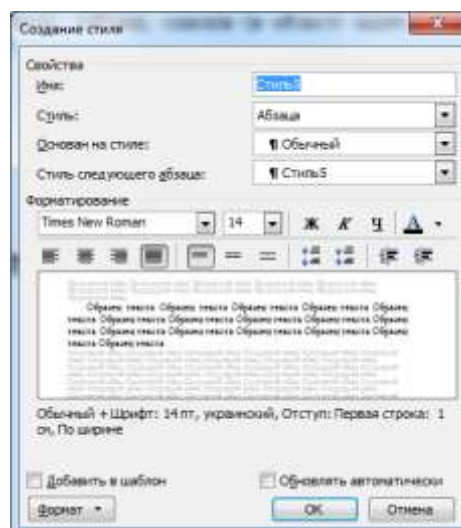


Рис. 7 Діалогове вікно «Создание стиля»

1.2.2 Поля

Поле – певна сукупність команд, реалізована спеціальними кодами, що застосовується для вставлення в документ додаткової інформації (номера сторінок, дати і т.д.). Будь-яке поле можна ввести за допомогою універсальної команди «**Вставка**»/«**Поле...**».

Всі поля в електронному документі мають сіре заливання, тобто елемент оновлюваний. Деякі з полів оновлюються автоматично, деякі – за допомогою команди контекстного меню «**Обновить поле**».

Список ілюстрацій – це список, елементами якого є назви (наприклад, Рис.1) що додають до таблиці, рисунку, діаграми і т.д.

З метою автоматичного створення списку ілюстрацій у документі слід створити назви вищеперелічених об'єктів. Для цього виконується команда **«Вставка»/«Ссылка»/«Название»** (Рис. 8).

Після створення назв ілюстрацій, необхідно виконати команду **«Вставка»/«Ссылка»/«Оглавление и указатели»/ вкладка «Список иллюстраций»** (Рис. 9), вибрати вид списку, встановити необхідні параметри і натиснути ОК.

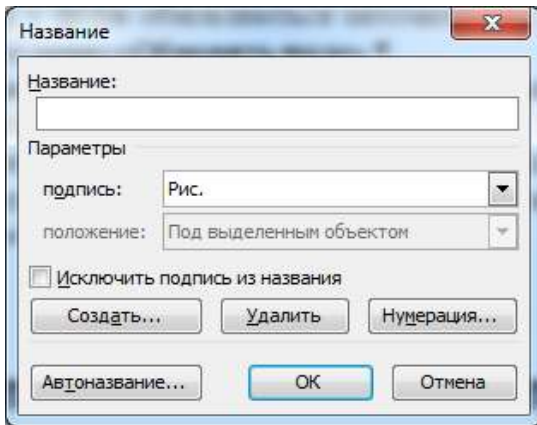


Рис. 8 Діалогове вікно «Название»

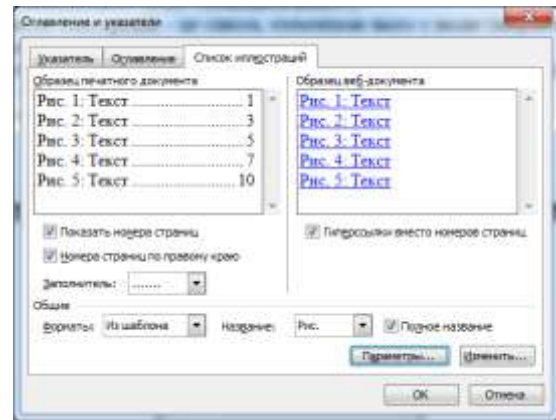


Рис. 9 Діалогове вікно «Оглавление и указатели», вкладка «Список иллюстраций»

Зміст – це список заголовків документа. Він використовується для перегляду тем, обговорюваних у документі, та спрощення переміщення від однієї теми до іншої безпосередньо зі змісту документа.

Найпростіший спосіб створення змісту ґрунтується на використанні стилів заголовків.

Для створення змісту всі заголовки, що становлять зміст, спочатку потрібно визначити стилем *Заголовок*. Заголовок можна створити тільки з цілого абзацу. Можна створювати до дев'яти рівнів заголовків.

Після використання команди **«Вставка»/«Ссылка»/«Оглавление и указатели»/вкладка «Оглавление»** з'явиться відповідне діалогове вікно (Рис. 10). У діалоговому вікні **«Параметры оглавления»** вказуються стилі абзаців, з яких формується зміст та обирається рівень кожного стилю.

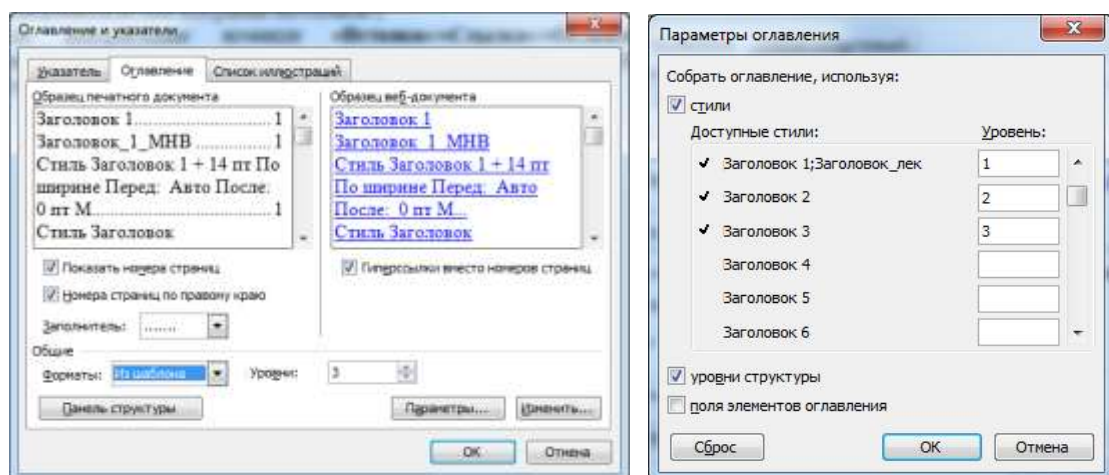


Рис. 10 Діалогові вікна «Оглавление и указатели» та «Параметры оглавления»

1.2.3 Таблиці

Щоб створити будь-яку таблицю, потрібно спочатку сформувати її структуру. Структура таблиці – це спосіб організації її комірок. Комірка – найменша структурна одиниця таблиці.

Є чотири способи створення таблиць:

- 1) перетворення тексту в таблицю; виконується за допомогою команди **«Таблица»/«Преобразовать»/«текст в таблицу»**;
- 2) формування таблиці виконанням команди **«Таблица»/«Добавить таблицу»** або використанням кнопки  панелі інструментів **«Стандартная»**;
- 3) створення таблиці шляхом використання команди **«Таблица»/«Нарисовать таблицу»** або кнопки  панелі інструментів **«Стандартная»**. Під час застосування цього способу активізується панель інструментів **«Таблицы и границы»** (Рис. 11);
- 4) вставка таблиці MS Excel (кнопка  панелі інструментів **«Стандартная»**).



Рис. 11 Панель інструментів «Таблицы и границы»

1.2.4 Діаграми

За допомогою команди **«Вставка»/«Рисунок»/«Диаграмма»** процесор Word надає можливість представити табличні дані у вигляді діаграми, використовуючи для цього вбудований засіб *MS Graph* – редактор ділової графіки. Після виконання цієї команди в документі Word з'являється умовна таблиця MS Excel та її графічне зображення у вигляді стовпчикової діаграми.

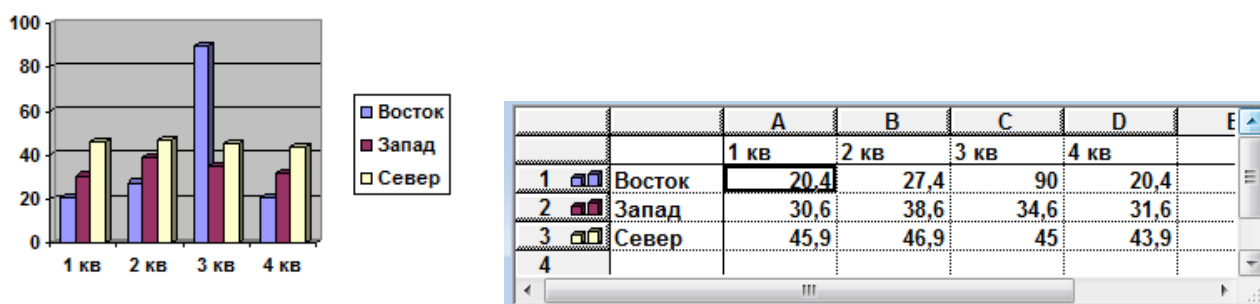


Рис. 12 Умовна таблиця MS Excel та її графічне зображення

В комірки таблиці вносять свій текст і числа, відповідно до тих, що необхідно відтворити на діаграмі. Побудова діаграми виконуватиметься автоматично у міру заповнення таблиці.

Діаграму можна форматувати, використавши команду **«Объект диаграмма»/«Изменить»** контекстного меню.

1.2.5 Формули

В текстовому процесорі Word засобом створення математичних формул є редактор формул *Microsoft Equation 3.0*, для запуску якого необхідно виконати команду «Вставка»/«Объект...»/«Microsoft Equation 3.0» – відкриється панель «Формула» (Рис. 13).

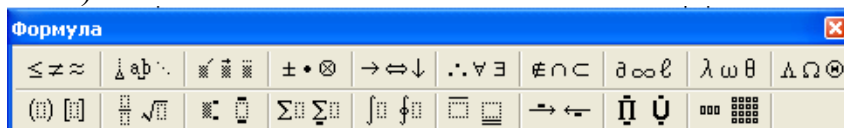


Рис. 13 Панель інструментів «Формула»

Формулу створюють, обираючи на панелі інструментів «Формула» потрібні шаблони.

1.3 Форматування сторінок документа

1.3.1 Параметри сторінки

Форматування здійснюється автоматично з параметрами, що зберігаються в шаблоні *Normal.dot*. Перед створенням нового документу, користувач має можливість приготувати середовище текстового процесора до роботи.

Щоб розмістити текст документа на папері, необхідно задати параметри сторінки за допомогою команди «Файл»/«Параметры страницы» (Рис. 14):

- розміри полів;
- орієнтацію сторінок документа;
- властивості сторінок;
- розмір паперу;
- розташування колонтитулів;
- графічне оформлення сторінок і т.д.

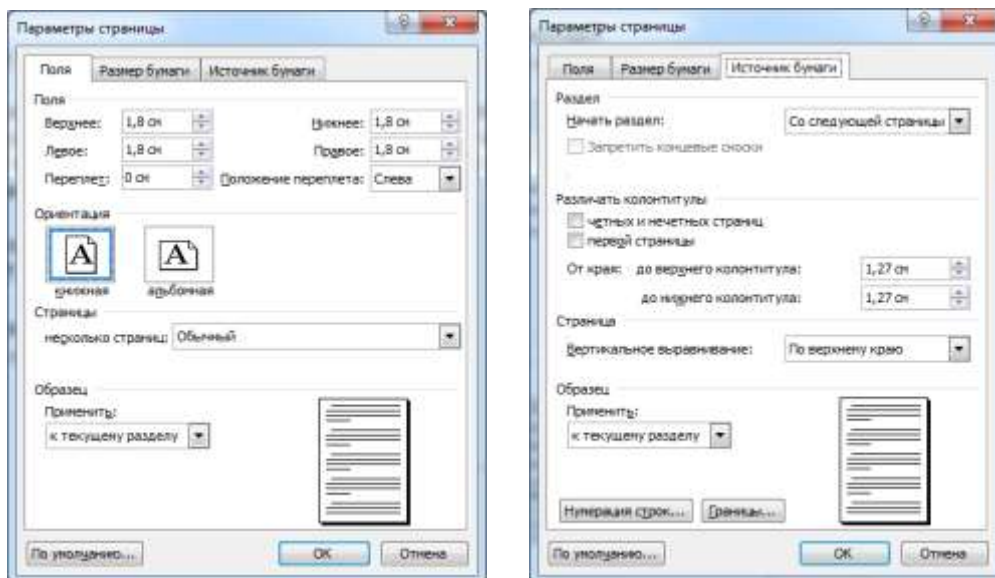


Рис. 14 Діалогове вікно «Параметры станицы»

1.3.2 Автоматична перевірка правопису

Мова, на якій буде вводиться текст документа, обирається за допомогою команди «Сервис»/«Язык»/«Выбрать язык» (Рис. 15).

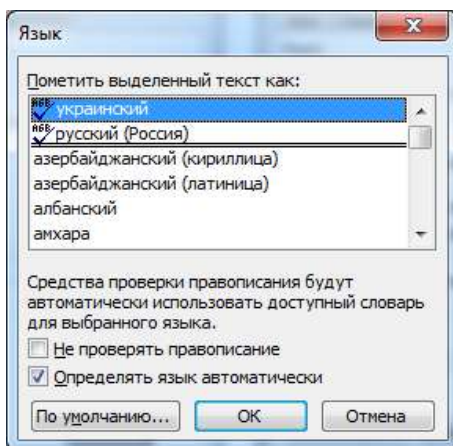


Рис. 15 Діалогове вікно «Язык»

Для перевірки правопису вибирається команда «Сервис»/«Правописание» або кнопка , що запустить процедуру перевірки орфографії виділеного фрагменту тексту, або всього документа. Розбіжності і пропонувані заміни Word видає на екран у діалоговому вікні (Рис. 16).

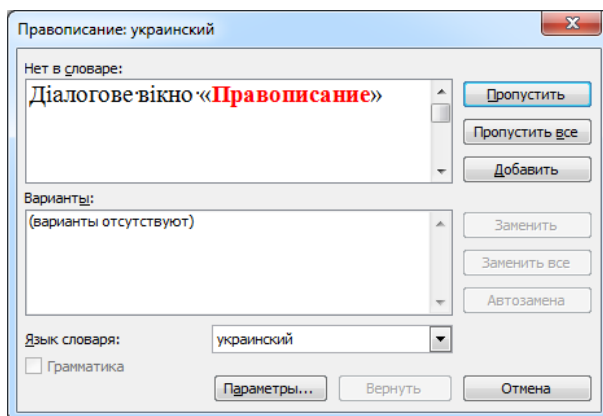


Рис. 16 Діалогове вікно «Правописание»

У разі необхідності можна застосувати засоби автоматичного перенесення слів. Для цього виконується команда «Сервис»/«Язык»/«Расстановка переносов» (Рис. 17)

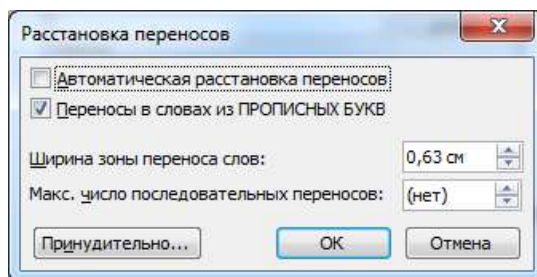


Рис. 17 Діалогове вікно «Расстановка переносов»

Важливі параметри встановлюють в діалоговому вікні, що викликається командою «Сервіс»/«Параметри». Це діалогове вікно містить багато вкладок, деякі з них показані на (Рис. 18).

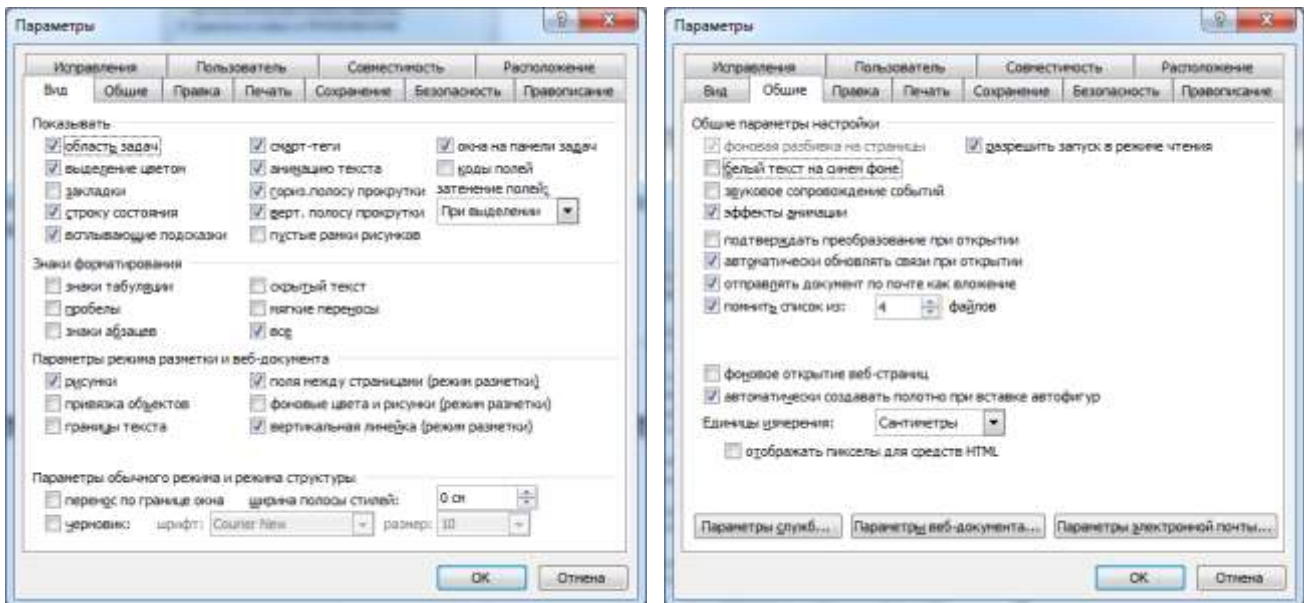


Рис. 18 Діалогове вікно «Параметри»

1.3.3 Розділи

Розділ – це структурна одиниця текстового документа, яка об'єднує певну його частину за змістом і зазвичай має свій заголовок. Щоб створити у тексті новий розділ, потрібно встановити текстовий курсор в кінці попереднього розділу і виконати команду «Вставка»/«Разрыв» (Рис. 19).

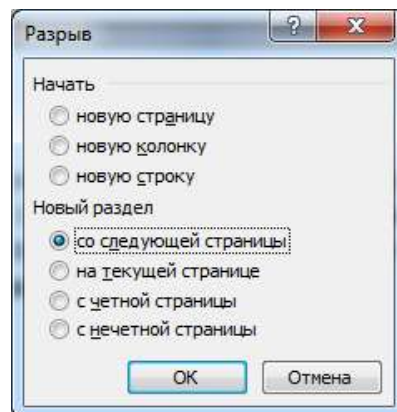


Рис. 19 Діалогове вікно «Разрыв»

Новий розділ може мати своє форматування усіх рівнів.

1.3.4 Виноски

Виноски – це символи, що використовуються у тексті у вигляді верхніх індексів, пов'язаних із певним, зазвичай роз'яснювальним текстом або текстом, який коментує (звичайні виноски), або як посилання на інші документи (в більшості випадків це кінцеві виноски).

Щоб додати виноску, потрібно встановити текстовий курсор у потрібному місці тексту і виконати команду «Вставка»/«Ссылка»/«Сноска...» (Рис. 20).

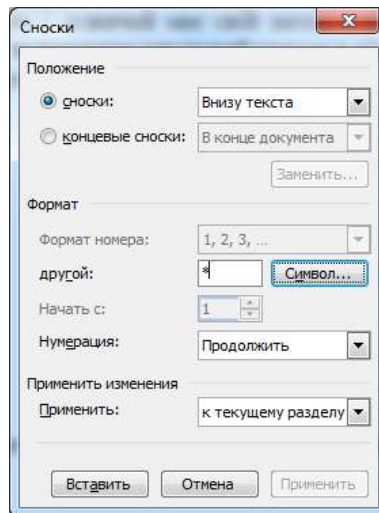


Рис. 20 Діалогове вікно «Сноски»

1.3.5 Колонтитули

Колонтитул – текст або графічні об’єкти, або те й інше, розташовані вгорі та внизу стосовно основного тексту (в іншому шарі документа) і повторюються на кількох або всіх сторінках розділу чи всього документа. З метою створення колонтитулів потрібно виконати команду «Вид»/«Колонтитулы», після якої відкриється панель інструментів «Колонтитулы» (Рис. 21).

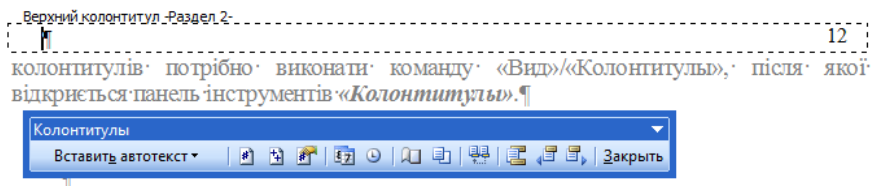


Рис. 21 Панель інструментів «Колонтитулы»

1.3.6 Нумерація сторінок

Перед друком документа його сторінки нумерують. Для цього використовується команда «Вставка»/«Номера страниц...», після виконання якого з’явиться діалогове вікно (Рис. 22).

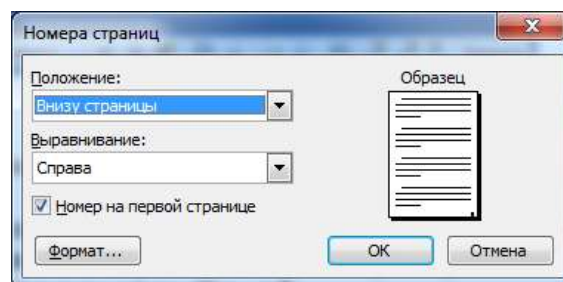


Рис. 22 Діалогове вікно «Номера страниц»

Контрольні запитання

1. Назвіть режими відображення документа Word і яке їх призначення.
2. Як задати параметри сторінки?
3. Як задати мову для перевірки правопису?
4. Які типи стилів використовуються у Word?
5. Як створити новий стиль?
6. Поняття колонтитул. Як створити колонтитул?
7. Яким чином створити в текстовому документі математичну формулу?
8. Скільки рівнів заголовків можна створити у документі Word?
9. Як створити зміст документа, використовуючи засоби автоматизації Word?
10. Як побудувати діаграму на основі табличних даних?

2 ОБРОБКА ТАБЛИЧНО-ОРГАНІЗОВАНОЇ ІНФОРМАЦІЇ

2.1 Основні поняття електронних таблиць

Для представлення даних в зручному вигляді часто використовують таблиці. Особливість електронних таблиць, у тому числі Excel, полягає в можливості використання формул для визначення зв'язків між значеннями різних комірок. Розрахунки за формулами виконуються автоматично.

Додаток Excel дозволяє ефективно систематизувати дані, проводити їх автоматичну обробку, аналізувати і структурувати дані, а також представляти результат у наочному вигляді за допомогою різноманітних діаграм.

2.1.1 Робоча книга, робочий аркуш

Документ Excel називається **робочою книгою**, яка представляє набір **робочих аркушів**, що мають табличну структуру.

На перетині стовпчиків та рядків – **комірка** з відповідною адресою (наприклад, G15, A31) (Рис. 23). Одна з комірок завжди є **активною** і виділяється рамкою активної комірки, яка в Excel є курсором. Комірка на неактивному робочому аркуші ідентифікується ім'ям аркуша та її адресою на аркуші, наприклад **Лист2!A1**

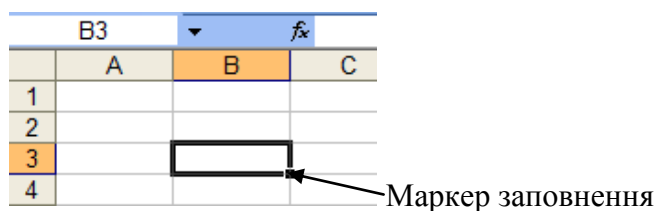


Рис. 23 Активна комірка

На дані, що розташовані у сусідніх комірках, можна посилатися у формулах як на єдине ціле. Таку групу комірок називають діапазоном (наприклад A3:F17 – суміжний діапазон, B4:D8; F5:G19 – несуміжний діапазон)

2.1.2 Форматування вмісту комірок

Окрема комірка може містити дані таких типів:

- текст;
- число;
- формула.

Тип даних, що розміщуються у комірку, визначається автоматично.

Якщо необхідно змінити формат даних у поточній комірці або діапазоні використовується команда «**Формат**»/«**Ячейки**» (Рис.24, 25). Вкладки цього діалогового вікна «**Формат ячеек**» дозволяють обирати формат запису даних (кількість знаків після коми, визначення грошової одиниці, спосіб запису дати і т.д.), задавати напрям тексту і метод його вирівнювання, визначати шрифт і накреслення символів, управляти відображенням і видом границь, задавати колір фону.

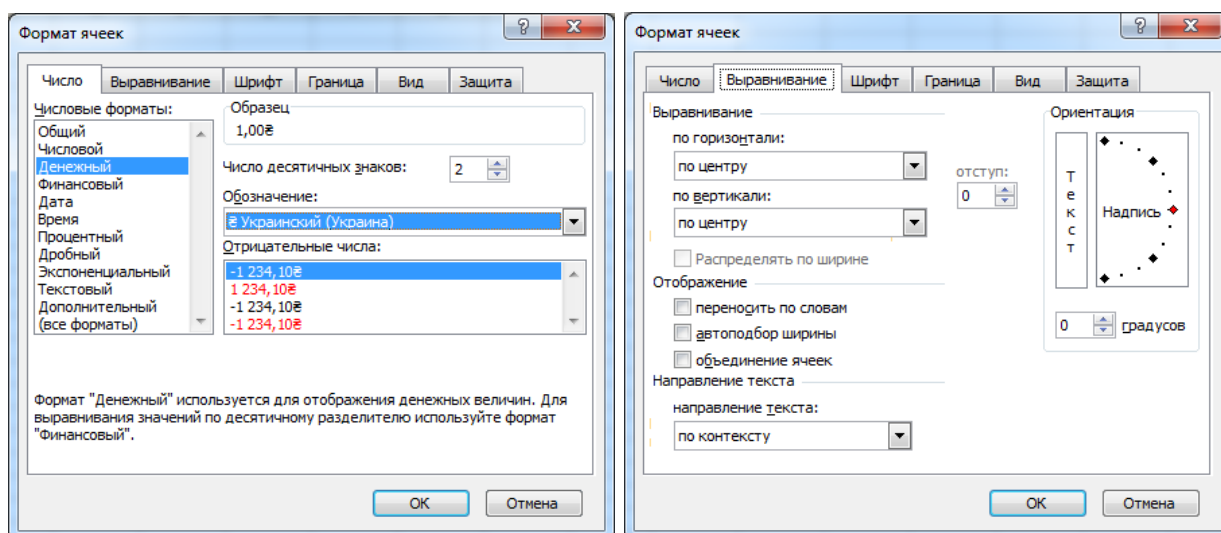


Рис. 24 Зміна формату комірки і методу вирівнювання вмісту комірки

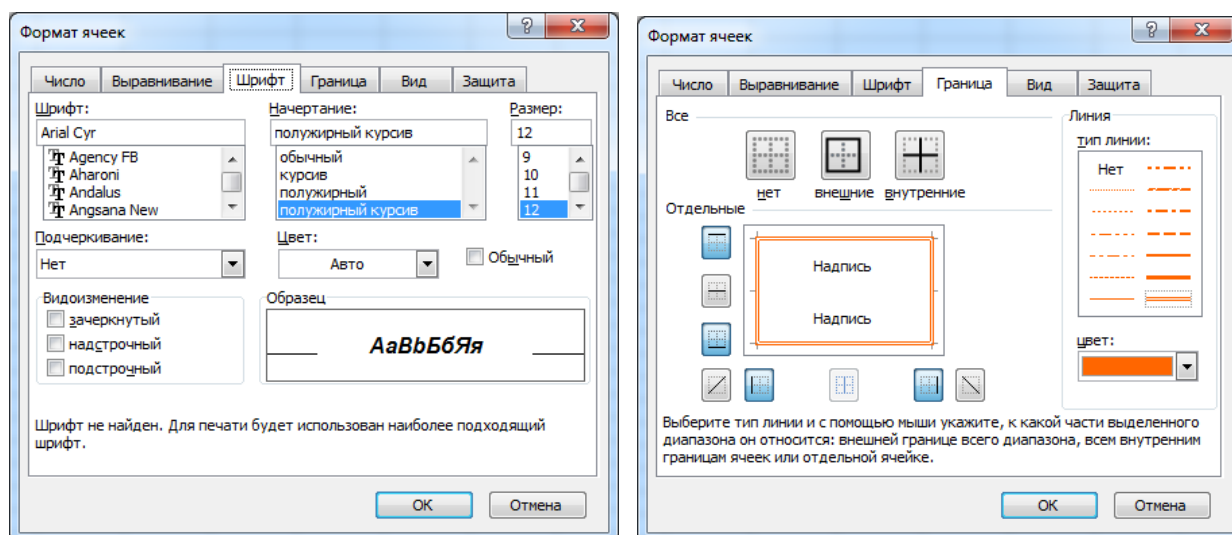


Рис. 25 Визначення шрифту, відображення границь

Нижче наведено приклад форматування діапазону комірок за допомогою команди «**Формат**»/«**Ячейки**» і використання вищенаведених вкладок.

	A	B	C	D
1	До форматування		Після форматування	
2	1,00		1,00₴	
3	2,00		2,00₴	
4	2,00		2,00₴	
5	5,00		5,00₴	
6	6,00		6,00₴	
7				

2.2 Обчислення в Excel

2.2.1 Формули, функції

Обчислення в таблицях Excel здійснюються за допомогою формул. **Формула** Excel завжди починається з символу **=**, і може містити числові константи, посилання на комірки і функції Excel, з'єднані знаками математичних операцій. Дужки дозволяють змінювати стандартний порядок виконання дій. Якщо комірка містить формулу, на робочому аркуші відображається поточний результат обчислення цієї формули. Якщо зробити комірку поточною, то формула з'явиться у рядку формул (Рис. 26)

	B1			=SIN(A1+0,7)-EXP(A1)		
	A	B	C	D	E	
1	0	-0,35578				
2	0,1	-0,38781				
3	0,2	-0,43808				
4	0,3	-0,50839				
5	0,4	-0,60062				
6	0,5	-0,71668				

Рис. 26 Приклад формули

Стандартні (вбудовані функції) Excel призначені для виконання різноманітних обчислювальних та логічних процедур.

Всі функції мають однаковий стандартний формат:

Ім'я_функції(аргумент_1;аргумент_2;...аргумент_n)

В якості аргументів функції, як правило, виступають адреси комірок, що містять відповідні значення. Якщо в якості аргументу функції виступає інша функція, то говорять про **вкладену функцію**

Функції в Excel поділяються на категорії: математичні, статистичні, фінансові, дати і часу, текстові, логічні та ін. Також їх поділяють на такі види: без аргументу, з одним аргументом, з фіксованою кількістю аргументів, з невизначеною кількістю аргументів, з необов'язковими аргументами.

Суттєву допомогу користувачеві в процесі конструювання формул, може надати **Майстер функцій** (Рис. 27): команда «Вставка»/«Функция» або кнопка  у рядку формул.

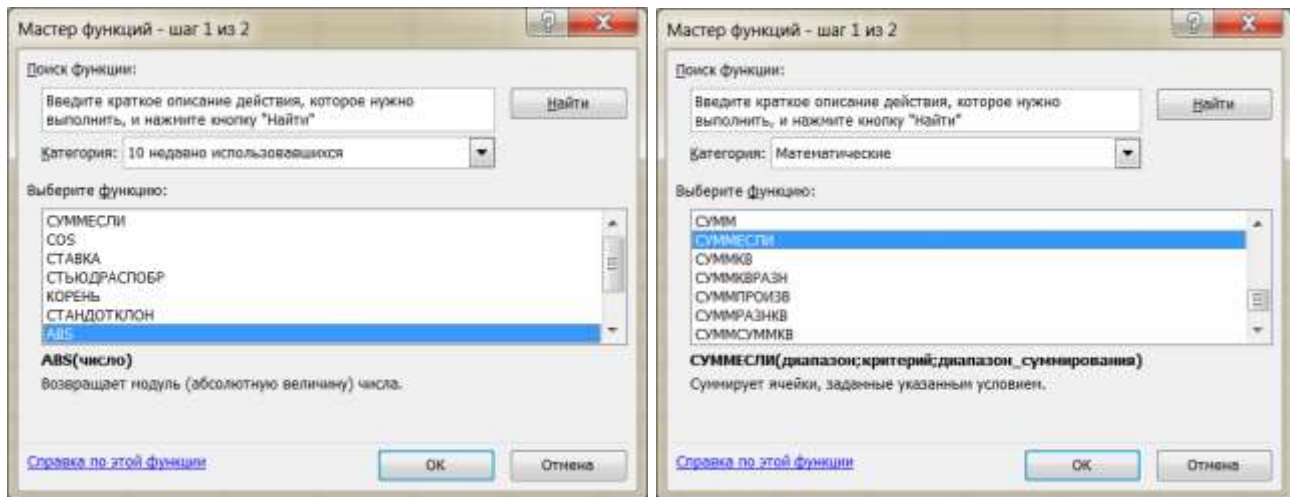


Рис. 27 Діалогові вікна Майстра функцій

На другому кроці Майстра функцій з'являється діалогове вікно (рис. 28) для роботи з обраною функцією, наприклад, з функцією СУММЕСЛИ:

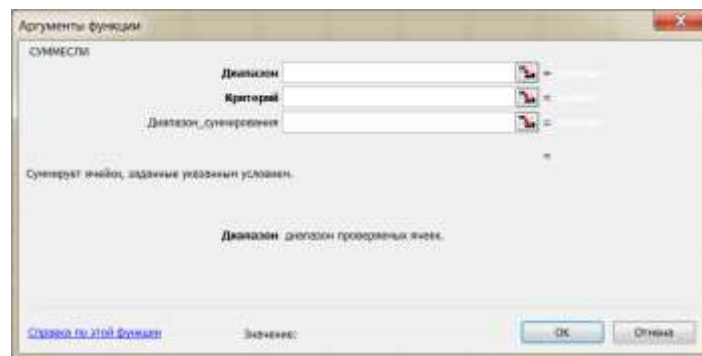


Рис. 28 Діалогове вікно «Аргументы функции»

В формулах Excel використовуються різноманітні **математичні** функції. Використання цих функцій дозволяє прискорити та спростити процес обчислень. В якості аргументів більшості функцій виступають числові значення, які можна представити в вигляді математичних виразів з використанням констант і посилань.

У табл. 1 представлено записи основних математичних функцій в Excel.

Таблица 1

Запис математичних функцій в Excel

Функція в Excel	Математична функція
ABS(число)	$ число $
ACOS(число)	$arccos(число)$
ASIN(число)	$arcsin(число)$
ATAN(число)	$arctg(число)$
COS(число)	$cos(число)$
COSH(число)	$ch(число)$
EXP(число)	$e^{число}$
LN(число)	$ln(число)$

Функція в Excel	Математична функція
LOG(число;основание)	$\log_{\text{основание}}(\text{число})$
LOG10(число)	$\lg(\text{число})$
SIN(число)	$\sin(\text{число})$
SINH(число)	$\text{sh}(\text{число})$
TAN(число)	$\text{tg}(\text{число})$
ЗНАК(число)	$\text{sign}(\text{число})$
КОРЕНЬ(число)	$\sqrt{\text{число}}$
СТЕПЕНЬ(число;степень)	$\text{число}^{\text{степень}}$

У табл. 2 представлено приклади формул для обчислення математичних виразів в Excel. В записі формули Excel замість змінної x використано посилання на комірку A2.

Таблиця 2

Приклади формул

Математичний запис	Формула в Excel
$ x $	=ABS(A2)
$\sin x$	=SIN(A2)
$\cos(x-2)$	=COS(A2-2)
$\text{tg}x$	=TAN(A2)
$\text{ctgx} = \frac{\cos x}{\sin x}$	=COS(A2)/SIN(A2)
$\arccos x$	=ACOS(A2)
e^{5x-2}	=EXP(5*A2-2)
$\ln x-4 $	=LN(ABS(A2-4))
$\sqrt{1-\cos^2(x)} = (1-\cos^2 x)^{\frac{1}{2}}$	=(1-COS(A2)^2)^(1/2) або =КОРЕНЬ(1-COS(A2)^2)
$e^{\cos x \sin x+5}$	=EXP(COS(A2)*SIN(A2)+5)

Для проведення логічних обчислень в Excel використовують шість логічних функцій: ЕСЛИ, И, ИЛИ, ИСТИНА, ЛОЖЬ, НЕ.

Логічна функція **ЕСЛИ** має наступний синтаксис:

ЕСЛИ (лог_выражение;значение_если_истина; значение_если_ложь)

і повертає одне значення. Якщо **лог_выражение** – істина, то це результат обчислення **значение_если_истина**, у протилежному випадку це –

значення_если_ложь. Наприклад, на Рис.29 представлено результат обчислення значення функції:

$$y = \begin{cases} \sin x, & \text{если } x > 4 \\ \cos x, & \text{если } x \leq 4 \end{cases}$$



Рис. 29 Приклад використання функції ЕСЛИ

2.2.2 Посилання на комірки

Формула може містити **посилання**, тобто адреси комірок, вміст яких використовується в обчисленнях. Це означає, що результат обчислення формули залежить від числа, що знаходиться в іншій комірці.

Види посилань:

1. **Відносні** посилання (Рис.30). При копіюванні формули адреси в посиланнях автоматично змінюються відповідно до відносного розташування вихідної комірки і копії що створюється. Посилання A1, A2, A3, B1, B2, B3, C1, D1 – відносні.

	F1					
	A	B	C	D	E	F
1	1	5	=A1+B1	=B1+C1		=C1+D1
2	2	6	=A2+B2			
3	3	7	=A3+B3			
4						

Рис. 30 Приклад відносних посилань

	F1					
	A	B	C	D	E	F
1	1	5		6	5	6
2	2	6		8		
3	3	7		10		

Рис. 31 Приклад роботи формули з відносними посиланнями

2. **Абсолютні** посилання (Рис.32). При абсолютній адресації адреси посилань при копіюванні не змінюються. Елементи номера комірки, що використовують абсолютні посилання, випереджаються символом \$. \$A\$1 – абсолютне посилання на комірку A1. Excel дозволяє використовувати в якості абсолютного посилання на комірку не тільки її адресу, але і привласнене цій комірці ім'я. Для присвоєння імені активній комірці виконується команда «Вставка»/«Имя»/«Присвоить».

	F1						
		A	B	C	D	E	F
1		1	5		=A\$1+B\$1	=A\$1+B\$1	=A\$1+B\$1
2		2	6		=A\$1+B\$1		
3		3	7		=A\$1+B\$1		

Рис. 32 Приклад абсолютних посилань

	F1						
		A	B	C	D	E	F
1		1	5		6	6	6
2		2	6		6		
3		3	7		6		

Рис. 33 Приклад роботи формули з абсолютними посиланнями

3. **Змішані** посилання. При копіюванні формули змінюється адреса або рядка, або стовпця. Наприклад, посилання **A\$5**, **\$A1** – змішані

	F1						
		A	B	C	D	E	F
1		1	5		=A\$1+B\$1	=B\$1+B\$1	=C\$1+B\$1
2		2	6		=A\$1+B\$2		
3		3	7		=A\$1+B\$3		

Рис. 34 Приклад змішаних посилань

	F1						
		A	B	C	D	E	F
1		1	5		6	10	5
2		2	6		7		
3		3	7		8		

Рис. 35 Приклад роботи формули зі змішаними посиланнями

Для зміни способу адресації при редагуванні формули треба виділити посилання на комірку і натиснути клавішу **F4**. Наприклад, при послідовному натисканні клавіші **F4** посилання на комірку **A1** буде записуватися: **A1**, **\$A\$1**, **A\$1**, **\$A1**.

2.3 Графічне представлення даних в Excel

В Excel термін діаграма використовується для позначення усіх видів графічного представлення числових даних. Побудова графічного зображення робиться на основі *ряду даних*. Так називають групу комірок із даними в межах окремого рядка або стовпця. На одній діаграмі можна відображати декілька рядів даних.

Категорії діаграми задають положення відповідних значень у ряді даних.

2.3.1 Побудова діаграм

Для побудови діаграм використовується **Майстер діаграм**, що запускається за допомогою кнопки **Майстер діаграм**  на панелі інструментів «Стандартная» або командою «Вставка»/«Діаграма». Перед тим, як

будувати діаграму можна виділити відповідний діапазон, але задати цю інформацію можна й в процесі роботи майстра.

Робота із **Майстром діаграм** складається з чотирьох кроків:

- вибір типу і виду діаграми;
- вибір діапазону для побудови діаграми (вкладка «Диапазон данных»); на вкладці «Ряд» можна переглянути імена і значення рядів даних, додати або видалити ряди, перевірити підписи осі X та внести необхідні зміни;
- вибір оформлення діаграми;
- вибір розміщення діаграми.

2.3.2 Редагування діаграм

Готову діаграму можна змінити. Вона складається з окремих елементів, таких, як самі графіки (ряди даних), осі координат, заголовок діаграми, область побудови, заголовки осей, легенда, підписи даних. При натисканні на елемент діаграми, він виділяється маркерами, а при наведенні на нього курсору миші – описується спливаючою підказкою. Елементи діаграми представлено на Рис.36.

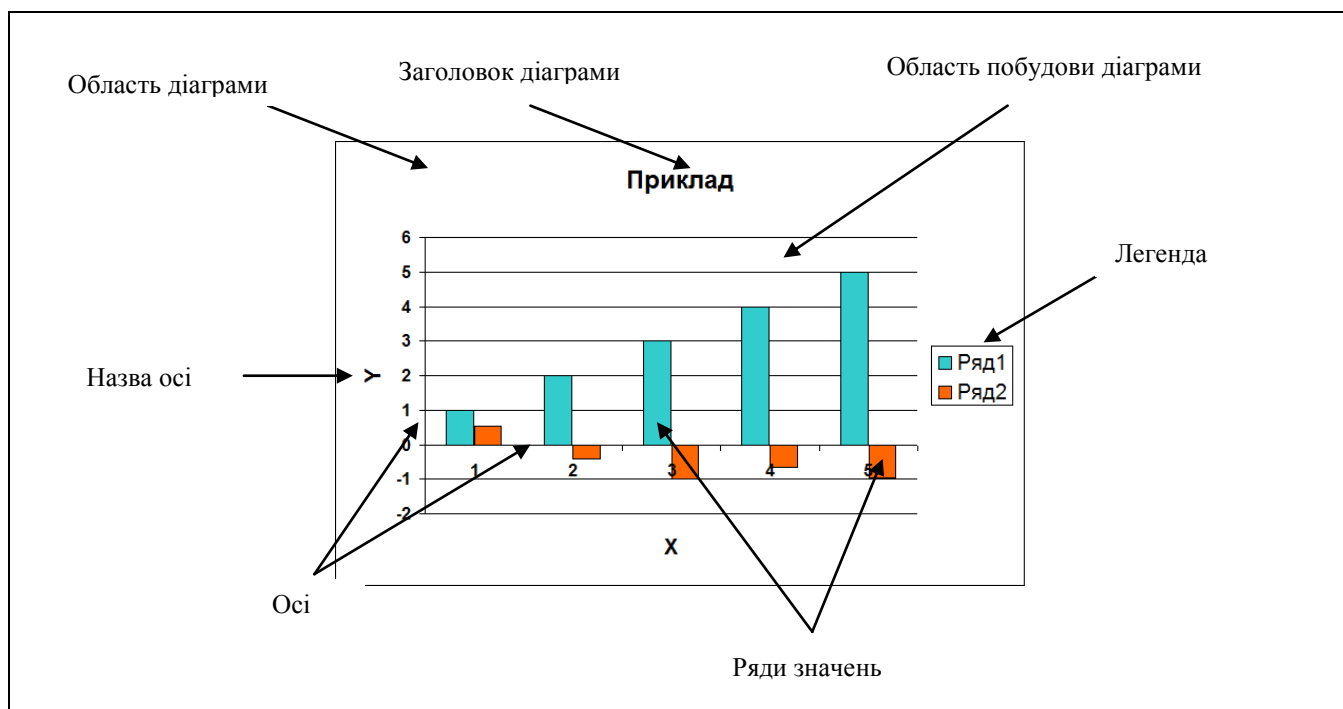


Рис. 36 Області діаграми

Діалогове вікно для форматування елемента діаграми можна відкрити, використовуючи меню «**Формат**» (для виділеного елемента) або контекстне меню (команда «**Формат**»).

Контрольні запитання

1. Що таке робочий аркуш, комірка, діапазон?
2. Як відтворити формули в комірках?
3. Як задається відносне посилання на комірку?
4. Як задати абсолютне посилання на комірку?

5. Яке призначення має кнопка  на панелі інструментів «Стандартная»?
6. Чи змінюються відносні та абсолютні посилання при копіюванні формули?
7. Які є різновиди числових форматів у MS Excel?
8. Який синтаксис має вбудована функція Excel?
9. Яким чином можна побудувати діаграму?
10. Які задачі можна розв'язувати, використовуючи надбудову «Поиск решения»?

3 РОЗРОБКА WEB-САЙТУ ЗАСОБАМИ МОВИ HTML

3.1 Елементи, атрибути HTML-документа

HTML (Hypertext Markup Language) – мова гіпертекстової розмітки. Ця мова дозволяє задавати спосіб представлення на Web-сторінках тексту, зображень, відео файлів або електронної інформації будь-якого іншого типу.

Будь-який HTML-документ – це набір елементів, що описують окремі складові документа, такі як заголовки, списки, абзаци тексту, таблиці та ін.

Найчастіше елемент розмітки складається з трьох частин: початкового і кінцевого компонентів, між якими розміщуються текст або інші елементи документа. Цими компоненти є спеціальні конструкції, що управляють, для розмітки вмісту HTML-документу і називаються **тегами**.

HTML-елементом називається комбінація начального тегу, кінцевого тегу та вмісту. Приклади HTML-елементів:

`<P>` це абзац `</P>`

`<I>` це курсивний текст `</I>`

HTML не чутливий до регістра – це означає, що тег `<b>` інтерпретуватиметься браузером так само як і тег `<B>`.

Загальний синтаксис написання тегів:

`<тег атрибут1="значення" атрибут2="значення">` – одинарний тег

`<тег атрибут1="значення" атрибут2="значення">...</тег>` – парний тег (контейнер)

Одинарний тег використовується самостійно, а парний може включати інші теги або текст. Щоб розширити можливості окремих тегів і гнучкіше управляти вмістом контейнерів застосовуються **атрибути** тегів, які розділяються між собою пробілами.

Атрибути задаються в початковому тегу елементів і складаються з імені і значення, наприклад в атрибуті `href="http://www.microsoft.com"` `href` є ім'ям, а `http://www.microsoft.com` значенням.

Порядок атрибутів в будь-якому тегу не має значення і на результат відображення елементу не впливає.

Згідно специфікації HTML всі значення атрибутів тегів слід вказувати в подвійних ("приклад") або одинарних лапках ('приклад').

3.2 Структура HTML-кода

Якщо відкрити будь-яку Web-сторінку, то вона міститиме в собі типові елементи, які не змінюються від вигляду і спрямованості сайту.

В наступному прикладі показано код простого документу, що містить основні теги:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
                                "http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <title>Приклад Web-сторінки </title>
  </head>
  <body>
    <h1>Заголовок</h1>
    <!-- Коментар -->
    <p>Перший абзац.</p>
    <p>Другий абзац.</p>
  </body>
</html>
```

Елемент `<!DOCTYPE>` призначений для того, щоб вказати тип поточного документа – DTD (document type definition, опис типу документа). Це необхідно, щоб браузер розумів, як слід інтерпретувати поточну Web-сторінку, адже HTML існує в декількох версіях.

Тег `<HTML>` визначає початок HTML-документу, усередині нього зберігається заголовок (`<HEAD>`) і тіло документа (`<BODY>`).

3.3 Дизайн тексту для Web-сторінок

Основні **одинарні теги** для розміщення тексту:

`<P>` – початок нового абзацу, але його прийнято записувати в кінці попереднього. Наступне речення починатиметься з нового абзацу без відступу, вирівняного по лівому краю.

`<BR>` – наступний за цим тегом текст буде наведено у новому рядку без пропуску рядка.

`<HR>` – буде проведена горизонтальна лінія.

Парні теги форматування символів тексту:

`<B> текст </B>` – жирний шрифт тексту;

`<I> текст </I>` – курсивний шрифт;

`<U> текст </U>` – підкреслений шрифт;

`<BIG> текст </BIG>` – великий шрифт;

`<SMALL> текст </SMALL>` – малий шрифт;

`<B><I> текст </I></B>` – жирний курсив (демонструє застосування принципу вкладених тегів).

Тег `<P>` може використовуватись як парний тег.

Теги заголовків:

`<H1> Заголовок 1 </H1>` – заголовок першого рівня;

`<H2> Заголовок 2 </H2>` – заголовок другого рівня;

`<H3> Заголовок 3 </H3>` – заголовок третього рівня і т.д.

Чим вище рівень заголовка, тим більше розмір шрифту. Самим верхнім рівнем є рівень 1 (`<H1>`), а самим нижнім – рівень 6 (`<H6>`). Заголовок за замовчуванням вирівнюється по лівому краю вікна.

Якщо вирівнювання заголовка чи іншого елемента на сторінці потрібно задати явно, то використовують **теги вирівнювання**:

`<CENTER> елемент </CENTER>` – вирівнювання по центру;

`<LEFT> елемент </LEFT>` – вирівнювання по лівому краю;

`<RIGHT> елемент </RIGHT>` – вирівнювання по правому краю.

Тег визначення параметрів **шрифту** `<FONT>...</FONT>`. З його допомогою можна вказати тип, розмір, колір:

`SIZE` = "число від 1 до 7" – розмір шрифту; за замовчуванням розмір шрифту є три пункти, в дизайні використовуються перші чотири розміри;

`FACE` = "назва шрифту" – тип шрифту; за замовчуванням браузері використовують шрифт TNr;

`COLOR` = "колір" – колір шрифту. Кольори можна задавати двома різними способами: можна використовувати словесне позначення кольору *black*, *white*, *red*, *blue* і т.д., а можна – і їх цифрове позначення, лише перед цифровим еквівалентом необхідно ставити #. Так, колір "black" має числове позначення "#000000", колір "red" – позначення "#FF0000". Приклади використання тегів `<FONT>...</FONT>`:

Текст 1 – `<font size="1" face="Comic Sans MS" color="red">Текст 1</font>`

Текст 2 – `<font size="4" face="Arial" color="#FF0000">Текст 2</font>`

Текст 3 – `<font size="5" face="Courier" color="#FFFFFF">Текст 3</font>`

За допомогою тегу `<MARQUEE атрибути>...</MARQUEE>` можна створити ефект рухомого тексту або графічного об'єкту у смугі.

3.4 Створення списків та їх дизайн

Списком називається взаємозв'язаний набір окремих фраз або пропозицій, які починаються з маркера або цифри. Списки надають можливість упорядкувати і систематизувати різні дані і представити їх в наочному і зручному для користувача вигляді.

Список може мати заголовок, який охоплюється тегамі `<LN>...</LN>`.

Маркований список утворюється за допомогою парного тегу `<UL>...</UL>` і одинарних тегів `<LI>`. Приклад:

```
<ul>
<li>Перший елемент</li>
<li>Другий елемент</li>
<li>Третій елемент</li>
</ul>
```

Результат:

- Перший елемент
- Другий елемент
- Третій елемент

Нумерований список утворюється за допомогою парного тегу `<OL>...</OL>` з необов'язковим параметром `TYPE` і одинарних тегів `<LI>`. Приклад:

```
<ol>
<li>Перший елемент</li>
<li>Другий елемент</li>
<li>Третій елемент</li>
</ol>
```

Результат:

1. Перший елемент
2. Другий елемент
3. Третій елемент

Тип маркера може задаватися за допомогою параметра `TYPE` тегу `<OL>`. В якості маркерів можуть виступати:

- арабські числа (1, 2, 3, ...) – `type="1"`;
- римські числа у верхньому регістрі (I, II, III, ...) – `type="I"`;
- римські числа в нижньому регістрі (i, ii, iii, ...) – `type="i"`;
- великі латинські букви (A, B, C, ...) – `type="A"`;
- малі латинські букви (a, b, c, ...) – `type="a"`.

Приклад:

```
<ol type = "A">
<li>Перший елемент</li>
<li>Другий елемент</li>
<li>Третій елемент</li>
</ol>
```

Результат:

- A. Перший елемент
- B. Другий елемент
- C. Третій елемент

Список визначень. Список визначень складається з термінів і їх визначень. Такий список задається за допомогою тегів `<DL>...</DL>`. Терміни в такому списку задаються тегом `<DT>`, а їх визначення тегом `<DD>`. Нижче приведена структура списку визначень.

```
<dl> <dt>Перший елемент</dt>
<dd> Опис першого елементу </dd>
<dt>Другий елемент</dt>
<dd>Опис другого елементу</dd>
<dt>Третій елемент</dt>
<dd>Опис третього елементу</dd>
</dl>
```

Результат:

- Перший елемент
Опис першого елементу
- Другий елемент
Опис другого елементу
- Третій елемент
Опис третього елементу

3.5 Створення таблиць та їх дизайн

Таблиці створюють за допомогою тегів

`<TABLE атрибути> <TC> Заголовок таблиці </TC> </TABLE>`.

Тег `<TABLE>...</TABLE>` є таким, що визначає таблицю, і в нього вкладаються рядки за допомогою тегу `<TR>...</TR>`, а в рядки вкладаються стовпці за допомогою тегу `<TD>...</TD>`.

Тег `<TH> текст </TH>` формує комірку-заголовок рядка чи стовпця.

Приклад створення таблиці 2x2.

```
<table width="200" border="1">
<tr>
<td> 1 </td>
<td> 2 </td>
</tr>
<tr>
<td> 3 </td>
<td> 4 </td>
</tr>
</table>
```

Результат:

1	2
3	4

Тег `<TABLE>` може містити такі атрибути:

- width – ширина таблиці в пікселях
`<table width="значення">...</table>;`
- border – товщина рамки таблиці в пікселях
`<table border="товщина">...</table>;`
- align – визначає вирівнювання таблиці
`<table align="left | center | right">...</table>;`
- background – задає рисунок фону в таблиці
`<table background="URL">...</table>;`
- bgcolor – колір фону таблиці
`<table bgcolor="колір">...</table>;`
- bordercolor – колір рамки
`<table bordercolor="колір">...</table>;`
- cellpadding – відступ від рамки до вмісту комірки
`<table cellpadding="значення">...</table>;`
- cellspacing – відстань між комірками
`<table cellspacing="значення">...</table>;`
- height – висота таблиці
`<table height="значення">...</table>.`

3.6 Додавання графічних та відео файлів

Комбінуючи зображення і звичайний текст можна представляти інформацію максимально ефективно.

Тег `<IMG>` призначений для відображення на Web-сторінці зображень в графічному форматі *.gif*, *.jpeg* або *.png*. Адреса файлу з рисунком задається за

допомогою атрибуту SRC. Якщо необхідно, то рисунок можна зробити посиланням на інший файл, розмістивши тег до контейнеру <A>.

Синтаксис тегу :

```
<IMG SRC="адреса графічного файлу" ALT="альтернативний текст">
```

Обов'язковим є лише перший атрибут SRC.

Основні атрибути тегу :

- alt – альтернативний текст, який виводитиметься замість зображення, якщо браузер не може прийняти графічний файл або якщо режим відображення графіки вимкнено

```
<img alt="текст">;
```

- align – задає місце розташування графічного об'єкту на екрані

```
<img align="bottom | left | middle | right | top">;
```

- width – задає розмір графічного об'єкту за шириною в пікселях або відсотках сторінки

```
<img width="значення">;
```

- height – задає розмір графічного об'єкту за висотою в пікселях або відсотках сторінки

```
<img height="висота">;
```

- border – товщина рамки в пікселях навколо зображення

```
<img border="товщина рамки">
```

За допомогою тегу можна вставити також відеофільм, який запускатиметься в момент відкриття Web-сторінки:

```
<IMG DYN SRC="адреса відеофайла">
```

3.7. Додавання гіперпосилань

Посилання є основою гіпертекстових документів і дозволяють переходити з однієї Web-сторінки на іншу. Особливість їх полягає в тому, що посилання може вести не лише на HTML-файл, але і на файл будь-якого типа, причому цей файл може розміщуватися зовсім на іншому сайті.

Для створення посилання необхідно повідомити браузеру, що є посиланням, а також вказати адресу документа, на який слід зробити посилання. Обидві дії виконуються за допомогою тегу <A>.

Загальний синтаксис створення посилань:

```
<A HREF="URL"> текст посилання </A>
```

Атрибут HREF визначає URL (Universal Resource Locator, універсальний покажчик ресурсу), тобто, адресу документа, на яку слід перейти, а вміст контейнера <A> є посиланням.

Адреса посилання може бути як абсолютною, так і відносною. Абсолютні адреси повинні починатися з назви протоколу (зазвичай http://) і містити ім'я сайту. Відносні посилання ведуть відлік від кореня сайту або поточного документа.

Наприклад, посилання виду Сайт Microsoft є абсолютним і веде на головну сторінку сайту Microsoft.

Відносно посилання вказує на адресу сторінки відносно поточного документа і має, наприклад, вигляд `<a href="page.html">Задача</a>`.

За замовчуванням, при переході за посиланням, документ відкривається в поточному вікні або фреймі. При необхідності ця умова може бути змінена атрибутом `target` – вказує ім'я вікна або фрейму, куди браузер завантажуватиме документ

```
<a target="ім'я вікна">...</a>
```

В якості значення використовується ім'я вікна або фрейму, що задається атрибутом `name`. Якщо встановлено неіснуюче ім'я, то буде відкрито нове вікно. В якості зарезервованих застосовуються наступні імена:

- `_blank` – завантажує сторінку в нове вікно браузера;
- `_self` – завантажує сторінку в поточне вікно (це значення задається за замовчуванням, тобто `target="_self"`);
- `_parent` – завантажує сторінку у батьківський фрейм; якщо фреймів немає, то це значення працює як `_self`;
- `_top` – відмінює всі фрейми і завантажує сторінку в повному вікні
- `_blank` – завантажує сторінку в нове вікно браузера.

В якості гіперпосилань можна використовувати картинки. Для цього усередині тегу `<a>...</a>` необхідно додати тег `<img>`: Наприклад,

```
<a href="page.html"></a>
```

3.8 Використання фреймів

Фрейм – це частина вікна браузера (може бути оформлена рамкою), в якій можна переглядати окремо від інших частин певні Web-сторінки. Поділ вікна браузера на фрейми виконується за допомогою спеціальних тегів.

Парний тег `<FRAMESET атрибути>...</FRAMESET>` задає кількість вертикальних чи горизонтальних фреймів та їх розмір. Тег `<FRAMESET>` замінює елемент `<BODY>` на Web-сторінці.

Одинарний тег `<FRAME атрибути>` визначає, що буде відображатись у певному фреймі і яке ім'я матиме фрейм.

Приклад:

Файл *index.html* має вигляд:

```
<html>
<head>
<title> фрейми </title>
</head>

<frameset cols="20%,*">
<frame src="menu.html" name="MENU">
<frame src="content.html"
name="CONTENT">
</frameset>

</html>
```

Результат:



При використанні фреймів необхідно як мінімум три HTML-файла: перший визначає фреймову структуру і ділить вікно браузера на дві частини (файл `index.html`), а інші два документи завантажуються в задані вікна. Кількість фреймів не обов'язково дорівнює двом, може бути й більше, але не менше двох.

Для вищенаведеного прикладу: в тегу `<FRAME>` задається ім'я HTML-файла, що завантажується за допомогою атрибуту `SRC`. В ліве вікно завантажено файл `menu.html`, а в праве – файл `content.html`.

Файл `menu.html` має вигляд:

```
<html>
  <head>
    <title>Навігація по сайту</title>
  </head>
  <body style="background:#f0f0f0">
    <p>МЕНЮ</p>
  </body>
</html>
```

Файл `content.html` має вигляд:

```
<html>
  <head>
    <title>Зміст сайту</title>
  </head>
  <body>
    <p>ЗМІСТ</p>
  </body>
</html>
```

Атрибути тегу `<FRAMESET атрибути>...</FRAMESET>`:

- `border` – товщина границі між фреймами
`<frameset border="значення">...</frameset>`;
- `bordercolor` – колір лінії границі
`<frameset bordercolor="колір">...</frameset>`;
- `cols` – встановлює ширину або пропорції фреймів у вигляді колонок (ширина задається у вигляді пік селів, відсотків або символу *)
`<frameset cols="ширина 1, ширина 2, ...">...</frameset>`;
- `rows` – встановлює висоту або пропорції фреймів у вигляді рядків
`<frameset rows="висота 1, висота 2, ...">...</frameset>`.

Атрибути тегу `<FRAME>`:

- `bordercolor` – колір лінії границі
`<frame bordercolor="колір">`;
- `src` – шлях до файлу, призначеному до завантаження у фреймі
`<frame src="URL">`;
- `name` – задає унікальне ім'я фрейму
`<frame name=" ім'я ">`.

3.9 Оформлення HTML-документу за допомогою CSS

Cascading Style Sheets (Каскадні Таблиці Стилів – CSS) – це спеціальна мова опису стилів, яка володіє набагато багатшим і функціональним, в порівнянні з HTML, набором засобів форматування і управління стилями елементів документа.

Після введення CSS деякі теги і атрибути в HTML стали вважатися застарілими (за допомогою CSS можна застосовувати форматування набагато ефективніше).

Таблиці стилів складаються з набору правил. Кожне правило складається з одного або декількох селекторів і блоку визначення, що виділяється фігурними дужками.

Блок визначення може містити одне або декілька властивостей відокремлених крапкою з комою (;) (після останньої властивості крапка з комою необов'язкова). Кожна властивість повинна мати значення відокремлене двокрапкою (:).

Приклад коду з використанням і без використання CSS, виконання якого приводить до форматування таблиці:

```
<html>
<head>
<style type='text/css'>
#css
{
border-collapse:collapse;
font-family:'arial';
}
#css th
{
background-color:#DC63C2;
padding:7px;
font-size:1.1em;
border:1px black solid;
color:white;
}
#css td
{
padding:7px;
border:1px solid;
}
</style>
</head>
<body>
<p><b>Приклад таблиці, оформленої за допомогою CSS</b> </p>
<table id='css'>
<tr>
<th>Атрибути тегу FRAME</th>
<th>Значення</th>
</tr>
<tr>
<td>bordercolor</td>
<td>копір лінії границі</td>
</tr>
<tr style='background-color:#ded2db'>
<td>src </td>
<td>шлях до файлу</td>
</tr>
```

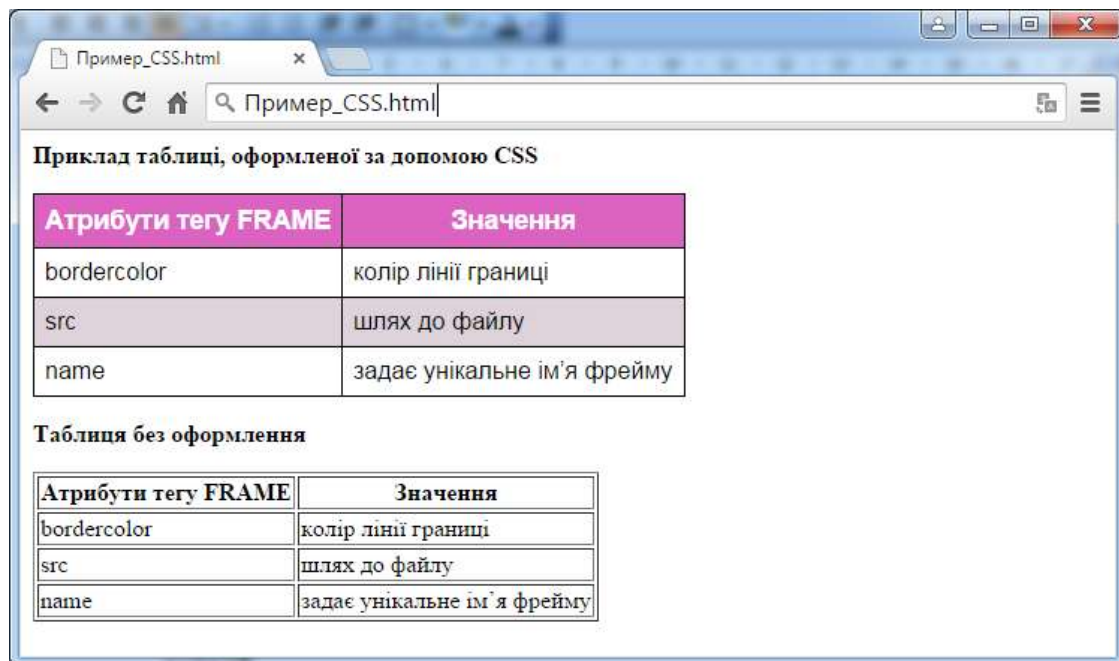
```

<tr>
<td>name </td>
<td>задає унікальне ім'я фрейму</td>
</tr>

</table>
<p><b>Таблиця без оформлення</b> </p>
<table border='1'>
<tr>
<th>Атрибути тегу FRAME</th>
<th>Значення</th>
</tr>
<tr>
<td>bordercolor</td>
<td>колір лінії границі</td>
</tr>
<tr>
<td>src </td>
<td>шлях до файлу</td>
</tr>
<tr>
<td>name </td>
<td>задає унікальне ім'я фрейму</td>
</tr>
</table>
</body>
</html>

```

Результат роботи HTML-коду:



Контрольні запитання

1. Назвіть теги, що створюють заголовки.
2. Яку загальну структуру має HTML-документ?

3. Наведіть засоби дизайну тексту на мові HTML.
4. Синтаксис HTML-атрибутів.
5. Перелічіть атрибути тегу ...
6. Чи можна зробити гіперпосиланням графічний об'єкт?
7. Як створити ефект руху для тексту?
8. Як створити фрейми?
9. Як створити на Web-сторінці маркований список?
10. Для чого призначені CSS?

4 ПРОГРАМУВАННЯ В Excel

4.1 Створення макросів

Середовище MS Office пропонує два способи створення програм, що відрізняються підходом до процесу: використання макрорекордера і ручне кодування (на мові VBA). Ці підходи орієнтовані на різні категорії: безпосередньо користувачів і програмістів відповідно. Макрорекордер (Macrorecorder) – це програмний інструмент, що записує дії користувача при роботі з документами і додатками, із збереженням запису у вигляді **макросу** – вихідного коду на мові VBA.

Розглянемо побудову макросу, що виконує заповнення діапазону A1:A10 послідовністю натуральних чисел від 1 до 10. Нижче представлено процес створення подібного макросу.

Вибрати команду «Сервіс»/«Макрос»/«Запис макроса...». В результаті розкриється діалогове вікно, що показано на Рис. 37.

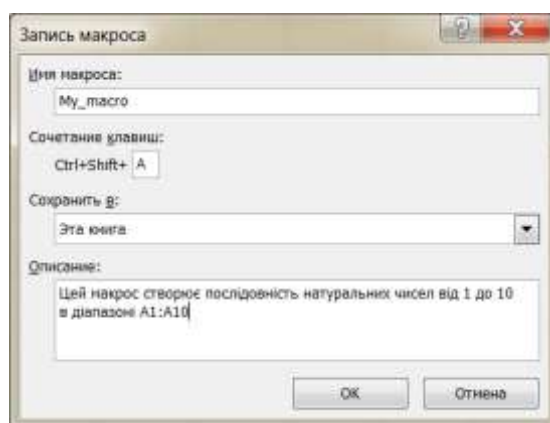


Рис. 37 Вікно запису макросу

Після натискання на кнопку ОК починається запис макросу.

Алгоритм запису макросу:

1. Ввести число від 1 до 10 в діапазон A1:A10 за допомогою маркеру заповнення (Рис 38, Рис 39)

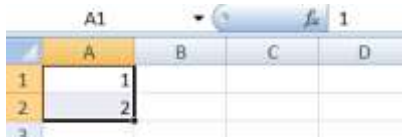


Рис. 38 Внесення чисел до комірок A1 та A2

2. Виділити комірку B1. Результат, що отримано, показано на Рис. 40.

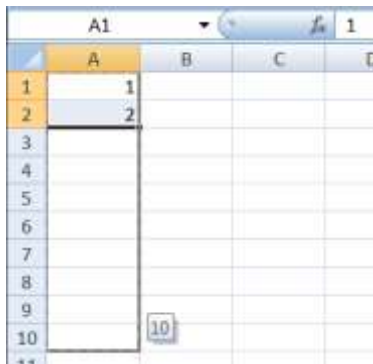


Рис. 39 Заповнення діапазону A1:A10 за допомогою маркера заповнення

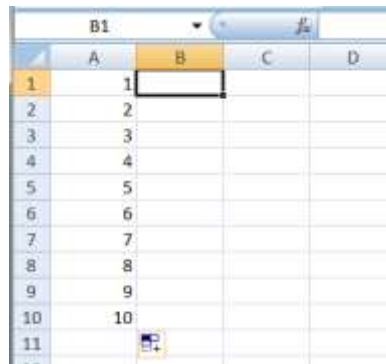


Рис. 40 Вигляд робочого аркуша перед завершенням запису макросу

3. Наприкінці слід натиснути на кнопку закінчення запису макросу (Рис 41).

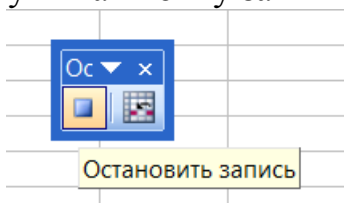


Рис. 41 Зупинка запису макросу

В подальшому, для запуску макросу необхідно вибрати команду «Сервис»/«Макрос»/«Макросы». Відкриється діалогове вікно Макрос (Рис 42), в якому потрібно вибрати необхідний для роботи макрос. Інший спосіб запустити макрос – використати комбінацію клавіш Ctrl+Shift+A.

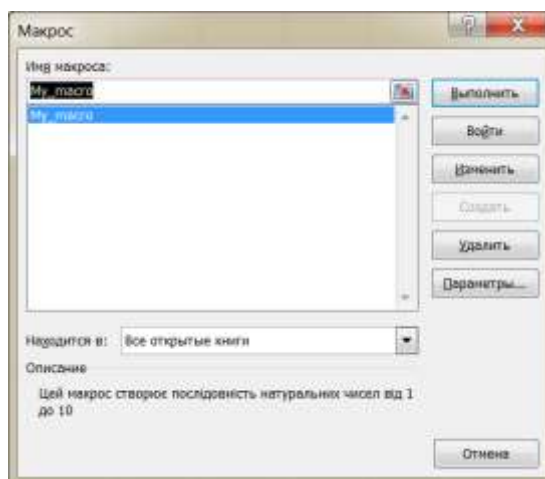


Рис. 42 Вікно Макрос, що можна використовувати для запуску макросів

У наведеному прикладі діапазон комірок, що заповнюється не залежить від того, яка комірка на момент запуску макросу є активною, тобто посилання на вказаний діапазон є абсолютним. При створенні макросу корисно використовувати не абсолютні, а відносні посилання.

Для того, щоб посилання на комірки, або на діапазони комірок інтерпретувались як відносні, на панелі **«Остановка записи»** при записі макросу необхідно натиснути кнопку **«Относительная ссылка»** (Рис 43).

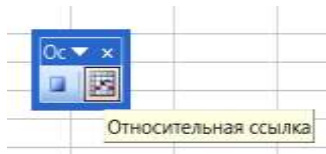


Рис. 43 Перехід до режиму використання відносних посилань

4.2 Редактор VBA

Microsoft Excel – це дуже потужний інструмент, який має багатий набір можливостей стандартного інтерфейсу користувача і який можна використовувати для управління даними, їх аналізу і візуалізації. Але інколи, може знадобитися знайти нескладний спосіб виконувати повсякденні завдання, що повторюються, або якісь завдання, які не можна вирішити за допомогою інтерфейсу користувача. В додатки Office, такі як Excel, інтегрований Visual Basic для додатків (VBA), мова програмування, що дозволяє розширювати можливості цих додатків. Найпоширенішою причиною використання VBA в Excel є автоматизація завдань, що повторюються.

VBA працює, виконуючи макроси, покрокові процедури, написані на мові Visual Basic.

Відповідні коди створюються та редагуються в спеціальному редакторі, який визивається з вікна додатку Excel. Для цього достатньо вибрати команду **«Сервіс»/«Макрос»/«Редактор Visual Basic»** (для Excel 2007-2010 використати стрічку **«Разработчик»**) (Рис. 44).

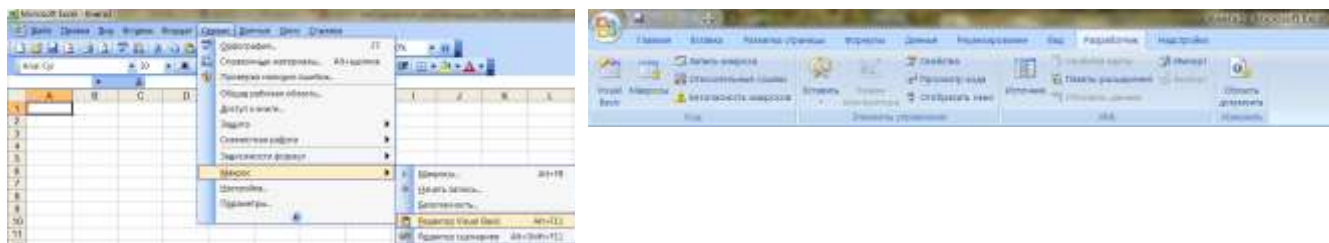


Рис. 44 Виклик редактора VBA

В результаті відкривається вікно редактора VBA (Рис. 45):

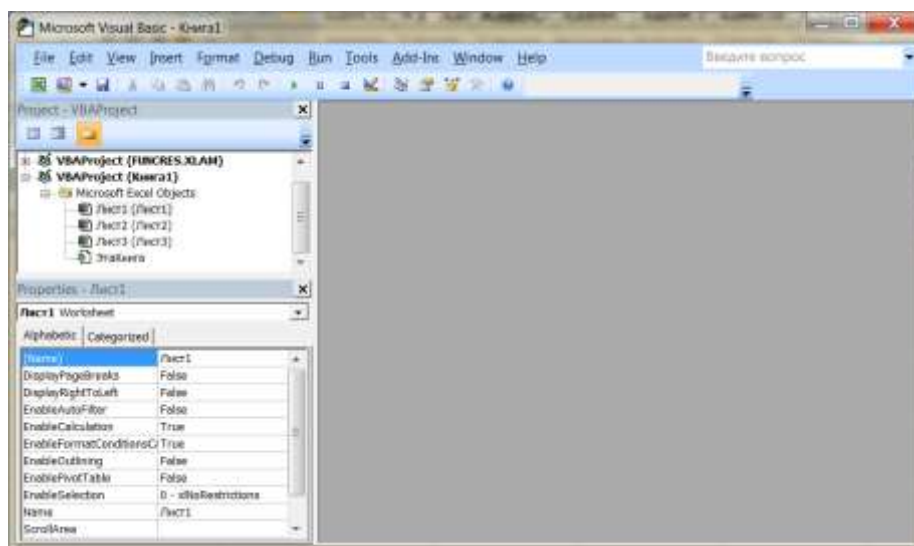


Рис. 45 Вікно редактора VBA

Модуль – місце, де зберігається програмний код.

Форма – діалогове вікно з різними елементами управління.

Проект – сукупність модулів, що створюють програму.

Для редагування програмного коду макросу (або функції) слід відкрити той модуль, у якому він був записаний. В якості прикладу наведено код макросу, який був створений в п. 4.1 (Рис. 46)

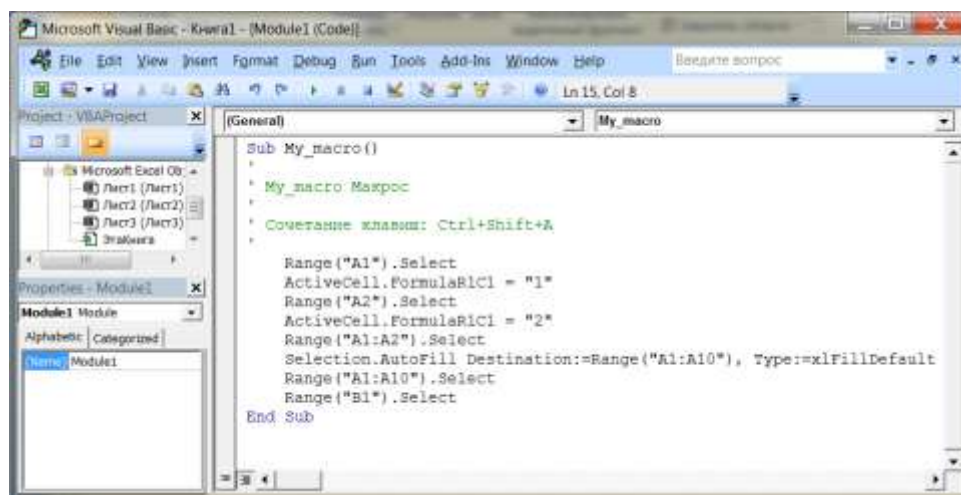


Рис. 46 Вікно з кодом макросу

В цей код, вже після створення макросу, можна вносити зміни.

4.3 Поняття об'єкту, властивості, методу, події

В VBA застосовується об'єктно-орієнтований підхід до розробки додатків. Кожний додаток – Word, Excel, Access, PowerPoint тощо - доповнює засоби VBA новими функціональними можливостями за допомогою об'єктів, властивостей і методів, притаманних цьому додатку. Як об'єктно-орієнтована мова програмування VBA використовує наступні основні поняття: об'єкт, властивість, метод, подія, клас і сімейство об'єктів.

Об'єкти VBA – це такі програмні одиниці, що містять у себе і дані і процедури їх обробки. Багато з них мають візуальне відображення на екрані: комірки таблиці, робочі листи, панелі керування і навіть Excel в цілому – все це є екземпляри базових об'єктів. Об'єкти можуть бути вкладені один в одного, до них можна звертатися з мови програмування VBA.

Приклад: необхідно виконати якісь дії з коміркою робочого листа, наприклад, занести у комірку A5 листа "Лист1" книги "MyBook.XLS" число 13, треба записати наступне:

```
Workbooks("MyBook.XLS").Sheets("Лист1").Range("A5").value = 13
```

Тут з усіх відкритих на поточний час об'єктів-книг використовується екземпляр з назвою "MyBook.XLS". З усіх його листів обирається об'єкт-лист з назвою "Лист1". В ньому розшукується об'єкт-група комірок, що у даному випадку складається з однієї комірки "A5". І тільки після цього значення об'єкту-комірки набуває числа 13.

З об'єктом нерозривно пов'язаний клас. Клас задає властивість і поведінку об'єктів. Кожен об'єкт є екземпляром якого-небудь класу. Наприклад, кожен робочий аркуш MS Excel є об'єктом класу `Worksheet` (робочий аркуш).

Властивості об'єкту VBA – це константи і змінні, що належать екземпляру об'єкта і несуть у себе дані про цей конкретний екземпляр. Наприклад, розмір та колір окремої комірки, формула, що в неї записана, значення тій формули і багато іншого є властивостями об'єкту "комірка".

Наприклад, об'єкт `Range`, тобто група комірок, має понад 80 властивостей. З них часто використовуються наступні:

- .Value – значення, що записано або обчислено;
- .Formula – формули для обчислень;
- .Font – шрифт та його атрибути для даної групи комірок;
- .Borders – параметри рамки.

Методи об'єкту VBA – це макроси (процедури і функції), що зв'язані з об'єктом і можуть виконуватися над значеннями властивостей його екземплярів.

Методи і властивості записуються після назви об'єкту через крапку. Оскільки в результаті виконання методів можуть виникати нові екземпляри об'єктів або посилання на вже існуючі екземпляри, до них теж можна застосовувати методи. Тому мають сенс і часто використовуються ланцюжки типу `Об'єкт.Метод1.Метод2.Метод3...`

Наприклад, об'єкт `Range`, має понад 70 методів. З них часто використовуються наступні:

- .Select – обирає групу комірок для поточної обробки;
- .Activate – робить комірку активною, тобто видимою на екрані, і ставить на неї маркер;
- .Calculate – обчислює формули у комірках;
- .Clear – знищує всі дані та форматування у комірках;

.ClearContents – знищує зміст комірки, не порушуючи її форматування;
 .Delete – видаляє комірки.

Подія – це характеристика об'єкта, що описує зовнішній вплив, на який реагує об'єкт при застосуванні додатка. Події – це, наприклад, завантаження форми, натискання клавіш клавіатури або миші. Кожна подія має своє ім'я.

Найбільш поширеними при використанні миші є такі події:

Click – натискання на кнопку миші;

Dbclick – подвійне натискання на об'єкті керування.

4.4 Принципи роботи з об'єктами

Об'єкт – це програмний елемент, який має своє відображення на екрані, містить деякі змінні, що визначають його властивості, і деякі методи для управління об'єктом. В MS Excel є багато вбудованих об'єктів (Таблиця 3):

Таблиця 3

Приклади вбудованих об'єктів

Range ("Адреса")	Діапазон комірок (може включати тільки одну комірку)
Cells(i, j)	Комірка, що знаходиться на перетині i-го рядка і j-го стовпця робочого аркуша MS Excel
Rows (№ рядка)	Рядок із заданим номером
Columns (№ стовця)	Стовпець із заданим номером
Sheets ("Ім'я")	Аркуш з вказаним ім'ям
Sheets (№ аркуша)	Аркуш з вказаним номером
WorkSheet	Робочий аркуш

Встановлення значень властивостей – це один із засобів керування об'єктом.

Синтаксис встановлення значень властивостей:

Объект. Свойство = Выражение

Основною властивістю об'єктів Cells і Range, є Value (значення).
 Наприклад:

1) в діапазон комірок A5:A10 вноситься значення 0

Range("A5:A10"). Value = 0 або

Range("A5:A10") = 0.

2) в комірка, що знаходиться на перетині 2-го рядка і 4-го стовпця (комірка з адресою D2) вноситься значення змінної n

Cells(2, 4). Value = n або

Cells(2, 4) = n

Синтаксис читання значень властивостей:

Переменная = Объект. Свойство

Наприклад, змінній Xn привласнюється значення з комірки B1 поточного аркуша:

```
Xn = Cells(1, 2).Value або
Xn = Range("B1").Value
```

Синтаксис вживання методів до об'єкту:

Объект. Метод

Наприклад:

```
1) зробити активним Аркуш 2
  Sheets(2).Activate
2) видалити аркуш з ім'ям «Діаграма»
  Sheets("Діаграма").Delete
3) виділити діапазон комірок A2:B10
  Range("A2:B10").Select — выделить диапазон ячеек.
```

У MS Excel є об'єкти, які містять інші об'єкти. Наприклад, робоча книга містить робочі листи, робочий аркуш містить діапазон комірок і так далі. Об'єктом самого високого рівня є Application (додаток). Якщо необхідно змінити його властивості або викликати його методи, то результат застосовується до поточної книги MS Excel. Наприклад, завершення роботи з Excel: Application.Quit.

Крапка після імені об'єкту може використовуватися для переходу від одного об'єкту до іншого. Наприклад, наступне вираження очищає другий рядок робочого аркуша Май в робочій книзі Звіт:

```
Application.Workbooks("Отчет").Worksheets("Май").Rows(2).Delete
```

4.5 Представлення даних: типи даних, константи та змінні, масиви

Константи – це явно задані або означені дані. Вони об'являються зі словом Const.

Для завдання константи достатньо записати службове слово Const, після якого її ім'я, потім знак "=" і саме значення. Текстові константи беруться у лапки. Наприклад:

```
Const Pi_5 = 3.14159
Const Экзамен = "Інформатика"
Дні_тижня = ("Понеділок", "Вівторок", "Середа", "Четвер",
"П'ятниця", "Субота", "Неділя")
```

Змінні – це дані, значення яких можна змінювати при виконанні програми. Для завдання змінних використовується службове слово Dim, після якого ставиться її ім'я, потім службове слово As і вказується тип змінної.

Наприклад:

```
Dim Площа As Double
Dim Ширина
Dim i, g23, k3_a As Integer
Dim X As Byte, Z As Integer, C, Слово As String
```

Тип Variant використовується, якщо його задано явно, або коли інше не вказано. Змінні цього типу можуть приймати значення замість будь-якого з інших простих типів.

Окремим видом змінних є **масиви**. При їх завданні після імені змінної ставиться у дужках список інтервалів зміни індексів (у простішому випадку - кількість елементів у масиві).

Наприклад:

```
Dim AA(50)
```

```
Dim BB3(1 To 5, 4 To 9, 3 To 5) As Double
```

Якщо треба створити масив з наперед заданими елементами, можна скористатися функцією Array. Але слід пам'ятати, що вона повертає не справжній масив, а змінну типу Variant, що містить у себе заданий масив у символічному виді.

Типи даних розділяються на прості (скалярні) і складні (структуровані). В простих типах даних можливі значення даних єдині і неділимі. Складні ж типи мають структуру, в яку входять різні прості типи даних. Скалярні типи даних представлені в таблиці 4.

Таблиця 4

Скалярні типи даних VBA

Ім'я типу	Можливі значення
Boolean (логічний)	True, False
Byte	0...255
Integer (ціле)	-32768...+32767
Long (довге ціле)	-2147483648...+2147483647
Single (число з плаваючою точкою)	-3,4E38...-1,4E-45 для негативних значень; 1,4E-45...3,4E38 для позитивних значень
Double (число з плаваючою точкою подвійної точності)	-1,7E308...-4,9E-324 для негативних значень; 4,9E-324...1,7E308 для позитивних значень
String (рядковий)	рядки фіксованої довжини (до 216 символів); рядки змінної довжини (до 231 символів). Дані записуються в лапках
Object (об'єкт)	Посилання на об'єкт (показчик)
Variant	Універсальний тип; значенням можуть бути дані будь-якого з перерахованих вище типів, значення NULL і значення помилок ERROR.

4.6 Керуючі конструкції VBA: оператори розгалуження, вибору та циклу

Керуючі конструкції мови програмування – це інструкції та групи інструкцій, застосування яких дозволяє змінювати послідовність виконання інших інструкцій програми. Керуючі конструкції підрозділяються на розгалуження і цикли.

4.6.1 Розгалуження

Оператор розгалуження перевіряє задану умову та стосовно до її істинності виконує перший або другий операнди.

1) конструкція If... Then

```
If умова Then
    оператор
...
    оператор
End if
```

2) конструкція If... Then...Else

```
If умова Then
    оператор
...
    оператор
Else
    оператор
...
    оператор
End if
```

3) конструкція If... Then...ElseIf...Else

```
If умова1 Then
    оператор
...
    оператор
ElseIf умова2 Then
    оператор
...
    оператор
...
Else
    оператор
...
    оператор
End if
```

4) конструкція Select Case

```
Select Case вираз, що перевіряється
    Case вираз 1
        оператор
        ...
        оператор
    Case вираз 2
        оператор
        ...
        оператор
    Case Else
        оператор
        ...
        оператор
End Select
```

4.6.2 Цикли

Оператори циклу бувають декількох видів: з керуючою умовою, з лічильником, цикли за структурою даних.

1. У циклах з керуючою умовою оператор або група операторів повторюється доти, поки таку умову не буде виконано. Такі цикли називають циклами Do. Залежно від типу керуючої умови, розрізняють чотири види циклів Do.

Цикл першого типу Do While умова оператор ... оператор Loop	Цикл другого типу Do оператор ... оператор Loop While умова
Цикл третього типу Do Until умова оператор ... оператор Loop	Цикл четвертого типу Do оператор ... Оператор Loop Until умова

2. Циклами з керуючим лічильником користуються в тих випадках, коли заздалегідь відомо кількість повторень тілу циклу і перевіряти керуючу умову немає необхідності.

```

For лічильник=початкове_значення То кінцеве_значення
    Step крок_циклу
    оператор
    ...
    оператор
Next лічильник
  
```

3. У циклах за структурою даних тіло циклу по черзі повторюється для всіх однорідних об'єктів, що становлять масив.

```

For Each елемент In структура даних
    оператор
    ...
    оператор
Next елемент
  
```

4.7 Оператори вводу-виводу

4.7.1 Оператор MsgBox

Оператор MsgBox здійснює виведення інформації в діалоговому вікні і встановлює режим чекання натиснення кнопки користувачем і має такий синтаксис:

MsgBox Повідомлення[, Кнопки] [, Заголовок]

Аргументи:

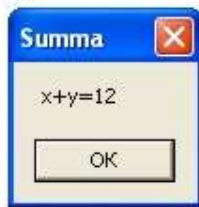
Повідомлення – обов'язковий аргумент, який задає у вікні інформаційне повідомлення, що виводиться. Може складатися з декількох текстових рядків, об'єднаних знаком &.

Кнопки – значення цього аргументу визначає категорії кнопок, що з'являються у вікні. Від значення аргументу кнопки залежить також, чи з'являється у вікні який-небудь значок. Якщо не вказано, які кнопки необхідно відображувати у вікні повідомлень, то використовується значення за умовчанням, відповідне кнопці ОК.

Заголовок – задає заголовок вікна.

Приклад процедури з використанням оператора MsgBox.

```
Sub example_01()
    Dim x As Integer, y As Integer, z As Integer
    Dim A As String
    A = "x+y="
    x = 5
    y = 7
    z = x + y
    MsgBox A & z, , "Summa"
End Sub
```



4.7.2 Функция InputBox

Функция Inputbox здійснює введення значень змінних за допомогою вікна вводу і має синтаксис:

Ім'я_змінної = InputBox(Повідомлення[, Заголовок])

Аргументи:

Повідомлення - обов'язковий аргумент. Задає у вікні інформаційне повідомлення, що зазвичай пояснює сенс величини, що вводиться.

Заголовок – задає заголовок вікна.

Приклад: Ввести значення змінної N з клавіатури, передбачивши значення за замовчуванням рівне 10.

N = InputBox("Введіть N", "Ввод исходных данных", 10)

В результаті буде виведено наступне вікно для вводу значення змінної N:



4.8 Функції та процедури

Процедурою називається фрагмент коду (мінімальна семантично закінчена програмна конструкція), розміщений між операторами `Sub` і `End Sub`.

Спрощений синтаксис процедури такий:

```
[Private|Public] Sub <ім'я_процедури> (<аргумент_1>, ... ,
<аргумент N>)
<оператор>
...
<оператор>
End Sub
```

Основною відмінністю між процедурою та функцією є те, що функція повертає значення. Синтаксис функцій трохи відрізняється від синтаксису процедур, тому що для функції необхідно вказати тип значення, що повертається нею:

```
[Private|Public] Function <ім'я_функції> (<аргумент_1>, ... ,
<аргумент N>) As <тип_значення_що_повертається>
<оператор>
...
<оператор>
<ім'я_функції> = <обчислене_значення>
End Function
```

Ключове слово `Private` обмежує область видимості процедури або функції, для якої воно задане, модулем, у якому вона описана. У результаті її можуть викликати тільки процедури або функції цього ж модуля.

Ключове слово `Public`, навпаки, оголошує процедуру або функцію доступною для всіх модулів проекту. За замовчуванням будь-яка процедура або функція загальнодоступна, тобто має статус `Public`.

Виконання програмного коду процедури або функції відбувається при передачі їй керування. Передача керування (виклик) може здійснюватися різними способами. У загальному випадку підпрограма викликається із програмного коду за допомогою спеціального оператора `Call`, у якому, крім цього ключового слова, вказуються ім'я процедури й фактичні параметри виклику, список яких вміщується в круглі дужки.

Дуже важливу роль у програмуванні на VBA грає такий різновид процедури, як процедура без параметрів. Процедури цього типу можуть виступати в ролі процедур обробки подій. Процедури обробки подій є основою механізму зв'язку подій з програмним кодом для їхньої обробки.

Функції в VBA, як правило, містять списки параметрів. Вони не можуть використовуватися для обробки подій. Викликати ж їх можна як за допомогою оператора `Call`, так і безпосередньо в процесі розрахунку значення виразу VBA (з коду інших процедур або функцій) або формули робочого аркуша MS Excel.

Звичайно програма мовою VBA складається з декількох процедур і функцій. Опис цих програмних одиниць зберігається в модулях. Модулі, у свою чергу, як складові частини входять у проекти. Проекти зберігаються у файлах робочих

книг MS Excel, але працювати з ними можна лише при наявності окремого додатка – редактора Visual Basic. Якщо потрібно викликати процедуру або функцію, опис якої перебуває в іншому модулі, її ім'я при виклику вказується в такий спосіб:

`ім'я_модуля.ім'я_процедури.`

Але якщо опис процедури або функції розміщено в іншому проекті, при виклику треба використати інший синтаксис:

`ім'я_проекту.ім'я_модуля.ім'я_процедури.`

Контрольні запитання

1. Які об'єкти має об'єктна модель Excel?
2. Як розробити функцію користувача?
3. Методи об'єкту VBA – це...
4. Яким чином можна створити модуль робочого аркуша, модуль робочої книги?
5. Як називаються процедури, пов'язані з подіями робочого аркуша?
6. Об'єкти VBA – це...
7. Властивості об'єкту VBA – це...
8. Яким чином можна записати макрос?
9. Як звернутися у процедурі до діапазону комірок, окремої комірки?
10. Назвіть керуючі структури VBA.

Лабораторна робота №1

Основні прийоми роботи з текстовими документами. Створення комплексного документу засобами MS Word

Мета: набуття практичних навичок роботи з інструментами редагування та форматування в MS Word.

Завдання:

1. Скопіювати файл **Вимірювання кількості інформації** з папки \\kit-server\estud\mnv\oeei в особисту папку. Відкрити файл.
2. Виділити весь текст, застосувати до тексту російську мову («Сервис»/«Язык»/«Выбрать язык»).
3. Виконати перевірку орфографії («Сервис»/«Правописание»). Якщо варіантів немає, то виправити помилки вручну.
4. Видалити зайві пробіли («Правка»/ «Заменить», два пробіли замінити на один, кнопка «Заменить все»). Процес заміни продовжувати до тих пір, поки не буде зроблено 0 замін.
5. Включити недруковані символи (кнопка ) , з'єднати розірвані речення і видалити зайві символи абзацу.
6. Оформлення абзаців тексту (крім заголовків): вирівнювання – **по ширині**, міжрядковий інтервал – **полуторний**, перший рядок абзацу – **1 см**. Оформлення заголовків: вирівнювання – **по центру**, міжрядковий інтервал – **полуторний**.
7. Оформлення шрифту основного тексту: розмір – **14**, гарнітура – **Times New Roman**, накреслення – **звичайне**. Оформлення шрифту заголовків: розмір – **16**, гарнітура – **Times New Roman**, накреслення – **напівжирне**.
8. Замість(11111111111111111111) вставити формулу:

$$I = -(p_1 \log_2 p_1 + p_2 \log_2 p_2 + \dots + p_N \log_2 p_N)$$

Замість (22222222222222222222) вставити формулу:

$$I = \log_2 N$$

Замість (33333333333333333333) вставити формулу:

$$-\left(\frac{1}{10^{10}} \log_2 \frac{1}{10^{10}} + \left(1 - \frac{1}{10^{10}}\right) \log_2 \left(1 - \frac{1}{10^{10}}\right)\right) \approx 3,4 \cdot 10^{-9}$$

Формули розмістити по центру.

9. Наприкінці документу, на новій сторінці («Вставка»/«Разрыв»), додати заголовок **Розклад занять**.
10. Використати можливості панелі інструментів «Таблицы и границы» та:
 - підготувати шаблон таблиці для розкладу занять;
 - заповнити таблицю розкладом занять на тиждень.
11. Зберегти файл.
12. Скопіювати три останніх абзаци до нового документу і виконати російсько-український переклад тексту. Провести перевірку правопису та граматики.

13. Отриманий переклад зберегти в особистій папці з ім'ям **Переклад**.

Запитання для самоперевірки:

1. Назвіть елементи форматування символів і абзаців у Word.
2. Що означає символ  ?
3. Які групи символів властиві для редактора математичних формул?
4. Способи створення таблиць у Word.
5. Яким чином з'єднати два абзаци в один?

Теоретичні відомості: див. 1.1.2, 1.2.3, 1.2.5, 1.3.2, 1.3.3.

Література: основна [3; 6; 7]; додаткова [1].

Лабораторна робота №2

Засоби автоматизації розробки документів

Мета: набуття практичних навичок роботи з інструментами MS Word для створення стилів, колонтитулів, виносок, списку ілюстрацій, змісту документу.

1. Відкрити файл **Вимірювання кількості інформації.doc** в особистій папці.
2. Вставити в різних місцях документа три рисунки та порожню таблицю («**Таблиця**»/«**Добавить таблицу**»: два рядки і два стовпці). Заповнити таблицю даними виконавця (прізвище, ім'я, спеціальність, номер групи).
3. Наприкінці документа, на новій сторінці, додати заголовок **Результати атестації**. Додати діаграму («**Вставка**»/«**Диаграмма**»). Діаграма повинна бути побудована за табличними даними про результати атестації чотирьох студентів групи.
4. До рисунків і діаграми додати назви («**Вставка**»/«**Ссылка**»/«**Название**»: стандартна назва – **Рисунок**).
5. Пронумерувати сторінки тексту в правому верхньому куті аркуша («**Вставка**»/«**Номера страниц**»).
6. На номер групи, що стоїть в одній з комірок таблиці Результати атестації додати посторінкову виноску з назвою факультету («**Вставка**»/«**Ссылка**»/«**Сноска**»).
7. Створити нижній колонтитул («**Вид**»/«**Колонтитулы**»), що містить тему лабораторної роботи і дату виконання лабораторної роботи («**Вставка**»/«**Дата и время**»).
8. Наприкінці документа, на новій сторінці, додати заголовок **Список рисунків**, створити список ілюстрацій («**Вставка**»/«**Ссылка**»/«**Оглавление и указатели**»/«**Список иллюстраций**»).
9. Створити новий стиль з ім'ям **№групи** і задати для цього стилю такі параметри: вирівнювання – **по ширині**, міжрядковий інтервал –

полуторний, перший рядок абзацу – 2 см, шрифт – **Arial**, розмір шрифту – **12**, накреслення – **звичайне**.

10. Застосувати стиль *№группи* до всього документу, крім заголовків і підписів рисунків.
11. Створити новий стиль з ім'ям *Заголовок№группи* і задати для цього стилю такі параметри: вирівнювання – **по центру**, міжрядковий інтервал – **полуторний**, інтервал перед та після абзацу – 18, шрифт – **Arial**, розмір шрифту – **16**, накреслення – **напівжирне**, нумерація – **Нумерований список**.
12. Застосувати стиль *Заголовок№группи* до всіх заголовків.
13. На окремій сторінці, на початку документа, додати заголовок **Зміст**, створити зміст документа (**«Вставка»/«Ссылка»/«Оглавление и указатели»**).
14. Зберегти файл.

Запитання для самоперевірки:

1. Як надати заголовку тексту стиль *Заголовок 1*, *Заголовок 2* і т.д.
2. Як створити назви для рисунків?
3. Як створити новий стиль або змінити існуючий?
4. Яким чином оновити вміст поля?
5. Яким чином здійснювати перехід між верхнім та нижнім колонтитулами?

Теоретичні відомості: див. 1.2.1, 1.2.2, 1.2.3, 1.2.4, 1.3.4, 1.3.5, 1.3.6.

Література: основна [6; 7]; додаткова [1].

Лабораторна робота №3

Використання засобу *Злиття* для автоматизації створення документів

Мета: набуття практичних навичок роботи з інструментами злиття MS Word для створення об'єднаних документів.

1. Створити документ (параметри шрифту: розмір – **14**, гарнітура – **Times New Roman**, накреслення – **звичайне**; параметри абзацу: вирівнювання – **по ширині**, міжрядковий інтервал – **полуторний**, перший рядок абзацу – 1 см), що містить такий текст:

Запрошуємо на презентацію нової продукції нашої фірми, яка відбудеться ____вказати дату____ в ____вказати час____ у великому конференц-залі за адресою: ____вказати адресу____.

З повагою, відділ маркетингу

Зберегти документ в особистій папці в якості шаблону з ім'ям ***Запрошення_им'я виконавця лабораторної роботи.***

2. Створити новий документ на основі шаблону ***Запрошення_им'я виконавця лабораторної роботи.*** Внести до пропущених місць зміни. На початку отриманого документа додати два порожні рядки. Зберегти документ з ім'ям ***Лист.***
3. Виконати команду «Сервис»/«Письма и рассылки»/«Слияние». Обрати тип документа «Письма».
4. Створити джерело даних («Создание списка»/«кнопка Создать») із 10 записів з полями: *Звернення, Ім'я, Прізвище, Організація, Адреса, Поштовий індекс, Країна*. В поле *Ім'я* ввести ім'я та по-батькові адресатів.
5. Зберегти дані з ім'ям ***Джерело*** в особистій папці.
6. На початку документа ***Лист*** додати звернення («Создание письма»/«Другие элементы»), використовуючи поля злиття «Звернення», «Ім'я». Між полями передбачити пропуск, а після імені – знак оклику.
7. Зберегти файл в особистій папці з ім'ям ***Лист_им'я виконавця лабораторної роботи*** і закрити його.
8. Створити конверт («Сервис»/«Письма и рассылки»/«Слияние»). Обрати тип документа «Конверты».
9. В отриманому документі, що є конвертом, додати *Блок адреси* одержувача і адресу відправника.
10. Вибрати режим структури документа («Вид»/«Структура») і наприкінці документу додати вкладений документ з ім'ям ***Лист_им'я виконавця лабораторної роботи*** (кнопка .
11. Об'єднати документи за допомогою кнопки «Слияние в новый документ» (кнопка .
12. Вибрати режим розмітки сторінки.
13. Зберегти отриманий файл в особистій папці з ім'ям ***Конверт_Запрошення.***

Запитання для самоперевірки:

1. Який документ називається основним, а який джерелом даних?
2. Як створити джерело даних?
3. Як додати в основний документ поля злиття?
4. Як активізувати панель «Слияние»?
5. Як додати блок адреси для злиття документів при створенні конвертів?

Теоретичні відомості: див. 1.1.1, 1.1.2, 1.3.3,

Література: основна [6; 7]; додаткова [1].

Лабораторна робота №4

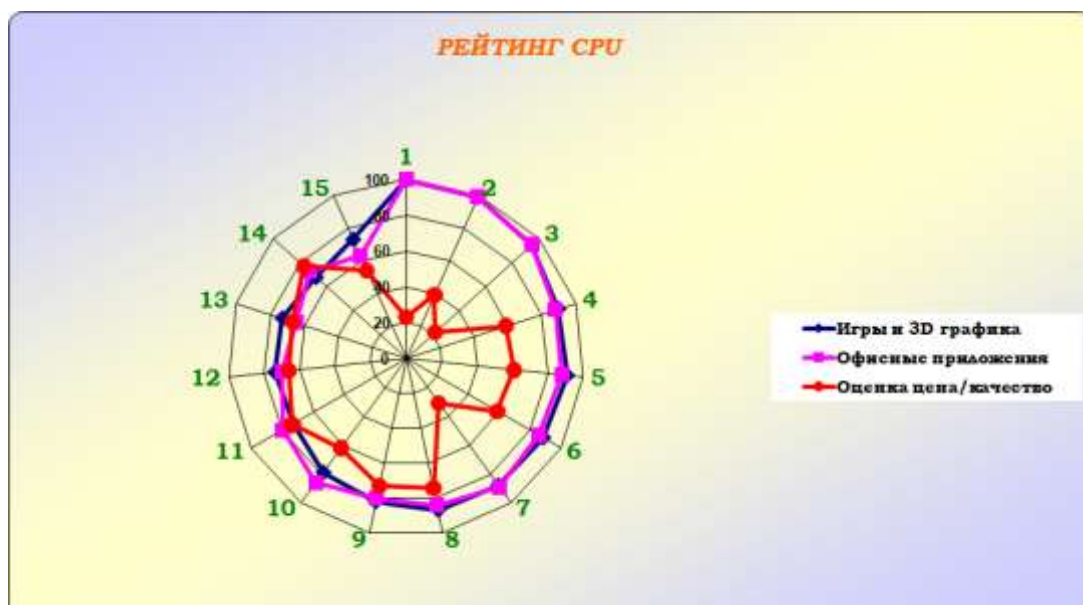
Створення та форматування таблиць і діаграм MS Excel

Мета: набуття практичних навичок роботи з інструментами MS Excel для створення, редагування та форматування таблиць і діаграм.

1. Починаючи з комірки **A1**, на **Аркуші 1** ввести дані таблиці. Для підрахунку загальної оцінки скористатися відповідною формулою.
2. Оформлення таблиці – у відповідності із заданими форматами.
3. Побудувати діаграму на основі даних таблиці у відповідності із заданими форматами.
4. Підрахувати число моделей із загальною оцінкою більше 80. Комірці результату надати відповідне ім'я.
5. Знайти загальну ціну моделей Intel Core i5. Комірці результату надати відповідне ім'я.
6. У таблиці закріпити область, яка утворює шапку таблиці.

Приклад таблиці і відповідної діаграми:

Рейтинг CPU						
№	Модель	Цена с НДС	Игры и 3D графика	Офисные приложения	Оценка цена/качество	Общая оценка (среднее)
1	Intel Core i7-3960X	8 537,0 грн.	100	100	23	74
2	Intel Core i7-3930X	4 930,0 грн.	99	99	39	79
3	Intel Core i7-990X Extreme Edition	8 327,0 грн.	95	95	22	71
4	Intel Core i7-2700K	2 769,0 грн.	90	88	59	79
5	Intel Core i7-3820	2 644,0 грн.	91	88	61	80
6	Intel Core i7-2600K	2 550,0 грн.	89	86	59	78
7	Intel Core i7-970	6 612,0 грн.	88	89	31	69
8	Intel Xeon E3-1230	2 011,0 грн.	87	84	74	82
9	Intel Core i5-2500K	1 867,0 грн.	82	80	73	78
10	AMD FX-8150	2 062,0 грн.	79	86	62	76
11	AMD FX-8120	1 587,0 грн.	73	80	74	76
12	Intel Core i5-2400S	1 693,0 грн.	74	70	66	70
13	AMD Phenom II X6 1055T	1 137,0 грн.	73	65	67	68
14	AMD FX-6100	1 227,0 грн.	68	73	77	73
15	AMD Phenom II X4 980 Black Edition	1 386,0 грн.	73	63	54	63



Запитання для самоперевірки:

1. Як змінити числовий формат для певних комірок?
2. Як присвоїти комірці або діапазонаві комірок ім'я?
3. Який синтаксис має функція СУММЕСЛИ?
4. Яким чином змінити тип побудованої діаграми?
5. Назвіть параметри форматування рядів даних діаграми.

Теоретичні відомості: див.

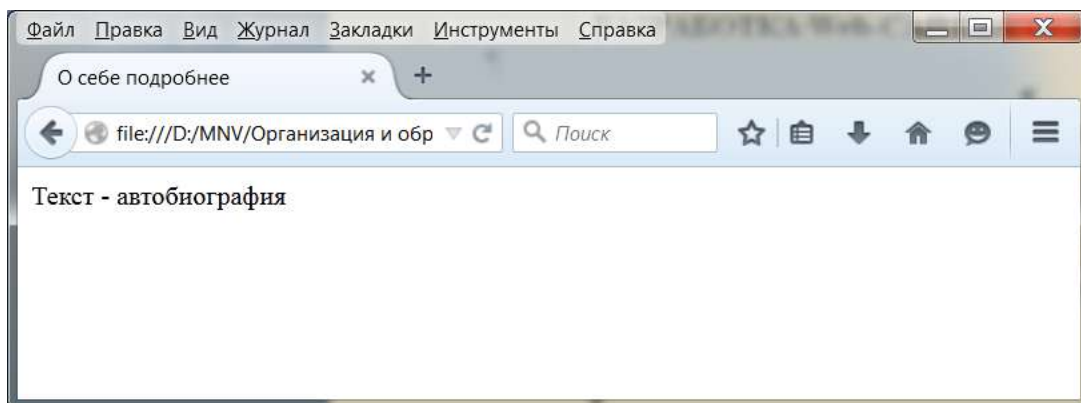
Література: основна [3; 4; 7]; додаткова [4].

Лабораторна робота №5

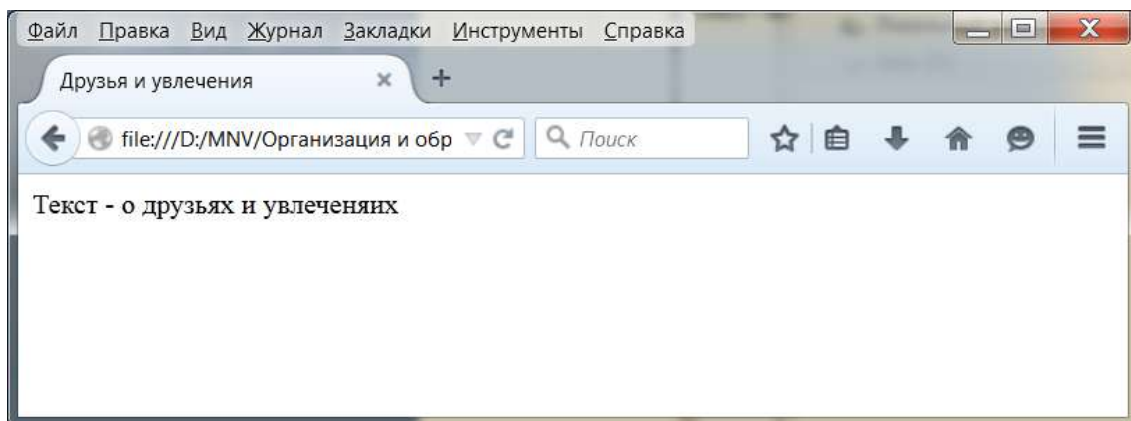
Розробка Web-сайту засобами мови HTML

Мета: набуття практичних навичок для створення та дизайну Web-сайту засобами мови HTML.

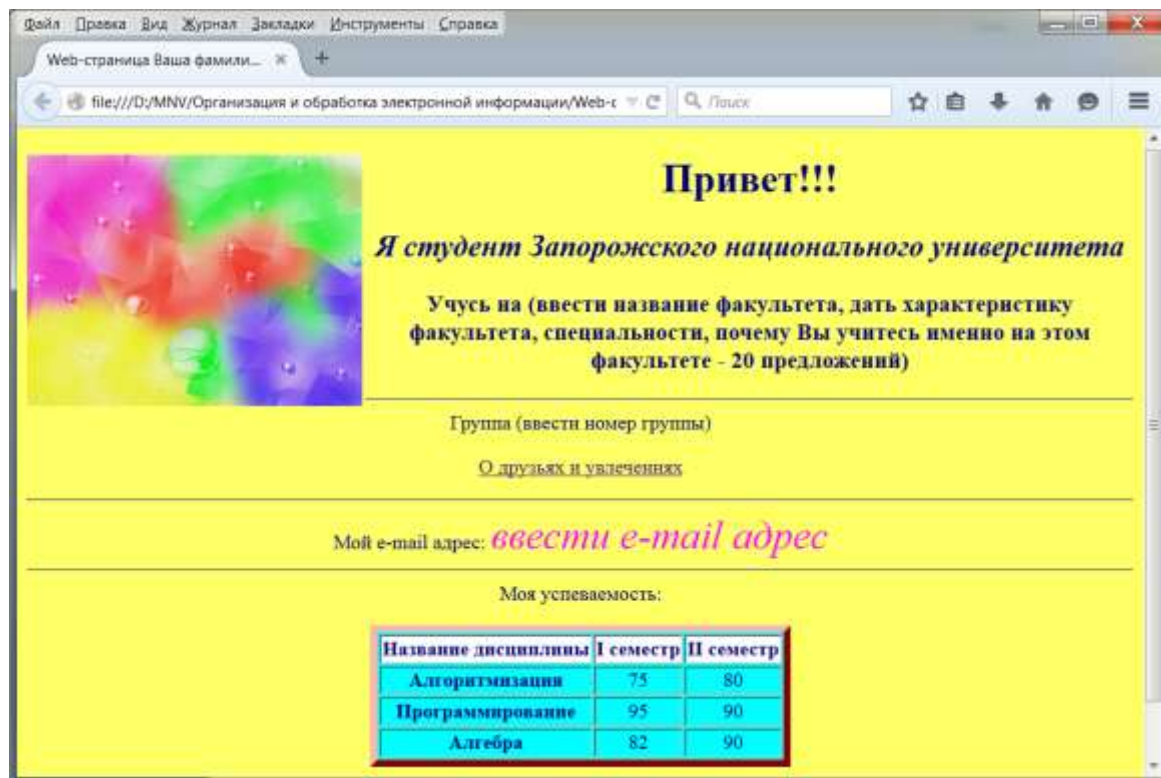
1. Створити файл з іменем file2.htm, результатом роботи якого є:



2. Створити файл з іменем file3.htm, результатом роботи якого є:



3. Створити файл з іменем file1.htm (додати на Web-сторінку таблицю, що містить бали успішності за трьома дисциплінами, додати графічний об'єкт). Результат роботи (наприклад):



Запитання для самоперевірки:

1. Яким чином можна змінити колір границь таблиці?
2. Яким чином додати на Web-сторінці горизонтальну лінію?
3. Як задати жовтий колір фону Web-сторінки?
4. Як задати назву, розмір і колір шрифту тексту Web-сторінки?
5. За допомогою якого тегу можна сформувати рядок таблиці?

Теоретичні відомості: див. 3.1, 3.2, 3.3, 3.5, 3.6, 3.7.

Література: основна [2; 5]; додаткова [2; 3].

Лабораторна робота №6

Розробка Web-сайту: створення фреймової структури, гіперпосилань

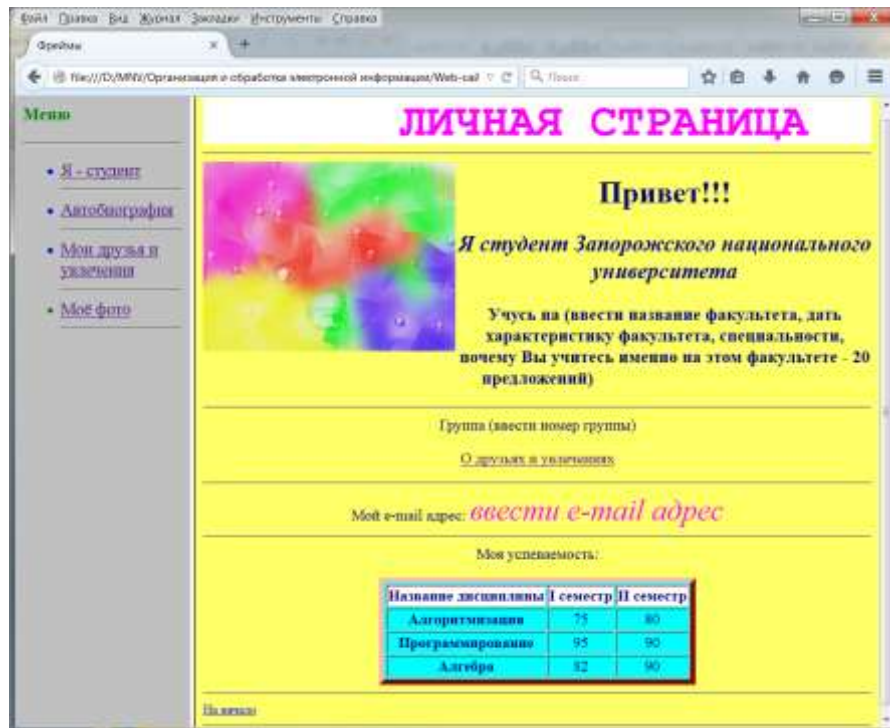
Мета: набуття практичних навичок для створення фреймової структури та дизайну Web-сайту засобами мови HTML.

1. В тексті коду Web-сторінки в файлі file1.htm додати гіперпосилання на файл file3.htm (після тексту про групу).
2. В тексті коду Web-сторінки в файлі file1.htm додати графічне гіперпосилання на файл file2.htm (після тегу <BODY>).
3. В тексті коду Web-сторінки в файлі file1.htm додати гіперпосилання *На початок* (малим шрифтом).
4. В тексті коду Web-сторінки в файлі file1.htm додати рухомий рядок: *Особиста сторінка_Прізвище_ім'я* (форматування – довільне).

5. Створити фреймову структуру вікна браузера в файлі з ім'ям Index.htm. Створити два вертикальних фрейми: 1 – 20% від горизонтального розміру вікна браузера з ім'ям "left" (відображається файл з ім'ям Menu.htm (см. п.6)); 2 – останні 80% від горизонтального розміру вікна браузера з ім'ям "right" (відображається файл file1.htm).

6. Спроектувати файл Menu.htm, який містить гіперпосилання на файли file1.htm, file2.htm, file3.htm і графічний об'єкт. При натисканні на гіперпосилання необхідна Web-сторінка повинна відображатися в правому фреймі.

Приблизний вигляд Web-сторінки:



Запитання для самоперевірки:

1. За допомогою якого тегу встановлюється розмір фрейму?
2. Для чого використовується тег `<a>...</a>`?
3. Записати синтаксис створення гіперпосилань.
4. За допомогою яких атрибутів встановити ширину та висоту фрейму?
5. Назвіть основні атрибути тегу `<frame>`.

Теоретичні відомості: див. 3.1, 3.2, 3.3, 3.7, 3.8, 3.9.

Література: основна [2; 5]; додаткова [2; 3].

Лабораторна робота №7

Мова програмування VBA: функції користувача у MS Excel

Мета: набуття практичних навичок роботи із засобами мови програмування VBA для розробки макросів, функцій користувача.

1. В особистій папці створити документ MS Excel, дати ім'я *Лабораторна_№7*.
2. Скопіювати таблицю з лабораторної роботи №4 в файл з ім'ям *Лабораторна_№7*, аркуш назвати *Таблиця*.
3. Створити макрос з ім'ям *Макрос_F*, доступний у всій робочій книзі, в процесі активізації якого відображаються всі формули активного робочого аркуша. Для активізації використовувати комбінацію клавіш Ctrl+f.
4. Створити макрос з ім'ям *Макрос_NF*, доступний у всій робочій книзі, в процесі активізації якого відключається режим відображення формул робочого аркуша. Для активізації використовувати комбінацію клавіш Ctrl+n.
5. Розробити процедуру для автоматичного підрахунку кількості переходів до робочого аркуша *Таблиця*.

Запитання для самоперевірки:

1. Як створити макрос і призначити для його активації командну кнопку?
2. Як вилучити макрос?
3. Об'єкт Application – це...
4. Яким об'єктом може задаватися діапазон комірок?
5. Які функції властиві для робочого аркуша?

Теоретичні відомості: див. 4.1, 4.2, 4.3, 4.4.

Література: основна [1; 8]; додаткова [3; 4].

Лабораторна робота №8

Мова програмування VBA: створення процедури

Мета: набуття навичок роботи з функціями та процедурами VBA.

Розробити програму для автоматичного розрахунку максимального і мінімального елементів на аркуші MS Excel. Зберегти процедуру в модулі робочого аркуша *МинМакс*.

1. Під час кожного внесення змін до певних комірок робочого аркуша, обчислюється максимальне і мінімальне значення в цих комірках, після чого здійснюється форматування комірок, які містять мінімальне і максимальне значення:
 - формат вмісту комірки з найбільшим значенням: накреслення – напівжирне, колір – червоний, розмір – 20;

- формат вмісту комірки з найменшим значенням: накреслення – напівжирне, колір – зелений, розмір – 20;
 - формат останніх комірок: накреслення – звичайне, колір – чорний, розмір – 14.
2. У вікні повідомлень виводиться сумарне значення вмісту комірок.

Приблизний результат аналізу і обробки даних

	A	B	C	D	E	F	G	H
1	Инвестиционная деятельность							
2		Германия	Россия	Чехия	Литва			
3	Январь	15	5	6	4			
4	Февраль	23	9	15	2,9			
5	Март	31	7	0,5	2,34			
6	Апрель	11	6	5	6,23			
7	Май	45	1	14	8,56			
8	Июнь	4	3,25	7	2			
9								



Запитання для самоперевірки:

1. Поняття процедури VBA.
2. Який синтаксис вживання методів до об'єкту?
3. Який синтаксис встановлення значень властивостей об'єкта?
4. Яке призначення властивості Value?
5. Для чого призначений оператор MsgBox?

Теоретичні відомості: див. 4.3, 4.4, 4.5, 4.6, 4.8.

Література: основна [1; 8]; додаткова [3; 4].

ТЕРМІНОЛОГІЧНИЙ СЛОВНИК

Абзац	– сукупність символів документа, яка завершується символом кінця абзацу ¶
Атрибут	– властивість тегу, що дає додаткову можливість форматування тексту
Властивість об'єкту VBA	– параметри об'єкта, які йому притаманні
Гіперпосилання	– фрагмент HTML-документа, що вказує на інший файл, який може бути розташований в Інтернет, і містить повний шлях (URL) до цього файлу
Макрос	– спеціальні програми, які призначено для виконання всередині інших програм, наприклад Word; використовуються для автоматизації часто вживаних або складних послідовностей дій користувача
Метод	– програма, яка впливає на об'єкти і їх параметри. За допомогою методів можна, наприклад, закрити об'єкт, вилучити його, або показати на екрані
Модуль VBA	– місце, де зберігається програмний код
Об'єкт	– те, чим можна управляти за допомогою програми на мові VBA, наприклад, форма, кнопка, робочий аркуш або діапазон комірок MS Excel
Проект VBA	– сукупність модулів, що створюють програму
Розділ	– структурна одиниця текстового документа, яка об'єднує певну його частину за змістом і зазвичай має свій заголовок
Список	– сукупність абзаців, на початку кожного з яких стоїть номер або маркер
Стиль	– іменований набір параметрів форматування, який застосовується до певних структурних одиниць документа
Тег	команда мови HTML
Форма VBA	– діалогове вікно з різними елементами управління
Форматування документа	– надання документу такого вигляду, який відповідає певним вимогам користувача
Фрейм	– частина вікна браузера, в якій можна переглядати окремо від інших частин певні Web-сторінки
Шаблон Word	– файл, що містить стилі, структуру, параметри сторінок і ін., на основі якого можна створювати нові документи
Елемент HTML	– основна структурна одиниця Web-сторінки, написана на мові HTML
CSS	– каскадні таблиці стилів – формальна мова опису зовнішнього вигляду документа, написаного з використанням мови розмітки
HTML	– мова гіпертекстової розмітки
VBA	– мова програмування (Visual Basic for Application)

ЛІТЕРАТУРА

Основна:

1. Гарбер Г.З. Основы программирования на Visual Basic и VBA в Excel 2007. / Гарбер Г.З. – М.: СОЛОН-ПРЕСС, 2008. –192 с.
2. Джамса К. Эффективный самоучитель по креативному Web-дизайну. HTML, XHTML, CSS, JavaScript, PHP, ASP, ActiveX. Текст, графика, звук и анимация. Пер с англ./ Крис Джамса, Конрад Кинг, Энди Андерсон – М. : ООО «ДиаСофтЮП», 2005. – 672 с.
3. Информатика базовый курс, учебник для вузов / под ред. С. В. Симоновича – СПб : Питер, 2008. – 640 с.
4. Кузьмин В. Microsoft Office Excel 2003. Учебный курс / В. Кузьмин – СПб. : Питер; Киев : BHV, 2004. – 493 с.
5. Мальчук Е.В. HTML и CSS. Самоучитель / Е.В.Мальчук – М. : Издательский дом “Вильямс”, 2008. – 416 с.
6. Новиков Ф., Сотскова М. Microsoft Office Word 2007 / Ф. Новиков, М. Сотскова – СПб. : BHV, 2007. – 960 с.
7. Тихомиров А.Н. Microsoft office 2007. Все программы пакета: Word, Excel, Access, Powerpoint, Publisher, Outlook, Onenote, Infopath, Groove. Самоучитель / А.Н. Тихомиров и др. – СПб. : Наука и техника, 2008. – 608 с.
8. Уокенбах Д. Excel 2010: профессиональное программирование на VBA. Пер. с англ. / Д.Уокенбах – М. : Издательский дом “Вильямс”, 2012. – 944 с.

Додаткова:

1. Волков В. Понятный самоучитель работы в Microsoft Word. / В. Волков – СПб. : Питер, 2003. – 224 с.
2. Комолова Н., Яковлева Е. HTML. Самоучитель./ Н. Комолова, Е. Яковлева – СПб : Питер, 2011. – 285 с.
3. Квинт И. HTML, XHTML и CSS на 100%. / И. Квинт– СПб. : Питер, 2010. – 384с.
4. Маликова Л.В. Практический курс по электронным таблицам MS Excel / Л.В. Маликова и др. – Учебное пособие для вузов. -2-е изд. испр. и доп. – М. : Горячая линия - Телеком, 2006. – 256 с.

ЗМІСТ	
ВСТУП.....	3
1 ІНСТРУМЕНТИ ТА ЗАСОБИ РОБОТИ З КОМПЛЕКСНИМИ ТЕКСТОВИМИ ДОКУМЕНТАМИ. ТЕКСТОВИЙ ПРОЦЕСОР.....	4
1.1 Основні можливості текстового процесора	4
1.1.1 Базові задачі текстового процесора	4
1.1.2 Форматування тексту.....	6
1.2 Засоби автоматизації розробки текстового документу	7
1.2.1 Стили	8
1.2.2 Поля.....	8
1.2.3 Таблиці.....	10
1.2.4 Діаграми.....	10
1.2.5 Формули.....	11
1.3 Форматування сторінок документа	11
1.3.1 Параметри сторінки	11
1.3.2 Автоматична перевірка правопису.....	12
1.3.3 Розділи	13
1.3.4 Виноски.....	13
1.3.5 Колонтитули	14
1.3.6 Нумерація сторінок.....	14
Контрольні запитання.....	15
2 ОБРОБКА ТАБЛИЧНО-ОРГАНІЗОВАНОЇ ІНФОРМАЦІЇ	15
2.1 Основні поняття електронних таблиць	15
2.1.1 Робоча книга, робочий аркуш.....	15
2.1.2 Форматування вмісту комірок	16
2.2 Обчислення в Excel.....	17
2.2.1 Формули, функції.....	17
2.2.2 Посилання на комірки	20
2.3 Графічне представлення даних в Excel.....	21
2.3.1 Побудова діаграм	21
2.3.2 Редагування діаграм	22
Контрольні запитання.....	22
3 РОЗРОБКА WEB-САЙТУ ЗАСОБАМИ МОВИ HTML	23
3.1 Елементи, атрибути HTML-документа.....	23
3.2 Структура HTML-кода	24
3.3 Дизайн тексту для Web-сторінок.....	24
3.4 Створення списків та їх дизайн	25
3.5 Створення таблиць та їх дизайн	27
3.6 Додавання графічних та відео файлів	27
3.7. Додавання гіперпосилань.....	28
3.8 Використання фреймів	29
3.9 Оформлення HTML-документу за допомогою CSS	30
Контрольні запитання.....	32
4 ПРОГРАМУВАННЯ В Excel	33
4.1 Створення макросів	33
4.2 Редактор VBA.....	35
4.3 Поняття об'єкту, властивості, методу, події	36
4.4 Принципи роботи з об'єктами	38
4.5 Представлення даних: типи даних, константи та змінні, масиви	39
4.6 Керуючі конструкції VBA: оператори розгалуження, вибору та циклу	40
4.6.1 Розгалуження.....	41
4.6.2 Цикли	42
4.7 Оператори вводу-виводу	42

4.7.1 Оператор MsgBox	42
4.7.2 Функция InputBox	43
4.8 Функції та процедури	44
Контрольні запитання.....	45
Лабораторна робота №1 Основні прийоми роботи з текстовими документами. Створення комплексного документу засобами MS Word	46
Лабораторна робота №2 Засоби автоматизації розробки документів	47
Лабораторна робота №3 Використання засобу <i>Злиття</i> для автоматизації створення документів	48
Лабораторна робота №4 Створення та форматування таблиць і діаграм MS Excel	50
Лабораторна робота №5 Розробка Web-сайту засобами мови HTML	51
Лабораторна робота №6 Розробка Web-сайту: створення фреймової структури, гіперпосилань	52
Лабораторна робота №7 Мова програмування VBA: функції користувача у MS Excel.....	54
Лабораторна робота №8 Мова програмування VBA: створення процедури	54
ТЕРМІНОЛОГІЧНИЙ СЛОВНИК	56
ЛІТЕРАТУРА	57

Навчальне видання
(українською мовою)

Надія Вікторівна Матвіїшина

ОРГАНІЗАЦІЯ ТА ОБРОБКА ЕЛЕКТРОННОЇ ІНФОРМАЦІЇ

Методичні рекомендації до лабораторних занять
для студентів освітнього ступеня «бакалавр»
напряму підготовки «Інформатика»

Рецензент *Д.П.Бувайло*
Відповідальний за випуск *С. Ю. Борю*
Коректор *Н.В.Матвіїшина*

Підп. до друку 00.00.2015. Формат 60×90/16. Папір офсетний.
Друк цифровий. Гарнітура Таймс. Умовн. друк. арк. 7,5.
Замовлення № 000. Тираж 20 прим.

Запорізький національний університет

69600, м. Запоріжжя, МСП-41
вул. Жуковського, 66

Свідоцтво про внесення суб'єкта видавничої справи
до Державного реєстру видавців, виготівників
і розповсюджувачів видавничої продукції
ДК № 2952 від 30.08.2007