

## Зміст

Передмова .....	5
ЛАБОРАТОРНА РОБОТА № 1 АВТОМАТИЗАЦІЯ ОБЧИСЛЕНЬ В MS EXCEL ЗА ДОПОМОГОЮ МАКРОСІВ.....	7
1.1. Основні теоретичні відомості.....	7
1.2. Опис лабораторних засобів та обладнання.....	24
1.3. Послідовність виконання роботи.....	24
1.4. Оброблення та аналізування результатів. Оформлення звіту.....	26
Контрольні запитання.....	26
ЛАБОРАТОРНА РОБОТА № 2 РОЗРОБЛЕННЯ ТА ПРОГРАМУВАННЯ ЛІНІЙНИХ АЛГОРИТМІВ .....	28
2.1. Основні теоретичні відомості .....	28
2.2. Опис лабораторних засобів та обладнання.....	55
2.3. Послідовність виконання роботи.....	55
2.4. Оброблення та аналізування результатів. Оформлення звіту.....	57
Контрольні запитання.....	57
ЛАБОРАТОРНА РОБОТА № 3 РОЗРОБЛЕННЯ ТА ПРОГРАМУВАННЯ АЛГОРИТМІВ РОЗГАЛУЖЕНОЇ СТРУКТУРИ.....	59
3.1. Основні теоретичні відомості.....	59
3.2. Опис лабораторних засобів та обладнання.....	72
3.3. Послідовність виконання роботи.....	72
3.4. Оброблення та аналізування результатів. Оформлення звіту .....	73
Контрольні запитання.....	73
ЛАБОРАТОРНА РОБОТА № 4 ОРГАНІЗАЦІЯ АРИФМЕТИЧНИХ ЦИКЛІВ .....	74
4.1. Основні теоретичні відомості.....	74

4.2.Опис лабораторних засобів та обладнання .....	86
4.3. Послідовність виконання роботи .....	87
4.4.Оброблення та аналізування результатів. Оформлення звіту .....	88
Контрольні запитання.....	88
Список рекомендованої літератури.....	88
Додаток 1.Приклад тексту макросу.....	90
Додаток 2. Команди меню редактора Visual Basic .....	91
Додаток 3.Панелі інструментів редактора Visual Basic .....	101
Додаток 4.Області видимості Private й Public .....	105
Додаток 5.Заходи безпеки під час виконання лабораторних робіт.....	106
Додаток 6.Схема єдиного ділення .....	109
Додаток 7.Типи даних VBA .....	116
Додаток 8.Аргумент buttons функції MsgBox .....	119
Додаток 9.Програми рішення квадратного рівняння .....	121
Додаток 10.Завдання до лабораторної роботи № 2 (частина 2) .....	122
Додаток 11.Поняття про алгоритм і етапи розробки програм.....	130
Додаток 12.Блок-схема алгоритму визначення Y .....	134
Додаток 13.Завдання до лабораторної роботи №3.....	135
Додаток 14.Завдання до лабораторної роботи №4 Частина 1.....	146
Додаток 15.Завдання до лабораторної роботи №4 Частина 2.....	155
Додаток 16. Приклади зчитування інформації з листів Excel в VBA і запису інформації з програми VBA в лист Excel .....	156

## Передмова

У технічній документації фірма Microsoft називає Visual Basic не інакше як системою програмування. Ця система програмування призначена для написання програм, що працюють під керуванням операційної системи Windows.

Visual Basic є продуктом багаторічної еволюції мови програмування Basic. Після появи системи Windows фірма Microsoft розробила графічний варіант мови Basic. В основному, ця мова призначалася для створення програм, що працюють у режимі діалогу з користувачем, тобто у візуальному режимі. Тому мова й названа Visual Basic. По мірі розвитку комп'ютерних технологій Visual Basic поступово перетворювався в потужний інструмент для розроблення прикладних програмних додатків.

Слово Basic у назві вказує лише на те, що синтаксис програм і оператори опираються на мову високого рівня Basic (Beginner's All Purpose Symbolic Instruction Code – символічна мова програмування загального призначення для початківців). Але при цьому Visual Basic дуже помітно відрізняється від звичайної мови Basic.

Технологія роботи в середовищі Visual Basic базується на ідеях візуального й об'єктно-орієнтованого програмування.

Візуальна мова програмування означає, що взаємодія із системою програмування при складанні й відладці програм реалізуються діалоговими засобами графічного інтерфейсу користувача.

Під об'єктно-орієнтованим програмуванням розуміють об'єднання даних і процедур їхньої обробки в об'єкти. Це спрощує створення й перетворення складних структур даних і дій над ними.

Інакше кажучи, середовище Visual Basic є автоматизованою графічною оболонкою над об'єктно-орієнтованою версією мови Basic.

Середовище програмування Visual Basic for Application(VBA) створене фірмою Microsoft на базі середовища програмування Visual Basic(VB).

VBA це об'єктно-орієнтована мова, призначена для програмування в рамках конкретної прикладної програми, і в цьому його головна відмінність від універсальних мов програмування (наприклад, від VB).

Найчастіше VBA використовують для програмування в прикладних програмах Microsoft Office. Він також вбудований у такі прикладні програми, як Visio, AutoCad, CorelDraw, WordPerfect і т.д.

Кожна з прикладних програм Microsoft Office містить велику кількість об'єктів, що відрізняються один від одного. Кожний з об'єктів має свої властивості, методи й події. Хоча основні властивості VBA залишаються незмінними для різних прикладних програм Microsoft Office (вони об'єднані базовими операторами мови VB), але при цьому кожна прикладна програма вносить свої специфічні команди й об'єкти. Це й відрізняє VBA від універсального середовища програмування VB. Крім того, програми, розроблені в середовищі VB, після компіляції можуть експлуатуватися незалежно від середовища розробки, а програми, розроблені в VBA, є компонентом певного документа однієї з прикладних програм, і вони можуть виконуватися тільки в середовищі своєї прикладної програми. Таким чином, програма, розроблена на VBA, входить до складу деякого документу Microsoft Office і слугує для автоматизації оброблення інформації в рамках цього документа.

Зазвичай в прикладній програмі (Microsoft Word, Excel, PowerPoint, Access) є вбудований редактор VB. У деяких програмах (OutLook, Internet Explorer) використовується скорочена версія VBA – VBA Script.

Вивчення мови VBA буде направлено стосовно прикладної програми Microsoft Office – Microsoft Excel.

# ЛАБОРАТОРНА РОБОТА № 1

## АВТОМАТИЗАЦІЯ ОБЧИСЛЕНЬ В MS EXCEL ЗА ДОПОМОГОЮ МАКРОСІВ

**Мета та основні завдання:** дослідити роботу і можливості спеціальної програми MacroRecorder для створення макросів в MS Excel. Ознайомитися з редактором Visual Basic і вивчити його основні команди.

**Вивчити та стисло описати в протоколі лабораторної роботи:**

- поняття макросу, запис і створення макросу, виконання записаного макросу;
- інтерфейс редактора Visual Basic.

### 1.1. Основні теоретичні відомості

Кожна з прикладних програм Microsoft Office допускає два різних рівні використання:

§ застосування прикладної програми як готового до використання універсального інструмента для рішення будь-яких завдань;

§ застосування прикладної програми як інструментального середовища для розроблення власних прикладних програм, що вирішують специфічні завдання більш ефективним способом.

Інструментальне середовище дозволяє написати підпрограми (МАКРОСИ), що дозволяють прискорити рутинну роботу, настроїти прикладну програму таким чином, щоб всі інструменти були під рукою, і тим самим підвищити ефективність роботи за рахунок того, що багато операцій, які доводилося робити вручну, будуть виконуватися автоматично.

**МАКРОС** – це підпрограма, що складається зі списку команд, які повинні бути виконані прикладною програмою. Макрос слугує для об'єднання декількох різних дій в одну процедуру, яку можна легко викликати.

При цьому слід пам'ятати, що записаний макрос не є універсальним, оскільки він може тільки повторити один раз виконану послідовність команд, і тільки у тому ж самому порядку, у якому вони були записані раніше. За допомогою ж мови VBA можна створити макрос, що буде перевіряти деякі задані умови й залежно від них змінювати послідовність або склад виконуваних команд. Допоможе мова VBA і при необхідності циклічного повторення послідовності команд. Також Visual Basic дозволяє додавати до вікон поля введення (text boxes), меню, командні кнопки (command buttons), перемикачі (option buttons), прапорці (check boxes), списки (list boxes), лінійки прокрутки (scroll bars), а також діалогові вікна для вибору файлу або каталогу. Програміст може використати сітку (grid) для обробки табличних даних, організувати взаємодію з іншими прикладними програмами Windows і мати доступ до баз даних.

### **Створення (запис) макросів**

Макрос VBA – це послідовність дій і команд користувача, збережена під унікальним ім'ям, яку може виконати дана прикладна програма (Microsoft Excel).

Як правило, запис макросу складається з наступних кроків:

1. Створення початкових умов для запису макросу. Мається на увазі приведення програмного середовища точно в такий стан, при якому передбачається виконувати записуваний макрос. Коли початкові умови створені, можна починати запис макросу.

*Наприклад, створимо макрос, що буде задавати шрифт, кегль (розмір) і кольори тексту у виділених комірках аркуша Excel. У цьому випадку, початковими умовами для такого макросу буде відкритий аркуш Excel з виділеними однією або декількома комірками (тому що, якщо почнемо запис, а потім відкриємо аркуш і станемо виділяти комірки, то й всі ці дії будуть записані в макросі; отриманий макрос буде дуже специфічний – він завжди буде відкривати ту саму книгу, той самий аркуш і формувати в ньому ті самі комірки).*

## 2. Запустити запис макросу й задати йому ім'я.

Запис макросу здійснюється за допомогою спеціальної програми MacroRecorder.

Для початку запису слід виконати в меню «Сервіс» команду «Макрос» а «Начать запись». Excel розкриє діалогове вікно «Запись макроса» (рис.1.1).

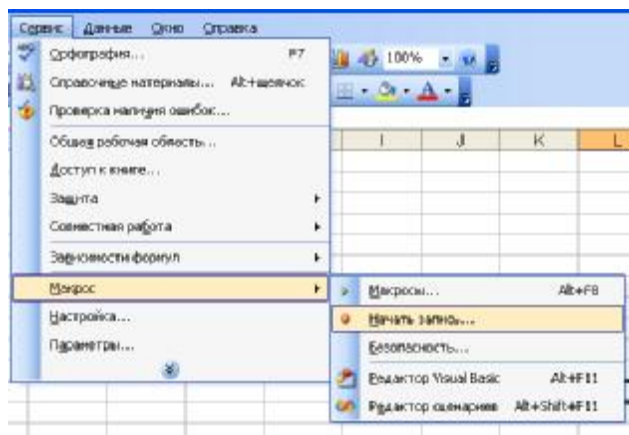


Рис.1.1 – Меню «Сервис»

Або, можна відразу скористатися кнопкою «Записать макрос», якщо увімкнена стандартна панель «Visual Basic» (рис.1.2).



Рис.1.2 – Панель Visual Basic

У діалоговому вікні «Запись макроса», що з'явилося, слід задати ім'я макросу й вказати де буде збережений макрос (за замовчуванням – у даній книзі), а також, – при необхідності, гарячі клавіші для його виконання (рис.1.3).

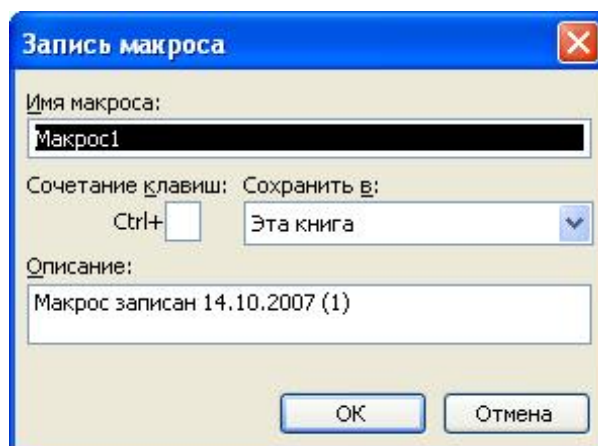


Рис.1.3 – Вікно «Запись макроса»

У текстовому полі **Имя макроса** Excel пропонує ім'я, що складається зі слова Макрос (Macro) з наступною цифрою, що відповідає порядковому номеру створюваного в цьому сеансі роботи макросу. Краще замінити це ім'я на інше, яке відповідає призначенню макросу (рис.1.3).

При іменуванні макросів варто дотримуватися наступних правил:

1. Довжина імен не повинна перевищувати **256 символів**;
2. В імені можна використовувати будь-яку комбінацію літер, цифр і символів, але воно повинне починатися з **літери**;
3. Імена не повинні включати **точок, пробілів** і наступних символів: %, &, !, #, @, \$ ;
4. Імена повинні бути унікальні усередині тієї області, де вони визначені.

У текстовому полі **Сочетание клавиш** можна задати комбінацію, за допомогою якої макрос буде запускатися на виконання. Не можна використовувати вже існуючі комбінації, оскільки вони при цьому будуть перепризначені (відбудеться скасування існуючих комбінацій клавіш). (Тому на початковому етапі цього робити не рекомендується).

Список **Сохранить в** призначений для вказування місця збереження макросу. Можливі такі варіанти збереження:

1. Эта книга.
2. Новая книга.
3. Личная книга макросов.

Личная книга макросов (PERSONAL.xls), зберігається в папці C:\Windows\...\Excel\XLSTART. Книга PERSONAL.xls (звичайно прихована) і, якщо вона існує, то автоматично відкривається при завантаженні Excel. Макроси, збережені в ній, завжди доступні, а макроси, які зберігаються в окремих книгах, доступні тільки при відкритті відповідних книг. Не рекомендується зберігати макроси в особистій книзі макросів (це роблять тільки тоді, коли макрос використовується регулярно). Варіант збереження Новая книга змусить Excel створити нову книгу й зберегти в ній макрос. Збереження макросу звичайно здійснюють у поточній книзі (Эта книга).



Текстове поле **Описание** призначене для користувачів (макроси цю інформацію не використовують). Тут корисно вказати призначення макросу. Excel поміщає сюди поточну дату й ім'я користувача (воно зазначено на вкладці **Пользователь** вікна **Параметры** в меню **Сервис**).

Записуючи макрос за допомогою MacroRecorder, рекомендується уникати будь-яких зайвих дій.

Після виконання всіх дій передбачених у даному макросі, запис макросу зупиняють. Для цього треба:

1. Вибрати команду **Макрос → Остановить запись** в меню «**Сервис**».
2. Натиснувши кнопку «**Остановить запись**» на панелі, що з'явилася



після початку запису, або на панелі «Visual Basic» (ця кнопка з'являється на місці кнопки «**Записать макрос**»).

Якщо панель «Остановка записи», не з'явилася, то її можна ввімкнути, викликавши вікно «**Настройка**» команди «**Панели инструментов**» в меню «**Вид**».

Щойно створений макрос тепер міститься в документі, що був активним при запуску запису. При збереженні документа на диск макрос зберігається разом із проектом.

Кожний макрос являє собою підпрограму (процедуру), що починається зі службового слова **SUB** і закінчується службовим словом **END SUB** (див. Додаток 1).

## Виконання макросу

Після того як макрос записаний, ним можна користуватися. Запуск макросу змушує Excel виконувати всі ті інструкції, які були записані в тілі макросу (рис.1.4).

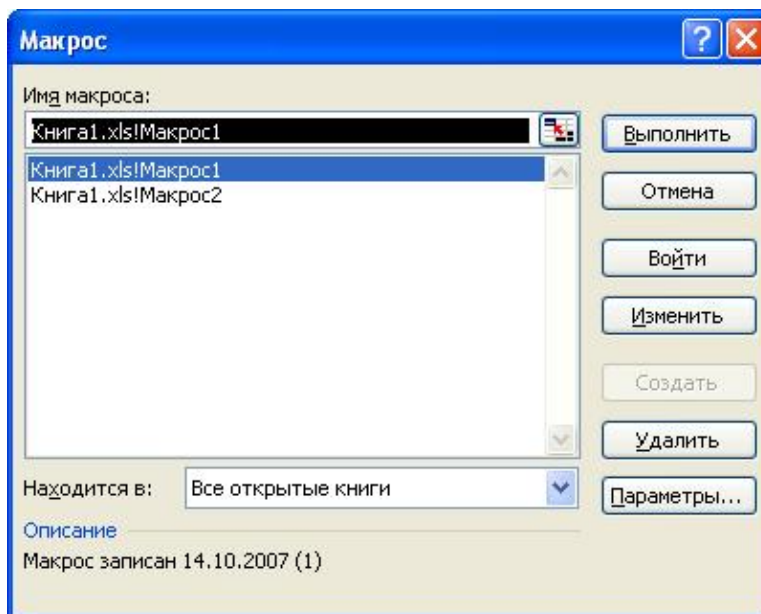


Рис.1.4 – Вікно «Макрос»

Для запуску макросу треба:

1. Вибрати об'єкт, до якого буде застосований макрос. Викликати вікно «**Макрос**» одним з наступних способів:

- виконати послідовність команд: «**Макрос**» → «**Макросы...**» в меню «**Сервис**» ;
- натиснути кнопку «**Выполнить макрос**» на панелі «**Visual Basic**»;
- натиснути комбінацію клавіш **ALT + F8**.

2. Вибрати зі списку необхідний макрос і натиснути кнопку «**Выполнить**».

*Якщо макросу призначена комбінація клавіш, то її натискання заміняє виконання пунктів 2 і 3.*

У вікні «**Макросы**» (рис.1.4) відображаються всі макроси, що містяться у відкритих в цей момент книгах. Імена макросів, що не відносяться до поточної книги, передують ім'ям книги (рис.1.4):

Книга1.xls

Макрос1



ім'я книги

ім'я макросу

Або - PERSONAL.xls!<ім'я макросу>


Якщо в списку немає потрібного макросу, то перед відкриттям списку **Ім'я макроса** слід відкрити книгу, де він збережений.

За допомогою інших кнопок вікна «**Макроси**» можна робити наступне:

1. Викликавши вікно <**Параметри**>, можна призначити гарячі клавіші, якщо це не зроблено при записі.
2. Команда <**Удалить**> – дозволяє видалити макрос, на якому в цей момент встановлений курсор.
3. Команда <**Войти**> – призначена для виконання макросу по кроках у середовищі редактора VB. Це дуже корисна процедура при відладці програми й пошуку помилок.
4. Команда <**Изменить**> – призначена для редагування програмного коду в редакторі VB.

### Редактор Visual Basic

Макроси програми Excel зберігаються у файлах робочих книг, а саме в особливій частині книги, яка називається **модулем**. Робоча книга Excel може містити кілька модулів. Модулі, що зберігаються в одній книзі, називаються проектом. Для того щоб переглянути текст модулів VBA або відредагувати записані макроси слід запустити редактор VB. Виклик редактора виконується:

- послідовністю команд: «**Макрос**» **à** **Редактор Visual Basic**» в меню «**Сервіс**»;
- кнопкою на панелі редактора <**Visual Basic**> ;
- комбінацією клавіш <**Alt+F11**>;
- послідовністю команд: «**Макрос**» **à** «**Макроси...**» **à** кнопка <**Изменить**> ( з вказівкою імені макросу у вікні «**Макроси**») в меню

«Сервис» (цей спосіб виклику можна застосовувати тільки тоді, коли хоч один макрос створений у цій книзі).

Розробка проекту для прикладних програм виконується практично повністю в редакторі VBA (спеціальному інтегрованому середовищу розробки) мовою Visual Basic. Вікно редактора і його інтерфейс VBA наведені на рис. 1.5.

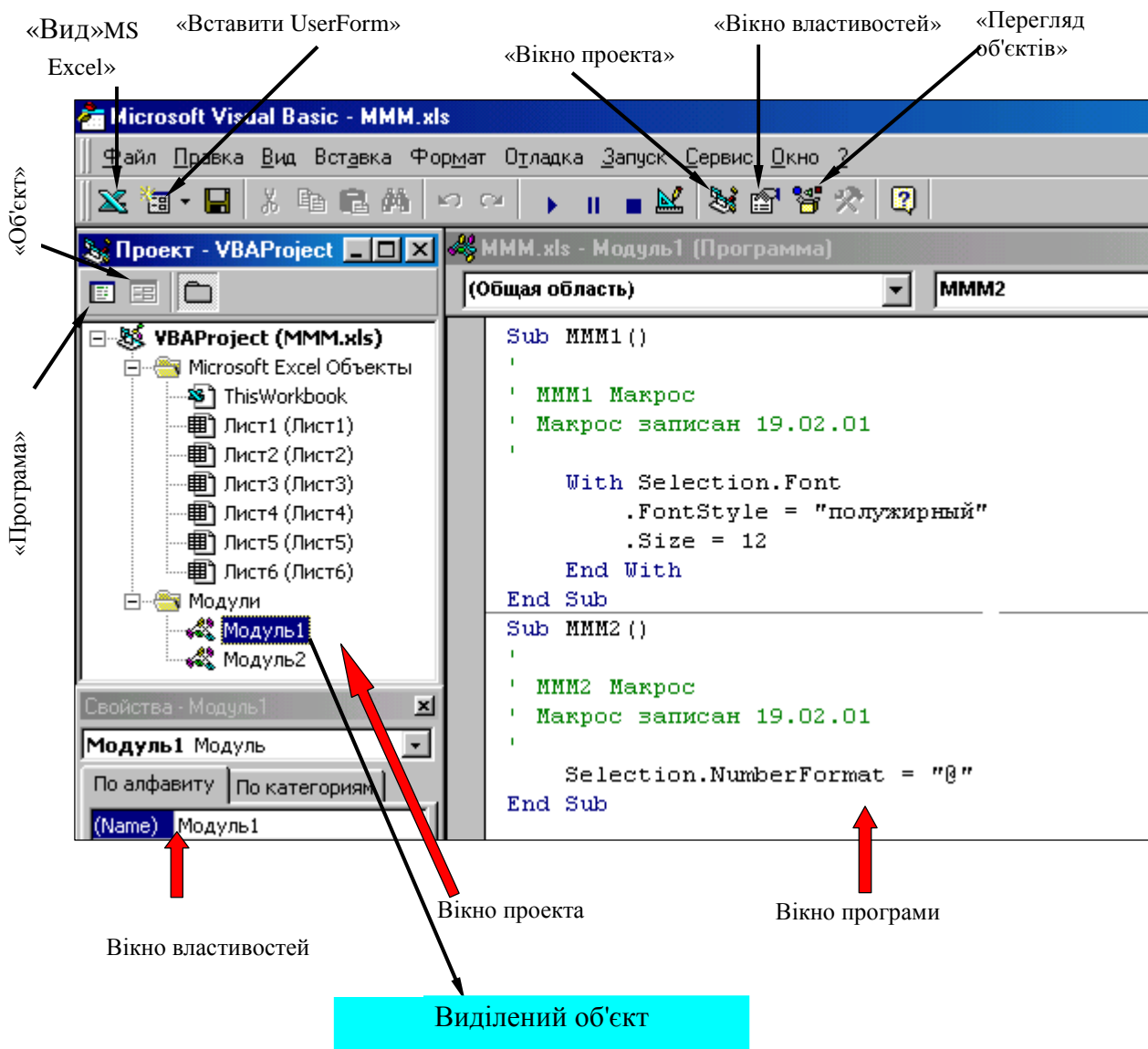


Рис 1.5 – Вікно редактора Visual Basic

Головні елементи вікна, такі як рядок заголовка, меню, панель інструментів традиційні для середовища Windows.

Меню **Файл** містить команди, що відносяться до роботи з файлами (відкриття, збереження, друк і т.д.).

В меню **Правка** зібрані команди, призначені для редагування тексту макросу у вікні програми.

В меню **Вид** зібрані команди, що дозволяють вибирати об'єкт (або елемент), а також режим перегляду.

Команди меню **Вставка** призначені для вставки різних об'єктів у проект, таких як модулі або форми.

Команди меню **Формат** використовуються при створенні власних діалогових вікон і інших форм.

Команди меню **Отладка** використовуються для перевірки макросів.

Команди меню **Запуск** слугують для запуску макросу на виконання, для переривання й поновлення роботи макросу, для відновлення початкових умов перед повторним виконанням.

Команди меню **Сервис** дозволяють вибрати макрос для виконання, одержати доступ до зовнішніх бібліотек макросів і форм. Крім цього, можна викликати діалогове вікно **Параметры** для настройки параметрів проекту.

В меню **Надстройки** можна додавати або видаляти програми-надбудови редактора VB.

Меню **Окно** і меню **Помощь** повністю аналогічні відповідним вікнам інших прикладних програм Windows.

Команди меню редактора Visual Basic наведені в Додатку 2.

Зазвичай редактор VB відображає тільки одну панель інструментів – **стандартную**. Але, крім стандартної, доступно ще три панелі (правки – **Edit**, налагодження – **Debug** і форм користувача – **UserForm**). Для виведення панелі інструментів використовується команда **Панели инструментов** в меню **Вид**.

Опис кнопок панелей інструментів **стандартної та налагодження** редактора Visual Basic наведені в Додатку 3.

При цьому вікно редактора VB містить три дочірніх вікна.

**Окно проекта** дозволяє здійснювати пошук модулів і інших елементів VBA-проекту й керувати ними. У вікні відображається ієрархічне представлення відкритих проектів у вигляді деревоподібної структури. Представлення елементів у вигляді дерева спрощує їхній перегляд і вибір. Крім того, використовуючи структуру, можна з легкістю скоротити обсяг відображуваної інформації, відключивши вивід гілки дерева. Кнопки вікна мають наступне призначення:

- кнопка **<Программа>** відображає текст програмного модуля в **<Окне программы>** (або **<Окне программного кода>**) і відкриває це вікно, розташоване в лівій частині вікна редактора VB, якщо воно було раніше закрито;
- кнопка **<Объект>** відображає на екрані вказаний об'єкт (документ, форму й т.п. );
- кнопка **<Папки>** служить для перемикання режимів відображення у вікні проекту.

**Окно свойства** призначено для відображення властивостей об'єктів. У вікні знаходяться дві вкладки: **По алфавіту** (об'єкти розташовують за абеткою) і **По категориям** (де об'єкти відсортовані за категоріями).

**Окно программы** (або **Окно(а) программного кода**) містить текст програми (програмний код). Тут кожний макрос виводиться в окреме вікно; вікна на екрані розділені пунктирною лінією. Вікно програмного кода можна розділити на дві частини для перегляду різних частин програми.

Текст програми, коментарі й службові слова виділяються різними кольорами. VBA розфарбовує частини макросу такими кольорами: коментарі виділяються зеленим кольором, ключові слова (Sub, End Sub і інші) – синім, а інші інструкції – чорним.

Редагування тексту макросу засновано на техніці й прийомах аналогічних тим, які використовують в Windows, Word, Excel. З використанням клавіатури й миші текст вводиться, виділяється, копіюється, вирізається й вставляються фрагменти, поміщені в буфер обміну.

Visual Basic перевіряє синтаксис кожного рядка. Якщо рядок набраний правильно, VBA розфарбовує її символи відповідно до правил, що описані вище. Якщо ж у рядку міститься помилка, VBA фарбує рядок у червоний колір й виводить повідомлення про помилку.

### **Переміщення й копіювання макросу з одного модуля в інший**

Перед редагуванням макросу рекомендується створення резервної копії (особливо в тому випадку, якщо плануються великі зміни або макрос великий і досить складний). Копія оберігає від невдалого редагування. Для цього треба:

- відкрити макрос, який треба копіювати – виділити макрос (включаючи рядки зі словами Sub і End Sub);
- скопіювати макрос у буфер обміну, а потім його вставити в потрібне місце (у той же самий модуль, в інший модуль, але в тому ж проекті, в інший модуль, але в іншому проекті або в документ іншої програми, наприклад, у **Блокнот**).

При цьому не рекомендується робити резервну копію в тому ж самому модулі, де зберігається вихідний макрос (проблема однакових імен).

Резервну копію всього модуля можна зберегти у вигляді текстового файлу. Це корисно при імпорті модуля в інший проект, при створенні архіву своїх модулів, для перетворення модуля у формат Visual Basic. Збереження модуля у вигляді текстового файлу називається **експортуванням модуля**. Після того як модуль експортований, його можна **імпортувати** в будь-який VBA- або Visual Basic-проект.

### **Експорт модуля**

Для експортування модуля в текстовий файл треба:

- відмітити модуль для експортування у вікні **Проект**;
- вибрати команду **Експорт файла** (Export File) в меню **Файл**;
- у діалоговому вікні **Експорт файла** (Export File), відкритому редактором VBA (вікно працює як будь-яке вікно **Сохранение документа**), вказати диск і папку, де варто зберегти текстовий файл,

ввести ім'я файлу (або погодитися з ім'ям, запропонованим редактором) і типом файлу (\*.bas), а потім клацнути по кнопці **Сохранить**.

### **Імпорт модуля**

Можна приєднати експортований файл типу \*.bas до будь-якого VBA проекту. Це робиться шляхом імпортування файлу.

Для імпортування файлу

- відмітити модуль, у який треба імпортувати файл типу \*.bas у вікні

#### **Проект;**

- вибрати команду **Импорт файла** (Import File) в меню **Файл**;
- у діалоговому вікні **Импорт файла** (Import File), відкритому редактором VBA (вікно працює як будь-яке вікно **Открытие документа**), вказати диск і папку, де варто шукати файл, раніше експортований у формат \*.bas), а потім двічі клацнути по імені файлу.

Перед відкриттям слід переконатися, що в списку типів обраний рядок VB Files. Редактор Visual Basic прочитає файл і створить у проекті новий модуль. Якщо в проекті є модуль із таким ім'ям, як в імпортованого, то Visual Basic додасть одиницю до імені нового модуля.

### **Видалення модулів із проекту**

Для видалення модуля треба:

- відмітити модуль, який варто видалити у вікні **Проект**;
- вибрати команду **Удалить** (Remove) в меню **Файл**.

Перед видаленням Visual Basic запропонує експортувати модуль. Експортування можна виконати натиснувши кнопку **Да**, а видалення – натиснувши кнопку **Нет**.

### **Створення макросу й процедури (Sub VBA)**

Створюються макроси в меню **Сервис** → **Макрос** → **Макросы** (або **Alt-F8**) (рис.1.6):

З'явиться вікно "**Макросы**".



У полі імені слід ввести ім'я (наприклад, **MY**"), а місце знаходження вказати в полі "**Находится в:**" (наприклад виберати – "**Эта книга**"). Кнопка "**Создать**" стане активною.

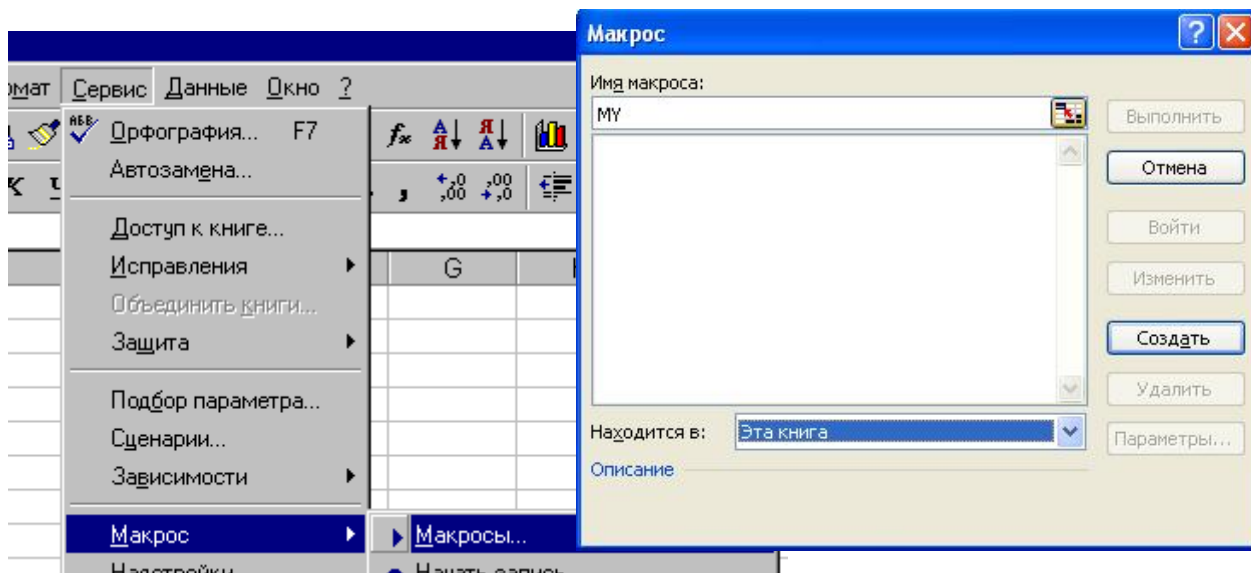


Рис 1.6 – Меню Сервис à Макрос à Макросы

1. Після натиснення на кнопку "**Создать**" з'явиться вікно програмного коду редактора Visual Basic куди між словами *Sub MY()* та *End Sub* слід ввести програмний код (текст макросу) (рис.1.7).

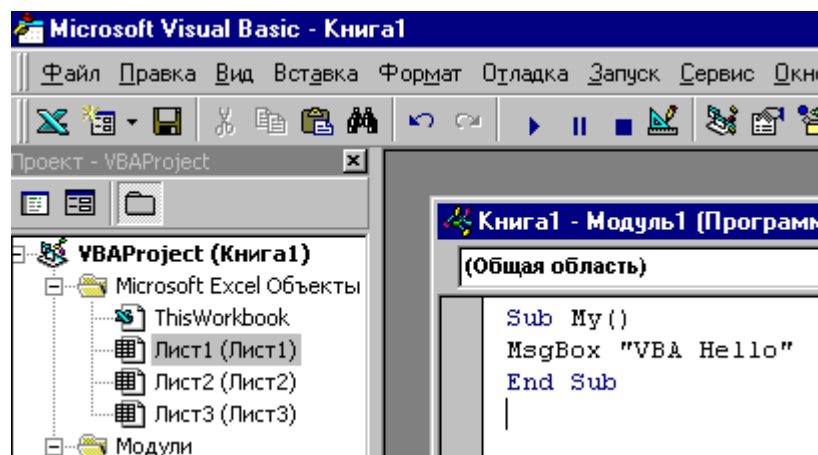


Рис 1.7 – Вікно програмного коду редактора Visual Basic

У макросі використовується функція **MsgBox**, що виводить на екран вікно повідомлення. Тепер після запуску макросу **MY** за допомогою кнопки на панелі "Visual Basic" (або пункт меню **Запуск** редактора **VB**) на екран буде виведене наступне вікно повідомлення:



Закриємо редактор **VBA**, використовуючи команду **Закрити** в меню **Файл**.

*Слід зауважити, що термін макрос повинен застосовуватися тільки для програм, записаних за допомогою засобів запису (MacroRecorder). А програму, що пишеться вручну, варто називати процедурою.*

Якщо книга, у якій необхідно створити процедуру, не містить жодного модуля, то перед написанням процедури, необхідно створити такий модуль у своєму проекті. Крім того, створення нового модуля може знадобитися, коли існуючий модуль переповнений (довжина модуля обмежена кількістю у 4000 рядків) або необхідно помістити процедуру в окремий модуль.

Для того щоб **помістити процедуру в новий модуль** необхідно:

1. Відкрити в Excel документ або книгу, де слід створити процедуру.
2. Відкрити редактор Visual Basic (наприклад, Alt – F8).
3. У вікні Проект відзначити проект, до якого додається модуль.
4. Вибрати в меню Вставка **Модуль** (або клацнути по відповідній кнопці панелі інструментів). Visual Basic створить новий модуль і відкриє вікно програмного коду (рис.1.8).

Новий модуль буде мати ім'я Модуль 1(2,3 і т.д.).

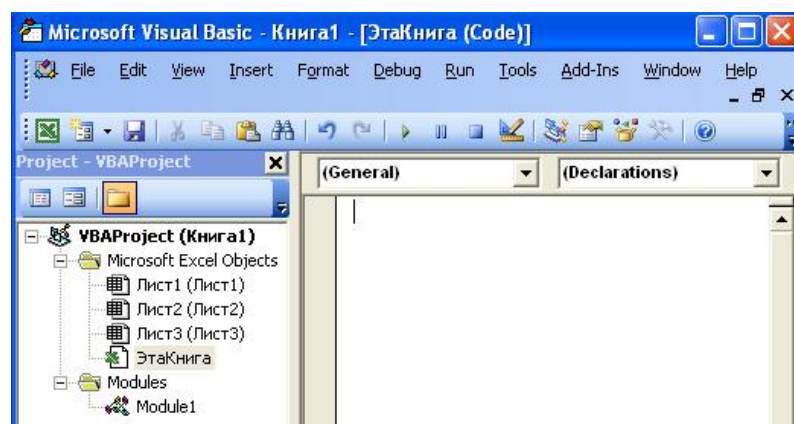


Рис 1.8 – Вікно програмного коду редактора Visual Basic

Краще щойно створеному модулю дати змістовне ім'я. Для **перейменування модуля** слід зробити наступне:

1. У редакторі Visual Basic відзначити модуль, обраний для перейменування.
2. Якщо вікно **Свойства** не відкрите, те його слід відкрити командою **Свойства**  $\Rightarrow$  в меню **Вид** (або клацнути по відповідній кнопці панелі інструментів). На рис 1.9 у вікні **Свойства** присутній один модуль, що має одну властивість – його ім'я.

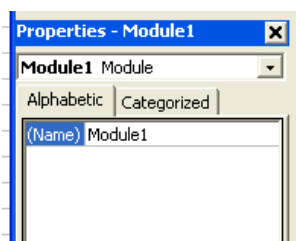


Рис 1.9 – Вікно Свойства

3. У текстовому полі слід ввести нове ім'я для модуля. Як тільки курсор покине це поле, модуль буде перейменований.

Для того **щоб створити процедуру в існуючому модулі**, слід відкрити вікно програмного коду для цього модуля. Для цього необхідно двічі клацнути мишею по імені модуля у вікні **Проект** або, відмітивши модуль, виконати команду Code (програма) в меню **Вид** (або функціональна клавіша **F7**).

Якщо в модулі присутні інші процедури, то нову варто починати або до наявної або після неї. (Більшість програмістів поміщають нову процедуру в кінець модуля). Починати запис процедури треба із поміщення в текст слова її заголовка *Sub ім'я ( )* й закінчення *End Sub*.

Наприклад (рис.1.10).

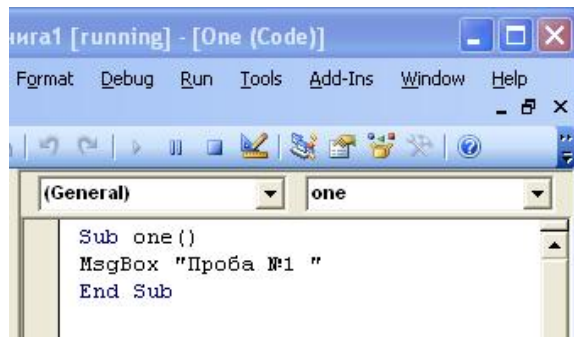

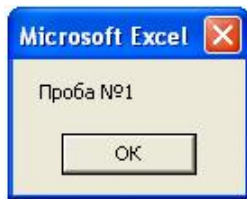


Рис.1.10 – Приклад процедури

Тепер після запуску макросу **ONE** за допомогою кнопки  на панелі «Visual Basic» (або меню **Запуск** редактора) на екран буде виведено наступне



вікно з повідомленням

### Додавання процедур в існуючий проект

Щоб додати процедуру в проект слід:

1. Відкрити вікно програмного коду модуля, в який потрібно додати процедуру за допомогою команди **Просмотр** контекстного меню модуля (або двічі клацнувши мишею на імені модуля, чи, відмітивши модуль, виконати команду **Code** (програма) в меню **Вид**, або функціональна клавіша або **F7**).
2. Вибрати в меню **Вставка (Insert)** **à** команду **Процедура (Procedure)**. При цьому виводиться діалогове вікно **Вставка процедури (Add Procedure)** (рис.1.11).

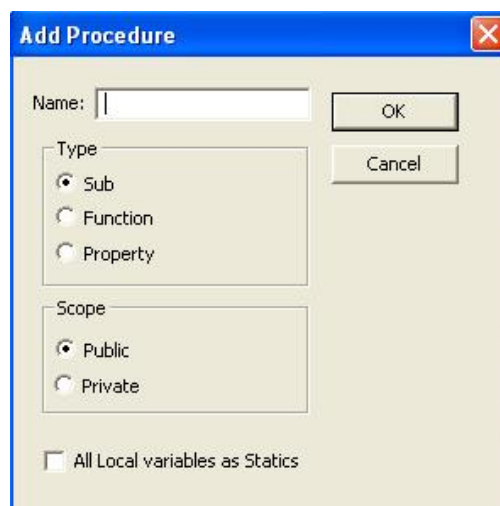


Рис.1.11 - Вікно Вставка процедури

3. Ввести ім'я процедури в поле **Имя (Name)**.

4. Вибрати перемикач, який задає тип процедури, що додається: **підпрограма, функція або властивість**.

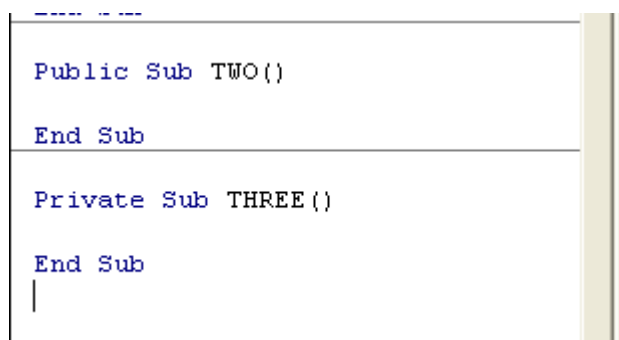
5. Задати загальну – **Загальна (Public)** або особисту – **Особиста (Private)** область визначення процедури, вибравши відповідний перемикач. (В VBA про змінні і константи, чия область видимості (Додаток 4) обмежена одним модулем, говорять, що вони мають область видимості **private**, а те, що доступно всім модулям всіх проектів, має область видимості **public**).

*Змінні й константи, описані на рівні модуля, доступні всім його процедурам і функціям. Хоча змінні й константи, описані на рівні модуля, доступні всім його процедурам і функціям, вони недоступні процедурам і функціям іншого модуля. Їхня область видимості обмежена модулем, у якому вони були визначені.*

6. Якщо локальні змінні є статичними, слід встановити прапорець **Всі локальні змінні вважати за статичні (All Local Variables As Statics)**.

*Поняття про статичні змінні розглянемо пізніше при вивченні мови.*

7. Натиснути кнопку **ОК**. У вікні модуля виводиться порожня процедура із заданим ім'ям (рис.1.12).



```
Public Sub TWO()  
End Sub  
  
Private Sub THREE()  
End Sub  
|
```

Рис.1.12 - Вікно модуля

**Примітка.** Зверніть увагу на один момент, пов'язаний з макросами Auto: оскільки раніше ці можливості активно використовувалися вірусами, в Office 2003 за замовченням ці макроси запускатися не будуть. Для того щоб

забезпечити їм можливість запуску, необхідно змінити встановлений рівень безпеки в меню **Сервіс | Макрос | Безпека** на **Низький**.

З метою більш поглибленого вивчення теоретичних основ даної лабораторної роботи рекомендується використати конспект лекцій з курсу та список рекомендованої літератури для даних методичних вказівок.

## **1.2. Опис лабораторних засобів та обладнання**

Лабораторна робота виконується на персональному комп'ютері стандарту IBM PC під керуванням операційної системи MS Windows зі стандартним пакетом MS Office.

### **Заходи безпеки під час виконання лабораторної роботи**

Заходи безпеки, яких треба дотримуватись при виконанні лабораторної роботи «Автоматизація обчислень в MS Excel за допомогою макросів» наведено у Додатку 5.

## **1.3. Послідовність виконання роботи**

Підготовка лабораторної роботи передбачає самостійне створення макросу, що включає алгоритм **схеми єдиного ділення** (Додаток 6) для чого:

5. Записати в макрос послідовність команд для створення шаблону **схеми єдиного ділення** в будь-якому листі MS Excel для чого:
  - відкрити книгу MS Excel;
  - почати запис макросу (як ім'я макросу використовувати прізвище студента; збереження макросу здійснити в поточній книзі (**Ета книга**)); комбінацію клавіш для запуску макросу не призначати;
  - ввести в потрібні комірки листа MS Excel текстові фрагменти і формули схеми єдиного ділення;
  - зупинити запис макросу.

6. Запустити макрос на виконання, відкривши інший лист даної книги MS Excel (у іншій книзі MS Excel).
7. Після запуску макросу ввести в лист MS Excel потрібні дані (відповідно до свого варіанту завдання) і одержати рішення системи рівнянь.
8. Запустити редактор Visual Basic. Відкрити створений макрос у вікні **програми**. Експортувати макрос в документ MS Word і зберегти його під ім'ям Сх-єд-ділVBA.
9. Роздрукувати текст макросу і додати його до протоколу лабораторної роботи.

### **Виконання лабораторної роботи**

Відповідно до отриманого завдання (варіант виданий викладачем – див. Додаток б) виконати наступне:

1. Записати в макрос послідовність команд для створення шаблону таблиці у відповідності до наданого варіанта в будь-якому листі MS Excel для чого:
  - відкрити книгу MS Excel;
  - почати запис макросу (як ім'я макросу використовувати прізвище студента та номер лабораторної роботи (наприклад, Rudenko-L1); збереження макросу здійснити в поточній книзі (**Ета книга**); комбінацію клавіш для запуску макросу не призначати;
  - ввести в потрібні комірки листа MS Excel необхідні текстові фрагменти;
  - зупинити запис макросу;
  - зберегти поточну книгу MS Excel в папці, що вказана викладачем.
2. Запустити макрос на виконання, відкривши інший лист даної книги MS Excel (або у іншій книзі MS Excel).

3. Після запуску макросу ввести в лист MS Excel потрібні дані (відповідно до свого варіанту завдання).
4. Зберегти макрос з внесеними даними під іншим ім'ям. Експортувати макрос в документ MS Word і зберегти його під ім'ям VBA-Прізвище виконавця в папці **“Word”** .
5. Продемонструвати роботу викладачу.
6. Оформити протокол лабораторної роботи.

#### **1.4. Оброблення та аналізування результатів. Оформлення звіту**

При оформленні звіту з лабораторної роботи до заздалегідь підготованого протоколу (див. завдання до самостійної роботи) додається роздрукований аркуш з результатами виконаної роботи.

#### **Контрольні запитання**

1. Що таке макрос? Навіщо створюють макроси?
2. Ознаки візуальної мови програмування?
3. Що таке об'єктно-орієнтоване програмування?
4. Відмінність універсальної мови програмування VB від VBA?
5. Як створити макрос?
6. Правила формування імені макросів?
7. Можливі місця збереження макросу?
8. Якими службовими словами починається і закінчується макрос?
9. Як можна подивитися та відредагувати текст макроса?
10. Як запустити макрос на виконання?
11. Основні елементи інтерфейсу редактора VB?
12. Які дочірні вікна містить редактор VB?
13. Призначення вікна проекту та кнопок вікна?
14. Призначення вікна властивостей і його вкладок?



15. Призначення вікна програми (програмного коду)? Як відокремлюються макроси у вікні?
16. Якими кольорами VBA розфарбовує різні частини макросу?
17. Що таке модуль та проект?
18. Як перейменувати модуль?
19. Як додати процедуру в існуючий проект?
20. Області визначення (видимості) процедур?

## ЛАБОРАТОРНА РОБОТА № 2

### РОЗРОБЛЕННЯ ТА ПРОГРАМУВАННЯ ЛІНІЙНИХ АЛГОРИТМІВ

**Мета та основні завдання роботи.** Дослідити роботу і можливості роботи оператора присвоєння, функцій *InputBox* і *MsgBox*. Ознайомитися з основними типами даних VBA (арифметичні та рядкові) і можливостями їх оголошення. Вивчити правила запису вбудованих (стандартних) функцій VBA та правила запису коментарів в тексті програми.

**Вивчити та описати (стисло) в протоколі лабораторної роботи:**

- роботу оператора присвоєння, функцій *InputBox* і *MsgBox*;
- правила запису коментарів в тексті програми та правила запису стандартних функцій VBA.

#### 2.1. Основні теоретичні відомості

##### Розробка програми (програмного коду) на VBA

Будь-яка програма, яка складена користувачем, є набором інструкцій, які послідовно виконуються. Редактор VBA включає ряд засобів, які значно полегшують написання коду. Ці засоби виконують аналіз введеного тексту й пропонують тільки припустимі типи даних, вбудовані шаблони процедур, функцій і т. п.

Будь-яка програма, написана алгоритмічною мовою VBA, повинна містити наступні частини:

- 1.Оголошення змінних й введення даних.
- 2.Обробка введених даних.
- 3.Виведення результатів.

##### Опис (оголошення) змінних і об'єктів

Типи даних VBA наведені в таблиці (Додаток 7). У редакторі VBA можна з легкістю описати об'єкти й змінні, використовуючи інструкцію **Dim**:

Формат:

**Dim** <Ім'я> **As** <Тип>

Наприклад: **Dim UserName as String**

Дана інструкція показує, що в програмі буде використатися строкова (символьна) змінна `UserName`.

Вбудована довідка VBA допомагає при описі змінних й об'єктів, що практично виключає можливість виникнення помилки (за умови використання списку припустимих типів даних й об'єктів).

При описі змінної (або об'єкта) відображається спеціальне вікно, у якому можна вибрати припустимий тип (або об'єкт) після уведення:

**Dim something As** редактор відображає список всіх відомих об'єктів і типів даних (рис.2.1).



Рис.2.1 – Список відомих об'єктів і типів даних

Після виведення списку треба вибрати у ньому необхідний елемент (або об'єкт) і нажати клавішу `<Enter>`. VBA підставляє виділений елемент в оператор **Dim**.

Зазначений список можна вивести скориставшись командою меню **Правка (Edit) → Список свойств/методов (List Properties/Methods)** або командою меню **Правка (Edit) → Завершить слово (Complete Word)**. Ці ж команди – **Список свойств/методов** і **Завершить слово** викликаються з контекстного меню, що виводиться при клацанні правою кнопкою миші по вікну модуля.

### Приклади оголошення змінних

```
Dim Nacl1 As Single
```

```
Dim Number As Integer, Dat As Date
```

`Dim Name As String` *'довжина довільна*

Для створення рядка фіксованої довжини можна скористатися інструкцією `Dim` з наступним синтаксисом:

`Dim State As String * 12` *'довжина 12 символів*

У цьому випадку створюється рядок з довжиною 12 символів.

`Dim Nt, Prof As Single`

У даному прикладі змінна `Nt` має тип `Variant` оскільки відсутнє її явне описання.

На відміну від інших мов програмування у VBA неможливо оголосити одночасно групу змінних, розділив їх між собою комами. Наприклад:

`Dim i, j, k As Integer`

Оператор не оголошує змінні `i`, `j`, `k` як `Integer`, хоча `i` записаний коректно. При такому запису тільки змінна `k` – є цілим числом (`Integer`). Не оголошені явно змінні `i`, `j` мають тип `Variant`.

Щоб оголосити всі змінні як `Integer` треба записати:

`Dim i As Integer, j As Integer, k As Integer`

У VBA можна вказати тип змінних, додаючи наприкінці імені змінної спеціальний символ. Такі символи називаються символами визначення типу (див. Додаток 7).

Змінні, оголошені в деякій процедурі, доступні тільки усередині цієї процедури (локальні змінні). Іноді необхідно, щоб одна змінна була доступна декільком процедурам. Це зазвичай робиться для того, щоб, обчисливши і зберігши деяке значення в змінній, яка доступна іншим процедурам, не повторювати подібні обчислення в інших процедурах.

Для того, щоб оголосити змінну, доступну всім процедурам в модулі, слід помістити інструкцію **Dim**, на самому початку модуля перед оголошенням всіх процедур.

У прикладі (рис.2.2) змінна `PI` оголошена глобальною і вона доступна всім процедурам в модулі.

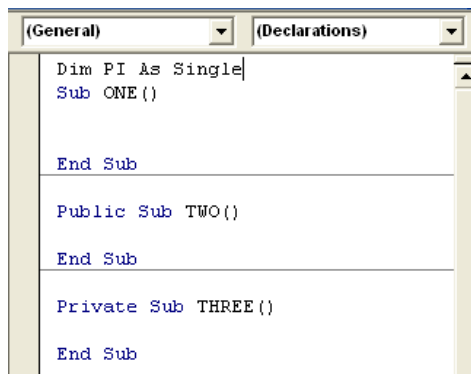


Рис.2.2 – Приклад оголошення змінних

Як відзначали раніше, змінні з невказаним типом мають тип *Variant*. Для того, щоб гарантувати відсутність помилок, внаслідок неявного помилкового оголошенням, в мові VBA існує команда Option Explicit. Коли застосовується ця команда VBA вимагає, щоб всі змінні були оголошені явно.

Для заборони неявного оголошення слід додати в область описання модуля, тобто в його початок, команду Option Explicit. Цю команду можна включити автоматично у всі новостворювані модулі. Для цього треба:

- 1.В меню редактора VBA **Сервіс**  $\rightarrow$  виконати команду **Параметри**. З'явиться діалогове вікно **Параметри** для налаштування параметрів проекту.
- 2.В діалоговому вікні **Параметри** на вкладці **Редактор (Editor)** встановити флажок **Require Variable Declaration (Вимагати опис змінних)**.
- 3.Клацнути по кнопці **ОК** (рис.2.3).

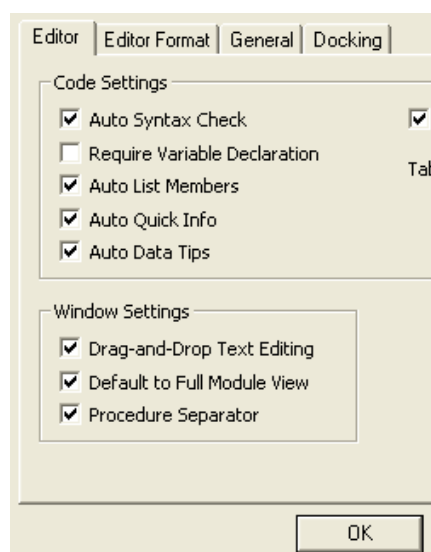


Рис.2.3 – Ввімкнення опису змінних автоматично у всіх новостворюваних модулях

Тепер на початку модуля буде з'являтися команда **Option Explicit** кожного разу, коли програма запису макросів або програміст буде додавати новий модуль в проект. Виконана операція впливатиме тільки на новостворювані модулі. А ось вставити команду у всі модулі, створені раніше, можна тільки вручну.

Для написання програм на мові VBA необхідно знати основні поняття алгоритмічної мови VBA.

### **Основні поняття алгоритмічної мови VBA**

Вивчення мови програмування і створення на ній програм – речі зв'язані. Перше вміщує знайомство з символікою, синтаксисом і семантикою мови, друге – у вмінні будувати алгоритми та реалізовувати їх у вигляді програм, з використанням при цьому знання мови.

Для цього розглянемо символіку мови VBA, правила утворення більш складних конструкцій – слів, виразів і операторів, а також прийоми створення програм з цих конструкцій.

Основні характеристики будь-якої мови програмування: алфавіт і символіка, синтаксис і семантика.

**Синтаксис** – це система правил, які визначають структуру конструкції мови.

**Семантика** – це відповідність між конструкціями мови.

Іншими словами синтаксис вказує, як потрібно писати речення мови, а семантика визначає сенс і правила використання цих речень.

Кожна конструкція мови повинна мати строго визначений сенс. Коли сенс порушується, ми говоримо про семантичну помилку.

### **Основні символи мови VBA**

Символіка всіх алгоритмічних мов (а також і VBA) складається із літер, цифр і спеціальних символів. Всі мовні конструкції можуть бути записані за допомогою основних символів:

1. Літерні символи – це великі і малі літери латинського алфавіту – 26 літер.

Це літери ( A...Z ) і ( a...z ).

2. Цифрові символи записують 10 арабськими цифрами:

0,1,2,3...9.

3. Спеціальні символи включають:

- Знаки арифметичних операцій: +, -, \*, /, ^.
- Знаки розділення: ., ,, :, ;, ( ), проміжок (пробіл), '(одинарна кавичка чи апостроф), ''(подвійна кавичка чи лапки).
- Знаки логічних операцій: =, <, > .
- Інші символи:
- !, # , \$ , % , & , @ , / , \_ , ....

З символів утворюються **ключові** (службові) слова.

**Ключове слово** – це послідовність символів, яка має особливий і назавжди встановлений сенс у даній мові.

### Типи даних в VBA

Програма оперує з об'єктами, які для ЕОМ прийнято називати даними. **Дані** – це деяка інформація (числова, текстова або інша), яка опрацьовується за допомогою комп'ютера. Тип даних вказується за допомогою операторів опису.

Перед роботою зі змінною рекомендується її оголосити.

У VBA передбачено 4 ключових слова для визначення області видимості змінних.

Інструкція **Dim** – використовується у більшості випадків. Якщо змінна оголошена за допомогою оператора Dim в області оголошень модуля, то вона буде доступна у всьому модулі, якщо у процедурі – тільки на час роботи цієї процедури.

Інструкція **Private** – діє як і оператор Dim при оголошенні змінних в стандартних модулях VBA. Відмінності проявляються лише під час створення своїх класів.

Інструкція **Public** – діє як і оператор Dim якщо змінна оголошена всередині процедури. А якщо змінна оголошена в області оголошень модуля, то вона буде доступна всім процедурам у всіх модулях даного проекту.

Інструкція **Static** вказує, що такі змінні можна використовувати тільки всередині процедури. Ці змінні видно тільки всередині процедури, в якій вони оголошені, але вони зберігають своє значення між різними викликами цієї процедури. Зазвичай це використовуються для накопичення будь-яких значень.

Якщо немає ніяких особливих вимог, то є сенс завжди вибирати інструкцію Dim.

Дані поділяються на **константи та змінні**.

**Константи** являють собою об'єкти, які не змінюються в процесі виконання програми. Використання констант покращує читабельність програми і зменшує ймовірність помилок при використанні постійних значень. Символьні константи заключають в лапки, а константи, що представляють дати, заключають в символи грати (#). Наприклад, **#1.12.2006#**.

Приклад:

**Константи**: 5, 6.74, 8.2E-03, "primer"

**Змінні**, на відміну від const, є такими об'єктами, значення яких можуть змінюватися в процесі виконання програми. Можна сказати, що змінна – це величина, яка зберігається в пам'яті і значення якої може змінюватися під час виконання програми.

Кожна змінна має власне унікальне ім'я (ідентифікатор), яке призначається програмістом і формується за наступними правилами:

- першим символом імені обов'язково має бути буква;
- ім'я може містити тільки букви, цифри і знак підкреслення («\_»);
- довжина імені не повинна перевищувати 255 символів;



– у одній і тій же підпрограмі (процедурі або функції) не можуть бути оголошені дві змінні з одним і тим же ім'ям.

Приклад:

**Змінні**: A3, BC, C1U\_T, ...

Кожна змінна при створенні повинна отримати певний тип. Тип змінної характеризує її довжину, спосіб уявлення і діапазон зміни тих значень, які можуть зберігатися в змінній.

Типи змінних, які використовуються у мові VBA, вказані у Додатку 7.

Імена змінним варто надавати керуючись змістом задачі і з врахуванням вищенаведених правил.

Наприклад.

Суму можна позначити буквою S, густину  $\rho$  буквами RO, в'язкість  $\mu$  буквами MU, корені рівняння  $x_1$  і  $x_2$  – X1, X2.

VBA не розрізняє регістри. Щоб зробити імена змінних легкими для читання, програмісти часто використовують змішаний регістр (наприклад, InteretDay, а не interetday).

Останнім символом в імені змінної може бути символ (!, #, %, ...) , який вказує на тип змінної (спадщина ранніх версій мови Бейсік).

Приклад:

**Змінні**: A%, BC!, C1#, ...

### **Іменовані (або користувацькі) константи**

VBA допускає вживання констант, які забезпечуються іменами. (Іноді кажуть, що числа можна позначити іменами.) Вони можуть розташовуватися в будь-якому місці програми, де використовуються звичайні змінні (за винятком лівої частини оператора присвоювання). Цій константі не можна присвоїти нове значення.

Константу зазвичай визначають на початку програми.

Формат:

**CONST < ім'я const > = вираз [, ім'я const = вираз]...**

**Ім'я const** – складається з букв латинського алфавіту і цифр, може закінчуватися символом, що вказує тип (!, #, %, ...).

**Вираз** – це число або арифметичний вираз, значення якого зберігається в константі.

Приклад:

```
CONST PI = 3.14159265
```

```
S = PI * D^2/4
```

Коли створюється іменована константа, то VBA приписує їй тип у відповідності з типом виразу, який їй присвоюється. Але можна вказувати тип константи і явно (кінцевий символ в імені константи або оголошення її з використанням оператора CONST наступного формату):

Формат:

```
CONST < ім'я const > As < тип > = вираз
```

Приклад:

```
CONST PI As Double = 3.14159265
```

Перевага використання іменованих констант (або користувацьких констант) полягає в тому, що ім'я константи може бути істотно коротшим за число або текст (при багатократному вживанні економиться місце і час набору).

У мові VB (VBA) є **вбудовані** (внутрішні) константи, імена яких зарезервовані. Імена констант VB починаються з vb, а константи Excel VBA починаються з xl. Внутрішні константи слугують для спрощення роботи з деякими вбудованими процедурами VBA (або з властивостями і методами об'єктів Excel). Для переглядання повного переліку вбудованих констант можна задати пошуковий рядок **Constants** списку **Classes** в **Object Browser** (меню **View (Перегляд)** редактора VBA, або кнопка на панелі інструментів).

### **Вирази і операції**

Над даними можуть здійснюватися різні дії, що називаються операціями.

Вираз складається з текстових, числових або інших даних (const та змінних), сполучених знаками операцій. Вирази в VBA можуть бути:

- числовими (арифметичними);
- відношення;
- логічними;
- рядковими (або символічними);
- з датами;
- об'єктні.

Знаки операцій розглянули при ознайомленні з символікою мови.

### **Вбудовані (стандартні) функції VBA**

Як і в більшості інших мов програмування, в VBA є ряд **вбудованих функцій**, які спрощують обчислення і операції. Функція – це вбудована формула за якою оброблюється аргумент і формується значення результату. Багато функцій VBA подібні (або ідентичні) до функцій Excel. За стандартними функціями закріплюються стандартні імена, які не можна використовувати в якості ідентифікаторів (імен змінних). Вбудовані функції VBA за призначенням можна поділити на категорії: математичні функції, строкові (текстові) функції, функції перетворення даних, функції взаємодії та функції, що надають інформацію про диск.

Перелік основних математичних функцій:

ABS(x) – абсолютна величина (модуль числа);

ATN(x) – обчислення arctg;

COS(x) – обчислення cos;

SIN(x) – обчислення sin;

TAN(x) – обчислення tg;

SQR(x) – обчислення квадратного кореня;

LOG(x) – обчислення натурального логарифма;

EXP(x) – обчислення експоненти;

INT(x) – знаходження ближнього цілого, що не перевершує x;

FIX(x) – виділення цілої частини x;

RND[(x)] – генерація чергового випадкового числа (аргумент не є обов'язковим).

## Основні функції перетворення даних

**VAL(S)** – повертає числове значення відповідне числу, що представлено рядком S;

**Cdbl(E)** – повертає числове значення типу **Double** (аргумент E – будь-який числовий або рядковий вираз, який може бути перетворений в число);

**CInt(E)** – повертає числове значення типу **Integer** (аргумент E – будь-який числовий або рядковий вираз, який може бути перетворений в число);

**CLng(E)** – повертає числове значення типу **Long** (аргумент E – будь-який числовий або рядковий вираз, який може бути перетворений в число);

**CSng(E)** – повертає числове значення типу **Single** (аргумент E – будь-який числовий або рядковий вираз, який може бути перетворений в число);

**CStr(E)** – повертає числове значення типу **String** (аргумент E – будь-який числовий або рядковий вираз).

**Str(число)** – повертає значення типа variant (string), що є строковим представленням числа.

*Функція **Str(число)**, як допустимий десятковий роздільник сприймає тільки точку. За наявності іншого десяткового роздільника (наприклад, коми) для перетворення чисел в рядки слід використовувати функцію **Cstr**.*

Та інші, які можна подивитись в описі мови.

## Правила запису стандартних функцій

Стандартна функція записується таким чином: спочатку записується ім'я функції, а потім в круглих дужках її аргумент.

Приклади :

- $\sin x$  – SIN(x)
- $\sin(x+1)$  – SIN(x+1)
- $\sin \text{tg} x$  – SIN(TAN(x))
- $\sin^2 \sqrt{\text{tg} x}$  – SIN(SQR(TAN(x)))^2

Для того, щоб знайти повний перелік вбудованих доступних функцій можна використати вікно Object Browser (меню **View (Просмотр)** редактора

VBA або кнопку на панелі інструментів). Це вікно дозволяє знайти потрібну функцію, продивитись формат її запису і вставити шаблон виклику функції в текст програми (через копіювання її в буфер обміну і вставки в потрібне місце програми за допомогою комбінації клавіш Ctrl + V).

При необхідності користувач може створювати свої користувацькі (нестандартні) функції, робота з якими буде розглянута пізніше. Імена нестандартних функцій користувач конструює сам.

### **Оператори VBA**

Мінімальною структурою в програмі, яка виражає закінчену думку, є оператор (інструкція). Оператори є основними конструкціями мови. Саме за допомогою операторів реалізується алгоритм роботи програми. У сучасних мовах існує величезна кількість операторів. Наприклад, арифметичні оператори призначені для виконання основних арифметичних операцій над операндами, які можуть виступати змінні, числа і константи. Знаками цих операторів є загальноприйняті математичні знаки „+”, „-”, „\*”, „/”. Наприклад, оператор “+” виконує операцію складання двох чисел або виразів, що є операндами. Більшість арифметичних операторів VBA вимагає наявності двох операндів.

Набір операторів VBA дуже широкий і не поступається в цьому таким мовам, як Паскаль і Сі.

Виконання програми йде зверху вниз, по рядкам. Мінімальний програмний рядок – один оператор.

### **Коментарі**

Текст програми може супроводжуватися коментарями, тобто текстом, який пояснює важливі моменти програми, але не впливає на хід виконання програми.

*(Якщо довго не користуватися програмою, то коментарі можуть допомогти заново розібратися в програмі. Свої програми слід починати з коментарів: прізвище, ім'я, група, номер роботи).*

Коментар – це описовий текст, який включається в програмний код. VBA ігнорує текст коментаря. Коментар можна записати в новий рядок або вставити в кінець програмного рядка.

Коментарі в програмі записують наступним чином:

Формат:

`' текст коментаря`

Приклад:

`' Розробив Іванов, гр. ХЕ-101`

Текст, що введений після апострофа, VBA сприймає як коментар (за винятком випадків, коли апостроф поміщений в лапки).

Наприклад:

`"Can't continue".`

В VBA коментарями можна завершувати програмний рядок (справа за коментарем записувати оператори вже не можна – вони увійдуть до тексту коментаря).

Приклад:

`K=Z+FNA (x, y) ' Розрахунок потужності K`

Коментар можна також записати після ключового слова `Rem` (спадщина старих версій BASIC, включена в VBA з міркувань сумісності).

Формат:

`REM текст коментаря`

Коментар, що записаний після `Rem` розташовується тільки в окремому рядку і не допускається застосовувати його в рядку після операторів.

Наприклад:

`REM Програма розрахунку кінетичних констант`

Досвідчені програмісти рекомендують використовувати коментарі в наступних випадках:

- для короткого опису призначення кожної створеної процедури;
- для опису змін, які вносять до процедури;

- для вказівки функції або конструкції, що використовуються незвичним або нестандартним способом;
- для опису призначення змінних (щоб і розробник програми, і інші користувачі могли тлумачити назви змінних, які без коментаря зрозуміти складно);
- для опису способів, які допоможуть уникнути впливу внутрішніх помилок Excel на застосування їх в програмі.



Коментарі слід додавати у програму у процесі написання кода.

Інколи виникає необхідність перевірити роботу програми без деяких програмних рядків (рядка). Замість того, щоб видаляти їх з програми, а потім знов додавати, можна перетворити цей фрагмент програмного коду в коментар. Для поновлення цих програмних рядків потрібно просто видалити апостроф.

Ці дії можна досить легко виконати за допомогою кнопок панелі інструментів Edit редактора VBA (рис.2.4) .



Рис.2.4 – Панель інструментів Edit

Спочатку треба виділити групу інструкцій, а потім використати кнопку Comment Block , щоб перетворити ці інструкції в коментарі. Кнопка Uncomment Block  перетворює групу коментарів назад в інструкції. Вказані кнопки застосовуються дуже часто, тому з метою підвищення зручності роботи їх можна скопіювати їх на стандартну панель інструментів.

### Оператор присвоювання

Оператор присвоювання, що задається необов'язковим службовим словом **Let**, використовується для присвоєння змінній значення виразу або властивості.

Формат:

**[LET] < ім'я змінної > = < вираз >**

Зліва від знаку рівності вказується (ідентифікатор), а справа – будь-який вираз. Значення виразу, що стоїть справа обчислюється і присвоюється змінній, що стоїть зліва. Після виконання цього оператора значення виразу засилається в область пам'яті, виділену для зберігання змінної. Це один з основних операторів мов програмування. При присвоєнні значення необхідно стежити за тим, щоб тип змінної відповідав типу виразу, розташованому праворуч від знаку рівності.

Так змінній числового типу можна присвоїти значення арифметичного виразу (числової константи, числової змінної), а рядковій змінній – значення рядкового виразу (рядкової константи, рядкової змінної) і так далі.

Наприклад:

$$A = 3.5$$

$$AB = 3.5 + 2 + 6.7 * 4.1$$

$$T = V$$

$$C = A + V$$

$$K = K + 1$$

Звернемо увагу на запис:  $K = K + 1$ .

Нехай перед виконанням цього оператора присвоєння змінна  $K$  має значення 5. При виконанні оператора присвоєння  $VBA$  спочатку виконує праву його частину: бере з пам'яті поточне значення змінної  $K$  – 5 і додає до нього 1, а потім отримане значення, що дорівнює 6 присвоює змінній  $K$ , що знаходиться зліва від знака «=». Попереднє значення змінної втрачається.

Кожна операція має пріоритет. Вирази обчислюються і програмуються згідно із пріоритетами математики. Обчислення проводяться зліва направо в порядку пріоритету операцій. Порядок дій може бути змінений за допомогою дужок.

Пріоритет виконання арифметичних операцій (правила математики):

1. Дії в дужках.
2. Обчислення стандартної функції.
3. Піднесення до степеня.



4. Множення і ділення (рівнозначні).
5. Додавання і віднімання (рівнозначні).

Приклади:

<b>Математичний запис</b>	<b>Програмна частина</b>
$y = 2x^2$	<code>y=2*x^2</code>
$y = \frac{a+b}{c+d}$	<code>y=(a+b)/(c+d)</code>
$y = \frac{abc}{dkn}$	<code>y=(a*b*c)/(d*k*n)</code> або <code>y=a*b*c/(d*k*n)</code> або <code>y=a*b*c/d/k/n</code>
$y = \frac{\sin^2 tg^3 x^4}{\sqrt{a+b} + \sqrt[5]{d+1}}$	<code>y=SIN(TAN(X^4)^3)^2/(SQR(a+b)+(d+1)^(1/5))</code>
$y = e^x + e^{2x} - e^{\sin x+1}$	<code>Y=EXP(x)+EXP(2*x)-EXP(SIN(x)+1)</code>

При записі тексту програм для спрощення читання, налагодження і модифікації програми зручніше кожний оператор розташовувати в окремому рядку. Цей стандарт не є обов'язковою вимогою і при необхідності можна в одному програмному рядку розміщувати декілька операторів розділивши їх між собою символом двокрапка (:).

Приклад:

```
Sub One()
f=1: a=2: d=3: x=f*a + d^2
End Sub
```

Багато програмістів для полегшення читання програм рекомендують дотримуватися правила: "Один рядок – один оператор". Тоді наведений приклад буде виглядати:

```
Sub One()
f=1
a=2
d=3
```

```
x=f*a + d^2
```

```
End Sub
```

**Введення рядка**, максимальна довжина якого в VBA не повинна перевищувати 1024 символа, виконується натисненням клавіші Enter.

Якщо довжина оператора занадто велика і його текст не видно повністю на екрані дисплея, що ускладнює читання і розуміння програми, то в цьому випадку оператор слід продовжити в інших рядках. Щоб продовжити (перенести) оператор на наступний рядок, використовується пара символів символ пропуску (пробіл) та символ підкреслення " \_".

Приклад:

```
Sub LongOperator( )  
y = 3*SQR(r+m/n) - SIN(x^2) _  
+t/(x+r) + m/n  
End Sub
```

У змінних до присвоєння ним значень користувачем зберігається:

- у змінних всіх числових типів даних значення 0;
- у строкових змінних змінної довжини – "" (рядок нульової довжини).

Програмний рядок можна відмітити за допомогою **мітки**. В якості мітки можна використати алфавітно-цифрову конструкцію, що записується в окремому рядку і закінчується двокрапкою, або цифрову – число без двокрапки, що записано перед оператором. Процедури VBA можуть містити будь-яку кількість міток. Але однією і тією ж міткою може бути помічений тільки один рядок програми.

Приклади:

```
20 x=1
```

```
m1:
```

```
y=1
```

З урахуванням вищесказаного загальний формат програмного рядка можна записати таким чином:

```
[мітка] [оператор 1] [:оператор 2] ... [:оператор N] ['коментар]
```

З цього запису видно що програмний рядок може бути пустим (на хід програми це не впливає, але структурно розділяє частини програми ).

### **Оператор конкатенації (об'єднання) рядків**

В мові VBA можна об'єднувати рядки, утворючі рядок більшої довжини. Приєднання одного рядка до іншого називається **конкатенація**. Оператор & (амперсанд) в VBA використовується тільки для об'єднання рядків.

Формат: **Операнд1 & Операнд2 [& Операнд3..]**

**Операнд** – це будь-який допустимий строковий або чисельний вираз (який перетворюється в строковий). Тип даних результату конкатенації рядків – **String**.

Символ (&) операції конкатенації обов'язково необхідно відокремлювати пропуском від імені змінної, бо інакше VBA може інтерпретувати цей символ як символ визначення типу **Long**.

Для конкатенації також можна використати символ «+» (спадщина старих версій BASIC). Синтаксис і дія цього знаку нічим не відрізняється від знаку &. При використанні оператора «+» не завжди вдається визначити, відбудеться об'єднання рядків або операція складання. Щоб уникнути неоднозначності і отримати код програми, що буде зрозумілим та легко сприйматись, краще використовувати для об'єднання рядків оператор &.

Приклад:

Якщо рядкові змінні А і В перед виконанням конкатенації рядків мали відповідно значення “John ” і “Lennon”, то після її виконання

`C = A & B`

або `C = A + B`

рядкова змінні С отримає значення “John Lennon”.

Приклад:

`sLastName = "Иванов "`

`sFirstName = "Иван "`

`sSecondName = "Иванович "`

```
sName = sLastName & sFirstName & sSecondName
```

В результаті конкатенації змінна **sName** отримає значення

"Иванов Иван Иванович".

### Оператори введення і виведення даних

Оператори введення/виведення призначені для обміну інформацією із зовнішніми пристроями. Для написання ефективно працюючої програми необхідно отримати дані, зберегти їх в змінній і правильно обробити. Дані отримані від користувача називають введеними даними або просто введенням.

*Примітка. Змінним можна надати значення і за допомогою оператора присвоювання.*

### Введення даних. Функція `InputBox`

Найпростіший спосіб прийняти інформацію від користувача – це скористатися функцією `InputBox`. Функція `InputBox` використовується для отримання даних від користувача. Виводить на екран діалогове вікно з текстом, що пропонує введення значення і містить повідомлення та поле введення. При цьому встановлюється режим очікування введення тексту користувачем. В вікні також є кнопки **OK** і **Cancel** (Отмена). Коли користувач натискає кнопку OK або клавішу ENTER, функція `InputBox` повертає вміст поля введення (значення типу **String**, що містить текст, введений в поле введення). Якщо користувач натискає кнопку Cancel (Отмена), функція поверне рядок нульової довжини ("").

Формат (повний):

```
InputBox(prompt[, title] [, default] [, xpos] [, ypos] [, helpfile, context ])
```

Для кращого розуміння складових функції `InputBox` наведемо їх з перекладом:

`InputBox` ("запрошення – повідомлення"[, "заголовок вікна"] [, "дані за замовчуванням"] [, відстань по горизонталі між лівою межею діалогового вікна і лівим краєм екрану] [, відстань по вертикалі між верхньою межею

діалогового вікна і верхнім краєм екрану] [, "ім'я Help-файла" , номер розділу Help-файла])

Складові функції *InputBox*:

**prompt** – строковий вираз, це те, що відображається як **повідомлення** в діалоговому вікні. Максимальна довжина рядка **prompt** складає приблизно 1024 символа. Строкове значення **prompt** може містити декілька фізичних рядків (для розділення рядків використовують: Chr(13), Chr(10) або комбінацію цих символів Chr(13) & Chr(10)). **prompt** – це єдиний обов'язковий аргумент функції *InputBox*.

**title** – строковий вираз (заголовок вікна), це те, що відображається в рядку заголовка діалогового вікна. Якщо цей аргумент опущений, в рядок заголовка відображається ім'я додатку (наприклад, Microsoft Excel).

**default** – строковий вираз (за замовчуванням), це те, що відображається в полі введення як значення використовуване за замовчуванням, якщо користувач не введе інший рядок. Якщо цей аргумент опущений, поле введення відображається порожнім.

**xpos** – числовий вираз, що задає відстань по горизонталі між лівою межею діалогового вікна і лівим краєм екрану (у твіпах, де 1 твіп = 1/20 точки = 0,0007 дюйма; 1 піксель приблизно рівний 27 твіпам). Якщо цей аргумент опущений, діалогове вікно вирівнюється по центру екрану по горизонталі.

**ypos** – числовий вираз, що задає відстань по вертикалі між верхньою межею діалогового вікна і верхнім краєм екрану (у твіпах). Якщо цей аргумент опущений, діалогове вікно відображається по вертикалі приблизно на одну третину висоти екрану.

**helpfile** – строковий вираз, що визначає ім'я файлу довідки в якому знаходяться довідкові відомості про дане діалогове вікно. Разом з цим аргументом слід вказувати також і аргумент **context**.

**context** – числовий вираз, що визначає номер відповідного розділу довідки (вказується одночасно з аргументом **helpfile**).

На початку роботи з цією функцією краще використовувати скорочений формат:

```
Рядок = InputBox(Запрошення [,Заголовок вікна][, Дані за замовчуванням])
```

**Рядок** – будь-яка текстова змінна, в якій може зберігатися текстовий рядок (тип String або Variant).

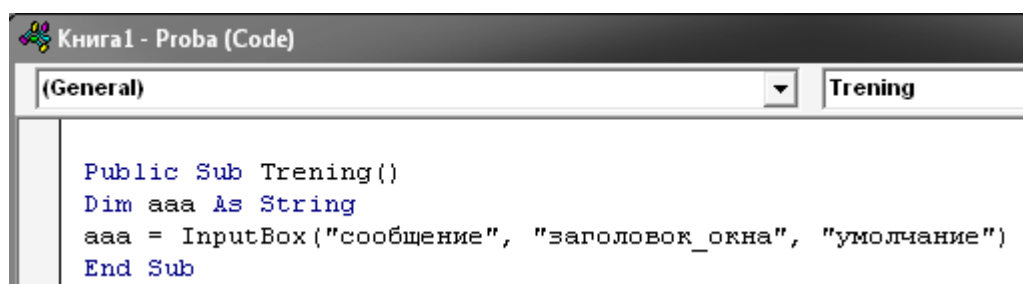
**Запрошення** – текст запрошення (текстова змінна, текстовий вираз або константа (може бути іменована)). Цей аргумент є обов'язковим.

**Заголовок вікна** – назва вікна в рядку заголовка (текстова змінна, текстовий вираз або константа (може бути іменована)). Цей аргумент є не обов'язковим. Якщо його опустити, то як заголовок буде виведено Microsoft Excel.

**Дані за замовчуванням** – це те, що відображається в полі введення як значення використовуване за замовчуванням (текстова константа).

Приклад :

Фрагмент програми із використанням **InputBox** (рис.2.5).



```
Public Sub Тренинг()  
Dim aaa As String  
aaa = InputBox("сообщение", "заголовок_окна", "умолчание")  
End Sub
```

Рис.2.5 – Код програми

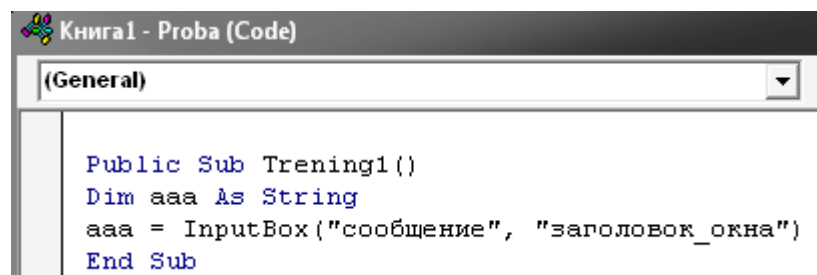
Результат виконання цієї програми зображено на рис.2.6.



Рис.2.6 – Результат виконання програми

Приклад :

Фрагмент програми із використанням **InputBox** (рис.2.7).



```
Public Sub Trening1()  
Dim aaa As String  
aaa = InputBox("сообщение", "заголовок_окна")  
End Sub
```

Рис.2.7 – Код програми

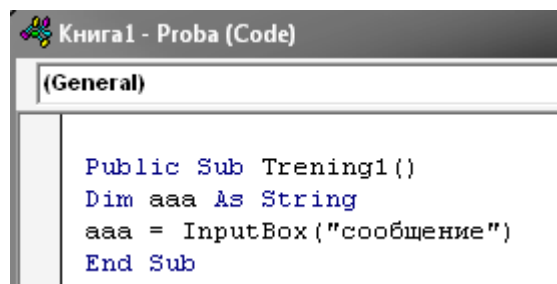
Результат виконання цієї програми зображено на рис.2.8.



Рис.2.8 – Результат виконання програми

Приклад:

Фрагмент програми із використанням **InputBox** (рис.2.9).



```
Public Sub Trening1()  
Dim aaa As String  
aaa = InputBox("сообщение")  
End Sub
```

Рис.2.9 – Код програми

Результат виконання цієї програми зображено на рис.2.10.

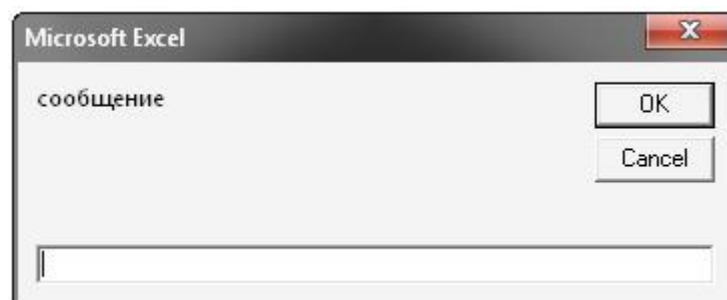
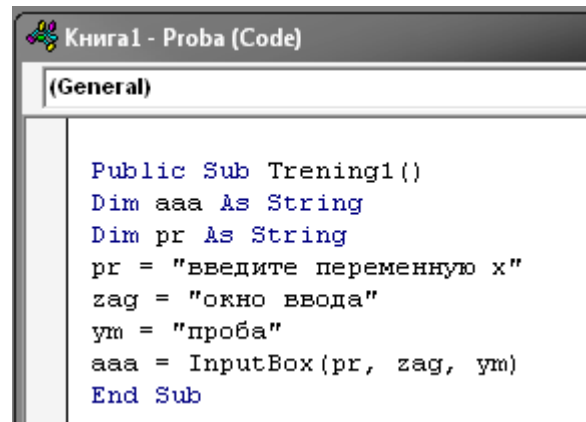


Рис.2.10 – Результат виконання програми

Приклад:

Фрагмент програми із використанням **InputBox** (рис.2.11).



```
Public Sub Trening1()  
Dim aaa As String  
Dim pr As String  
pr = "введите переменную x"  
zag = "окно ввода"  
um = "проба"  
aaa = InputBox(pr, zag, um)  
End Sub
```

Рис.2.11 – Код програми

Результат виконання цієї програми зображено на рис.2.12.

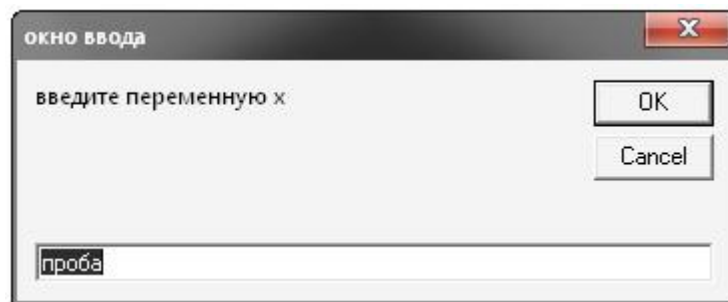


Рис.2.12 – Результат виконання програми

Оскільки функція **InputBox** повертає вміст поля введення – значення типу **String**, то для введення числових даних за допомогою цієї функції необхідно виконати перетворення рядкових даних в числові. Для цього можна скористатись функцією **VAL(S)**, яка повертає числове значення відповідне числу, що представлено рядком **S**.

Приклад:

```
Sub dva()  
Dim a As Double, x As Double  
a = Val(InputBox("Введите a"))  
x = Val(InputBox("Введите x"))  
End Sub
```



## Виведення даних

### Функція MsgBox

Функція *MsgBox* дає користувачу простий спосіб відображення повідомлення. Вона відображає повідомлення в діалоговому вікні, чекає натиснення кнопки і повертає ціле число, що визначає натиснуту кнопку.

За допомогою цієї функції можна виводити результати розрахунків у вигляді повідомлення в діалоговому вікні.

Формат (повний):

```
MsgBox(prompt[, buttons] [, title] [, helpfile, context])
```

Для кращого розуміння складових функції *InputBox* наведемо їх з перекладом:

```
MsgBox("повідомлення"[, кнопки] [, "заголовок вікна"] [,  
"ім'я Help-файла" , номер розділу Help-файла])
```

Складові функції *MsgBox*:

**prompt** – строковий вираз, це те, що відображається як **повідомлення** в діалоговому вікні. Максимальна довжина рядка **prompt** складає приблизно 1024 символа. Строкове значення **prompt** може містити декількох фізичних рядків (для розділення рядків використовують: Chr(13), Chr(10) або комбінацію цих символів Chr(13) & Chr(10)). **Prompt** – це єдиний обов'язковий аргумент функції *MsgBox*.

**buttons** – числовий вираз, що представляє суму значень, які вказують число і тип кнопок, що відображаються, тип використовуваного значка, тип основної кнопки за замовчуванням і модальність вікна повідомлення (Додаток 8). Якщо параметр **Buttons** пропущено, то за замовчуванням використовується нульове значення.

**title** – строковий вираз (заголовок вікна), це те, що відображається в рядку заголовка діалогового вікна. Якщо цей аргумент пропущено, у рядку заголовка відображається ім'я додатку (наприклад, Microsoft Excel).

**helpfile** – строковий вираз, що визначає ім'я файлу довідки в якому знаходяться довідкові відомості про дане діалогове вікно. Разом з цим аргументом слід вказувати також і аргумент `context`.

**context** – числовий вираз, що визначає номер відповідного розділу довідки (вказується одночасно з аргументом `helpfile`).

Як видно всі аргументи функції `MsgBox` відповідають тим же призначенням і вимогам, що і їх прототипі функції `InputBox`. Слід зауважити, що позиціонування діалогового вікна у функції `MsgBox` не передбачено. Діалогове вікно функції **MsgBox** завжди виводиться у центрі екрану.

Аргументи функції необхідно перераховувати в певному порядку. Так для функції `MsgBox` перший аргумент є повідомленням для відображення, другий – це число і тип кнопок для діалогового вікна, третій, – заголовок діалогового вікна і так далі. Про це підказує спливаюче вікно "Короткі відомості", що з'являється при введенні імені функції.

Слід звернути увагу на те, що якщо другий (необов'язковий) аргумент опустити, то все ж таки необхідно включати розділюючі коми для другого аргументу в список аргументів.

Приклад (рис.2.13):

```
MsgBox "сообщение" , , "заголовок"
```

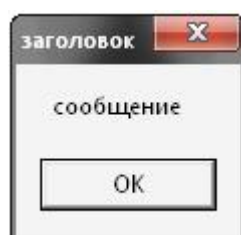


Рис.2.13 – Результат виконання рядку коду

На початку роботи з цією функцією краще використовувати скорочений формат. У простій формі запису функції **MsgBox** після ключового слова слідує тільки текст повідомлення. Текст повідомлення є рядком символів. Як текст повідомлення можна використовувати рядки символів об'єднані за допомогою оператора конкатенації(&).

Приклад (рис.2.14):

```
MsgBox "сообщение "
```

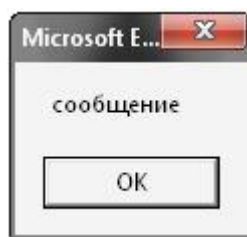


Рис.2.14 – Результат виконання рядку коду

Оскільки **Prompt** – це єдиний обов'язковий аргумент функції *MsgBox* є значенням типу **String**, то для виведення числових даних за допомогою цієї функції необхідно виконати перетворення числових даних в рядкові. Для цього можна скористатись функцією **CStr(E)**, що – повертає числове значення типу **String** (аргумент E – будь-який числовий або рядковий вираз).

Приклад (рис.2.15):

```
Public Sub coob()  
Dim a As Single  
a = 18.6758  
MsgBox CStr(a)  
End Sub
```



Рис.2.15 – Результат виконання програми

### Оператор **Debug.Print**

Оператор **Debug.Print** слугує для виведення значень у вікно **Immediate**.

Вікно **Immediate** в VBA використовується для тестування операторів і перевірки різних виразів VBA. Деякі програмісти рекомендують завжди відображати вікно **Immediate** на екрані, оскільки воно часто використовується для перевірки виразів і при налаштуванні коду. Вікно **Immediate** можна також

застосувати для швидких підрахунків. Для цього необхідно після знаку питання (?) набрати необхідний вираз і натиснути клавішу ENTER. Вікно Immediate виводиться на екран при натисканні комбінації клавіш **Ctrl + G** або командою **Immediate Window** в меню **View**.

Кожен оператор *Debug.Print* виводить дані у новий рядок. Оператор *Debug.Print* підтримує параметри форматування, що характерні для оператора *Print #*. В оператор можна включати багато виразів з використанням у якості роздільника коми (,) і крапки з комою (;):

- значення записуватимуться підряд, без проміжків між ними (роздільник крапка з комою);
- значення записуватимуться в 14-символьні зони виводу (роздільник кома).

Крім того, в списку значень оператора можуть бути присутніми функції:  
**Spc(n)** – для вставки *n* пропусків між значеннями у текстовому рядку;  
**Tab(n)** – для вказівки номера позиції рядка *n* для запису наступного значення.

У список виведення можуть бути включені вирази будь-якого типу (числові або символічні).

Приклад (рис.2.16):

```
Sub ff()  
x = a + b  
a= 1: b= 7: x= 8  
Debug.Print "a="; a, "b="; b, "x="; x  
End Sub
```

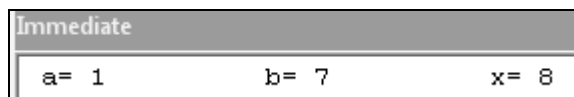


Рис.2.16 – Результат виконання програми

Приклад (рис.2.17):

```
Sub a()  
    `Начало программы  
Dim X As Double `Определяем x
```

```

X = (7)^(1/27) `Вычисляем x
`записываем формулу на языке программирования
Y = 2*x^2 + 3*(SIN(X + 2)^2)^3
`выводим результат на экран монитора
MsgBox ("ответ=" + Str(Y))
End Sub

```

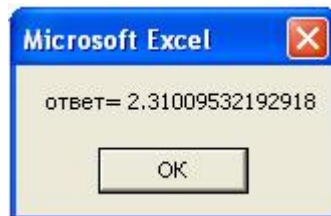


Рис.2.18 – Результат виконання програми

З метою більш поглибленого вивчення теоретичних основ даної лабораторної роботи рекомендується використати конспект лекцій з курсу та список рекомендованої літератури для даних методичних вказівок.

## 2.2. Опис лабораторних засобів та обладнання

Лабораторна робота виконується на персональному комп'ютері стандарту IBM PC під керуванням операційної системи MS Windows зі стандартним пакетом MS Office.

### Заходи безпеки під час виконання лабораторної роботи

Заходи безпеки, яких треба дотримуватись при виконанні лабораторної роботи «Автоматизація обчислень в MS Excel за допомогою макросів» наведено у Додатку 5.

## 2.3. Послідовність виконання роботи

Згідно із отриманим завданням виконати наступне:

### *Частина 1*

1. Набрати у вікні редактора приклад програми рішення квадратного рівняння, що надано викладачем (програма №1 – Додаток 9).
2. Підібравши необхідні значення коефіцієнтів отримати різні варіанти роботи програми:
  - a. немає дійсних коренів;
  - b. корені рівні:  $x_1 = x_2$ ;
  - c. два дійсні корені:  $x_1$  і  $x_2$ .
3. Додати у приклад програми два своїх коментарі, які пояснюють фрагменти програми.
4. Записати у протокол отримані результати і коефіцієнти рівняння, при яких вони були отримані.
5. Програму зберегти на диску в папці своєї групи (та іншому носії). Створити документ Microsoft Word, в який скопіювати текст програми і додати отримані результати роботи програми (всі варіанти).
6. Зберегти копію програми та вивчити на ній можливості редагування програм в середовищі VBA. Після цього одну з копій програми відредагувати за зразком (програма № 2 – Додаток 9). Перевірити роботу відредагованої програми і виконати з нею аналогічні дії (пункти 2–5).
7. Продемонструвати роботу програми викладачу.
8. Вихідні дані, програму та отримані результати надрукувати.
9. Оформити протокол лабораторної роботи.

## *Частина 2*

1. Згідно зі своїм варіантом (Додаток 10) розробити програму, яка обчислює значення заданого числового (числових) виразу (виразів).
2. Набрати у вікні редактора текст розробленої програми (всі змінні в програмі оголошуються явно). Включити у текст програми коментарі з інформацією про виконавця, академічну групу, тему і номер лабораторної роботи. Задавши необхідні значення змінних, що входять в розрахункові формули отримати результат роботи програми.

3. Програму зберегти на диску у папці своєї групи (та іншому носії). Створити документ Microsoft Word, в який скопіювати текст програми і додати отриманий результат роботи програми.
4. Змінивши значення змінних, що входять в розрахункові формули отримати новий результат роботи програми.
5. Новий результат роботи програми додати в документ Microsoft Word (з указанням новиз значень, що були надані змінним).
6. Записати у протокол отримані результати і значення змінних, при яких вони були отримані.
7. Продемонструвати роботу програми викладачу.
8. Вихідні дані, програму та отримані результати надрукувати.
9. Оформити протокол лабораторної роботи.

#### **2.4. Оброблення та аналізування результатів. Оформлення звіту**

При оформленні звіту з лабораторної роботи до заздалегідь підготованого протоколу (див. завдання до самостійної роботи) додаються роздруковані аркуші з результатами виконаної роботи (Частина 1 та Частина 2).

#### **Контрольні запитання**

1. Формат та призначення інструкції **Dim**?
2. Які типи арифметичних даних використовує VBA?
3. Особливості типу даних Variant?
4. Як записують коментарі в тексті програми?
5. Що таке локальні та глобальні змінні і як їх оголосити?
6. Навіщо потрібна команда **Option Explicit** і як нею користуватися?
7. Які символи мови VBA відносять до спеціальних?
8. Константи та змінні мови VBA та їх відмінність?
9. Правила формування імені змінної (ідентифікатора)?
10. Формат та призначення іменованих (або користувацьких) констант?

11. Якими можуть бути вирази в VBA?
12. Що таке стандартні функції і які правила їх запису?
13. Які Ви знаєте основні функції перетворення даних?
14. Які Ви знаєте арифметичні оператори мови VBA?
15. Формат та призначення оператора присвоєння?
16. Формат програмного рядка мови VBA?
17. Пріоритети виконання арифметичних операцій?
18. Яким чином можна відмітити програмний рядок?
19. Що таке операція конкатенації, і який оператор застосовується для її виконання?
20. Формат та призначення функції InputBox?
21. Формат та призначення функції MsgBox?
22. Призначення оператора Debug.Print та особливості розділення елементів виводу?



## ЛАБОРАТОРНА РОБОТА № 3

### РОЗРОБЛЕННЯ ТА ПРОГРАМУВАННЯ АЛГОРИТМІВ РОЗГАЛУЖЕНОЇ СТРУКТУРИ

**Мета та основні завдання роботи.** Дослідити роботу і можливості роботи рядкового і блокового операторів умовного переходу *If*. Ознайомитися з правилами запису складних умов за допомогою знаків логічних операцій.

**Вивчити та описати (стисло) в протоколі лабораторної роботи:**

- правила запису та роботу рядкового і блокового операторів умовного переходу *If*;
- правила запису складних умов за допомогою знаків логічних операцій.

### 3.1. Основні теоретичні відомості

#### Програмування розгалужених процесів на мові VBA

Більшість задач, що зустрічаються у практичній діяльності людини, не вкладаються у лінійну схему обчислень, розглянуту раніше. Як правило, у розрахунках зустрічається множина умов, від виконання яких, залежить схема розрахунку. Залежно від послідовності дій розрізняють наступні типи алгоритмів:

**Лінійний алгоритм** – це такий алгоритм у якому дії виконуються в порядку їх запису без пропусків від початку до кінця і без розгалуження.

**Розгалужений алгоритм** – це такий алгоритм у якому послідовність дій може змінюватися залежно від виконання чи невиконання деякої умови розгалуження.

Приклад:

$$y = \begin{cases} X^2 - 4 * X^{0.4}, & X > 0 \\ X^3 + 1.8 * X, & X \leq 0 \end{cases}$$

У даному випадку, в залежності від знаку змінної *X*, зміна *Y* обчислюється за різними формулами. Для зміни лінійної схеми обчислень

використовують оператори передачі управління (вибора дій) – оператори безумовного і умовного переходів.

### Оператор безумовного переходу **Goto**

Найпростіший спосіб змінити послідовність операцій в програмі – використати оператор *Goto*. Оператор безумовного переходу – це інструкція, яка просто змінює порядок (чергу) виконання операторів незалежно від якої небудь спеціальної умови. Цей оператор передає управління за вказаною міткою без будь-яких умов.

Формат:

**Goto** < мітка >

Мітка може бути числовою (номер рядка) або текстовою (алфавітно-цифрова мітка рядка). Алфавітно-цифрова мітка записується у новому програмному рядку і її написання завершується символом двокрапка (:).

Приклад:

```
Goto 100
...
Goto SV
...
...
100 a = a1 + a2
...
sv :
b = 0
...
```

У цьому фрагменті оператор **Goto 100** передає управління програмному рядку з номером 100, а оператор **Goto SV** передає управління програмному рядку, позначеному міткою **SV**.

При вирішенні питання про мітку програмного рядка багато програмістів віддають перевагу алфавітно-цифровій мітці і вважають, що використання номерів рядків в процедурах VBA є небажаним.

Мабуть, немає жодної програмної конструкції, яка б піддавалася настільки сильній критиці програмістами останнім часом, як оператор *Goto*. Використання операторів *Goto* в програмі ускладнює розуміння і реалізацію алгоритмів і робить пошук помилок практично неможливим. У програмі, що містить велику кількість операторів *Goto*, часто важко розібратися не тільки сторонньому користувачу, але і самому автору програми. Крім того, обов'язкове розташування оператора в першій колонці тексту порушує вирівнювання програмного кода.

*Visual Basic* (як, втім, і інші сучасні мови програмування), не зважаючи на підтримку оператора *Goto*, містить всі необхідні засоби, що дозволяють обійтися без нього при написанні програм. На думку програмістів фірми Microsoft, використання оператора *Goto* виправдано тільки при обробленні помилок.

### **Оператори умовного переходу If**

Оператори *If* – це найбільш вживана конструкція, що дає можливість розгалуження програми. Оператори *If* перевіряють задану умову і в залежності від результату перевірки, виконують одну з гілок програми. В *VBA* розрізняють рядкову і блокову конструкції оператора умовного переходу *If*.

#### **Рядковий оператор If**

Оператор *If* записується в одному програмному рядку і має формат.

Формат:

***If* <умова> *Then* <оператор 1> [*Else* < оператор 2 >] ,**

де <оператор 1>, <оператор 2> – оператори мови *VBA*.

В умові записується вирази (або змінні), що зв'язані знаками відношення:

=, <, >, >=, <=, <>, >< .

Оператор працює наступним чином.

Спочатку перевіряється умова. Якщо умова приймає значення *True* (Істина), то виконується ***оператор 1***, що записаний за ключовим словом

*Then*, а якщо умова приймає значення *False* (Хибність), то виконується **оператор 2**, що записаний за ключовим словом *Else*. Після виконання однієї з гілок – *Then* або *Else*, дія програми продовжується з оператора наступного, за оператором *If*.

Приклади:

```
If x >= 0 Then y = SIN(x) Else y = COS(x)
```

```
If a >= 5 Then y = SIN(a-2) Else y = 0.75*a -1
```

```
If temperature >= 100 Then MsgBox "Вода вже кипить" _  
Else MsgBox "Вода ще не кипить"
```

```
If x= 5 Then y = 1 Else y = 0
```

Як видно з формату оператора *If*, гілка *Else* є необов'язковою.

Тому крім повного оператора *If* (присутні частині *Then* і *Else*) можна використовувати **скорочену модифікацію** – слово *Else* відсутнє. Тоді, після перевірки умови, якщо вона приймає значення *True* (Істина), то виконується оператор, що записаний за ключовим словом *Then* і програма продовжується з програмного рядка, який записаний за скороченим оператором *If*, а, якщо вона приймає значення *False* (Хибність), то оператор *If* ігнорується і виконання програми продовжується з програмного рядка, який записаний за скороченим оператором *If*.

Приклад:

```
If x > 0 Then y = LOG(x)
```

```
c = a + b
```

Умова, що записана в операторі *If*, може бути складною.

Складні умови будуються з простих відношень за допомогою знаків логічних операцій:

**And** – операція логічне *I* або логічне множення (кон'юнкція). І один, і інший умовні вирази мають бути істинні, щоб весь складний вираз можна було вважати за істину (коли кожна умова істинна, то і результат також – істина).

**Or** – операція логічне *АБО* або логічне складання (диз'юнкція). Достатньо, щоб один з виразів був істинним, щоб весь складний вираз був істинним.

**Not** – операція логічне *НІ* або заперечення. Це операція з одним операндом. Якщо операнд є істинним, то весь вираз буде хибним. І навпаки.

*Та інші: Eqv – логічна еквівалентність двох виразів, Xor – виключаюча диз'юнкція – логічне заперечення двох виразів (якщо один і лише один умовний вираз має значення ІСТИНА, то результат буде ІСТИНА. Якщо обидві умови ІСТИНА або обидва ХИБНІСТЬ, то результат буде ХИБНІСТЬ), Imp – логічна імплікація двох виразів.*

Приклад:

```
If a >= 5 And a < 75 Then y=SQR(x) Else y = x^3-5
```

```
If x > 1 OR x < 5 Then MsgBox "x не належить інтервалу [1, 5]"
```

Іноді деякі програмісти заключають операції відношення в логічних виразах в дужки. І хоча це ніяким чином не впливає на виконання оператора умовного переходу IF, вони вважають, що такий запис виключає ряд помилок.

Приклад:

```
If (x > 1) OR (x < 5) Then MsgBox "x не належить інтервалу [1, 5]"
```

Порядок виконання операції у логічних виразах наступний :

1. Виконуються всі арифметичні вирази;
2. Виконуються всі операції відношення;
3. Виконуються всі операції Not;
4. Виконуються всі операції And;
5. Виконуються всі операції Or;
6. Виконуються всі операції Xor і Eqv;
7. Виконуються всі операції Imp.

**Зауваження.** Слід пам'ятати, що рядковий оператор розміщується тільки в одному програмному рядку, довжина котрого у середовищі VBA не повинна перевищувати 1024 символа. Якщо за ключовими словами *Then* і

*Else* необхідно записати декілька операторів VBA, то відділяти їх між собою, потрібно двокрапкою (:).

Приклад:

Розглянемо розробку алгоритму (правила побудови блок-схем наведені в Додатку 11) і програму для рішення наступної задачі.

Розрахувати значення  $y$  при наступних значеннях змінних:  $a=3.5$ ;  $b=14.76$ ;  $c=-2.4$ ;  $t=33.39$ , якщо значення  $x$  обчислюється за наступною формулою:

$$x = \sqrt{t} - \frac{abc}{a^2 + b^2},$$
$$y = \begin{cases} \cos^2 x^2, & x < -1 \\ 0, & x = -1, x = 1 \\ a + b^2, & -1 < x < 1 \\ e^{-x}, & x > 1 \end{cases}$$

Алгоритм задачі наведемо у вигляді блок-схеми (Додаток 12).

```
Public Sub prim1()  
'Програма розрахунку y  
Dim a As Double, b As Double  
Dim c As Double, t As Double  
Dim x As Double  
Dim y As Double  
a = Val(InputBox("Введіть значення a", "Вікно введення", "3.5"))  
b = Val(InputBox("Введіть значення b", "Вікно введення", "14.76"))  
c = Val(InputBox("Введіть значення c", "Вікно введення", "-2.4"))  
t = Val(InputBox("Введіть значення t", "Вікно введення", "33.39"))  
x = Sqr(t ^ 3) - (a * b * c) / (a ^ 2 + b ^ 2) 'Розрахунок x  
If x < -1 Then y = Cos(x ^ 2) ^ 2  
If x = -1 Or x = 1 Then y = 0  
If x > -1 And x < 1 Then y = a + b ^ 2
```

```

If x > 1 Then y = Exp(-x)
Debug.Print "При коефіцієнтах a= "; a, "b="; b, "c="; c, "t="; t
Debug.Print "Значення x і y :"; "x="; x, "y="; y
End Sub

```

Immediate			
При коефіцієнтах a= 3,5	b= 14,76	c=-2,4	t= 33,39
Значення x і y:x= 193,479854768167		y= 9,39218665133574E-85	

Рис. 3.1 – Результати виконання програми

### Блоковий оператор умовного переходу If

Основним недоліком рядкового оператора IF являється обмеженість його довжини. Часто виникає необхідність виконання великої групи операторів в залежності від виконання якої-небудь умови. Для цих цілей зручно використовувати блокову конструкцію оператора умовного переходу IF.

Формат:

```

If < умова > Then
    Перша група операторів
[ Else
    Друга група операторів]
EndIf

```

Оператор працює аналогічно рядковому If.

Якщо умова приймає значення True (Істина), то виконується перша група операторів, що записані за ключовим словом *Then*, а при хибності умови – друга група операторів, що записані за ключовим за словом *Else*. Після виконання однієї з гілок *Then* або *Else*, дія програми продовжується з оператора наступного, за оператором *EndIf* (тобто виконання будь-якої з альтернативних гілок приводить до виходу з блоку). Блокова конструкція оператора умовного переходу *If* закінчується ключовим словом *EndIf* (розташування *EndIf* вказує кінець блоку). Як видно з формату блокового оператора *If*, гілка *Else* також є необов'язковою.

Синтаксис блокового оператора *If* передбачає, що при написанні блокового *If*: групи операторів, що записані за ключовими словами *Then* і *Else*, починаються в нових програмних рядках і ключові слова *Then*, *Else* і *EndIf* також записується в нових програмних рядках.

Приклад:

```
If x >= 0 Then
y = SIN(x)
Else
y = COS(x)
EndIf
```

В групи операторів, що записані за ключовими словами *Then* і *Else* можна включити необхідну кількість нових конструкцій *If– Then – Else*. При цьому слід пам'ятати, що кожен блоковий оператор *If* повинен бути закритий оператором *EndIf*.

В межах одного блоку *If – EndIf* може бути використана будь-яка кількість конструкцій з додатковими перевірками *ElseIf*. Виконання будь-якої з альтернативних гілок приводить до виходу з блоку.

Форму такого блоку в загальному випадку умовно можна записати у вигляді:

Формат:

```
If < умова 1> Then
    1-ша група операторів
[ElseIf < умова 2> Then
    2-га група операторів]
    ...
[Else
    n- на група операторів]
EndIf
```

Ця схема діє за наступним принципом. Перевіряється *умова1*. Якщо вона істинна, то виконується *група операторів 1* і здійснюється перехід до



оператора, що стоїть за ключовим словом *Endif*. Якщо ж *умова1* хибна, то перевіряється *умова2* і так далі. Якщо перевірка дійшла до останнього оператора *Elseif* і умова теж виявляється хибною (тобто всі вищестоящі умови хибні), то виконується *група операторів n*, що записані за ключовим словом *Else*. Частина *Else* і *Elseif* (як і в рядковому операторі *If*) є необов'язковими.

Для кращого розуміння логіки програмних частин бажано використовувати елементи структурного програмування. Для цього при вирівнюванні програмного кода слід зміщувати залежні речення операторів вправо. Наприклад, при написанні блокового оператора *If* його залежні частини *If*, *Else*, *Elseif*, *EndIf* слід записувати в одних позиціях програмних рядків з використанням для цього необхідної кількості пропусків (тобто візуально буде визначено, до якого *If* відносяться частини *Else* і *EndIf*).

Приклад:

```
Public Sub prim2()  
    'Програма розрахунку  
    Dim x As Single, y As Single  
    x = Val(TextBox("Введіть значення x"))  
    y = Val(TextBox("Введіть значення y"))  
    If x > 0 Then  
        If y > 0 Then  
            Debug.Print " x > 0 "  
            Debug.Print " y > 0 "  
        Else  
            Debug.Print " тільки x > 0 "  
        End If  
    ElseIf x = 0 Then  
        If y = 0 Then  
            Debug.Print " x = 0 "  
            Debug.Print " y = 0 "
```

```

Else
    Debug.Print " тільки x = 0 "
End If

Else
    Debug.Print " x < 0 "
End If

End Sub

```

### Оператор вибору **Select Case**

Оператор **Select Case** подібний блоковому оператору *If* і в будь-якому місці програми може бути замінений ним. Основна різниця при використанні цих операторів полягає в тому, що конструкція *Select Case* приймає рішення на основі аналізу значення одного виразу (оператор **Select Case** обчислює єдиний вираз і потім в залежності від його значення виконує ті або інші блоки). Цей вираз вказується у рядку *Select Case*. Оператор вибору *Select Case* зручно використовувати, коли залежно від значення деякого виразу необхідно виконати різні дії. Формат оператора **Select Case** має вигляд :

Формат:

```

Select Case P
    Case K1
        1-ша група операторів

    Case K2
        2-га група операторів
    . . .
    Case Else
        n- на група операторів

End Select

```

де *P* – перемикаючий вираз (або змінна); *K<sub>1</sub>, K<sub>2</sub>, ...* – перемикаючі ключі: числа, умови, проміжок. Блокова конструкція оператора вибору *Select Case* –

**End Select** закінчується ключовим словом **End Select** (розташування **End Select** вказує кінець блока).

Оператор працює наступним чином.

Спочатку обчислюється значення перемикаючого виразу, що стоїть після ключових слів **Select Case**. Потім виконується перевірка того, чи задовольняє значення виразу **P** одному з ключів  $K_i$  (Case-умов) і в залежності від цього виконується відповідна група операторів (якщо значення виразу співпадає з  $K_1$ , то виконується 1-ша група операторів, якщо з  $K_2$ , то виконується 2-га група операторів і т.д.). Виконання однієї з гілок призводить до виходу з блоку. Якщо значення перемикаючого виразу **P** не співпало з жодним з ключів  $K_1, K_2, \dots, K_n$ , то виконується n- на група операторів, що записана за ключовими словами **Case Else**. Після виконання однієї з гілок дія програми продовжується з оператора наступного, за оператором **End Select**.

Змінні, за якими обчислюється значення виразу **P** повинні бути визначені до початку роботи з блоком **Select Case**.

Ключі  $K_i$ , що записані за ключовими словами **Case** можуть бути:

1. Числовими ( рядковими ) виразами.

За ключовим словом **Case** можна записати один або декілька виразів (констант, змінних), які розподіляються комами. Якщо Case-умова  $K$  записана в такій формі, то достатньо, щоб значення виразу **P** дорівнювало значенню одного з цих виразів.

Приклад:

**Case** 0

**Case** 1, 3, 5

**Case** 2, 4, 6

В прикладі перевіряється рівність значення виразу **p** наведеним числам: в першій Case-умові перевіряється рівність значення виразу **p** нулю, в другій – одному з чисел 1, 3, 5, а в третій – одному з чисел 2, 4, 6.

2. Діапазонами числових ( рядкових ) виразів.

Для завдання діапазону можна використовувати одну з двох форм запису:

### Перша форма

**Case < вираз 1 > TO < вираз 2 >**

При цьому, значення < виразу 2 > повинно бути більше за значення < виразу 1 >. При такому запису діапазону перевіряється, чи знаходиться значення виразу **p** у вказаному діапазоні.

Приклади:

**Case 1 TO 9**

**Case 2.5 TO 273.2**

**Case -7 TO -3**

В прикладі перевіряється, чи знаходиться значення виразу **p** в заданому діапазоні. В першій Case-умові перевіряється чи знаходиться значення виразу **p** в діапазоні чисел від 1 до 9, в другій – від 2.5 до 273.2, а в третій – від -7 до -3.

### Друга форма

**Case Is < знак відношення > < вираз 1 >**

В цьому випадку використовують порівняння якого-небудь вигляду (наприклад, менше – <, більше – >, тощо).

Приклади:

**Case Is < 1**

**Case Is > 9**

**Case Is < 2.1 + b^2**

В прикладі перевіряється, чи задовільняє значення виразу **p** умові, яка задає діапазон. В першій Case-умові перевіряється чи менше значення виразу **p** за 1, в другій – чи більше за 9, а в третій – чи менше значення виразу **p** за результат обчислення виразу  $(2.1 + b^2)$ .

**Зауваження:** Слід пам'ятати, що передати управління в середину блоків **IF** і **SELECT CASE** (тобто увійти в них по мітці) не можна. Передача управління можлива лише на початок блоку.

Приклад:

Програма визначає яким є число А:

1. Цілим з проміжку від 1 до 9 ( парні або непарні ).
2. Дробовим на проміжку від 1 до 9.
3. Будь яким, але меншим за 1.
4. Будь яким, але більшим за 9.

```
Public Sub prim3 ()
'Програма з SELECT CASE
Dim a As Single
a = Val(InputBox("Введіть значення x"))
SELECT CASE a
CASE 1,3,5,7,9
    Debug.Print " a – непарне ціле число з інтервалу [1,9] "
CASE 2,4,6,8
    Debug.Print " a – парне ціле число з інтервалу [1,9] "
CASE IS < 1
    Debug.Print " a < 1 "
CASE IS > 9
    Debug.Print " a > 9 "
CASE ELSE
    Debug.Print " a – дробне ціле число в інтервалі [1,9] "
END SELECT
End Sub
```

З метою більш поглибленого вивчення теоретичних основ даної лабораторної роботи рекомендується використати конспект лекцій з курсу та список рекомендованої літератури для даних методичних вказівок.

### **3.2. Опис лабораторних засобів та обладнання**

Лабораторна робота виконується на персональному комп'ютері стандарту IBM PC під керуванням операційної системи MS Windows зі стандартним пакетом MS Office.

#### **Заходи безпеки під час виконання лабораторної роботи**

Заходи безпеки, яких треба дотримуватись при виконанні лабораторної роботи «Автоматизація обчислень в MS Excel за допомогою макросів» наведено у Додатку 5.

### **3.3. Послідовність виконання роботи**

Згідно з отриманим завданням виконати наступне:

1. Вивчити та описати правила запису та роботу рядкового і блокового операторів умовного переходу IF.
2. Вивчити та описати правила запису складних умов за допомогою знаків логічних операцій.
3. У відповідності до свого варіанту завдання (див. Додаток 13) розробити розгалужений алгоритм та написати програми (з використанням рядкового і блокового операторів IF). Передбачити в програмі друк не тільки кінцевого результату, але також вихідних даних і проміжних результатів. Набрати у вікні редактора текст розробленої програми (всі змінні в програмі оголошуються явно). Включити в текст програми коментарі з інформацією про виконавця, його академічну групу, тему і номер лабораторної роботи.
4. Підібравши необхідні значення змінних, що входять в розрахункові формули перевірити роботу всіх гілок розробленого алгоритму. Записати в протокол отримані результати і значення змінних, при яких вони були отримані.

5. Програму зберегти на диску в папці своєї групи (та іншому носії). Створити документ Microsoft Word, в який скопіювати текст програми і додати отримані результати роботи програми (всі варіанти).
6. Продемонструвати роботу програми викладачу.
7. Вихідні дані, програму та отримані результати надрукувати.
8. Оформити протокол лабораторної роботи (алгоритм задачі навести у вигляді блок-схеми).

### **3.4. Оброблення та аналізування результатів. Оформлення звіту**

При оформленні звіту з лабораторної роботи до заздалегідь підготованого протоколу (див. завдання до самостійної роботи) додаються роздруковані аркуші з результатами виконаної роботи та рисунок блок-схеми алгоритму.

#### **Контрольні запитання**

1. Формат та призначення оператора безумовного переходу?
2. Формат та призначення рядкового оператор умовного переходу?
3. Відмінність в роботі рядкового оператор умовного переходу: повного та скороченого?
4. Формат та призначення блокового оператору умовного переходу?
5. Відмінність в роботі блокового оператору умовного переходу: повного та скороченого?
6. Формування складних умов в операторах умовного переходу?
7. Порядок виконання операцій у логічних виразах?
8. Формат та призначення блокового оператору умовного переходу з додатковими перевітками *ElseIf*?
9. Формат та призначення оператору вибору **Select Case**?
10. Варіанти запису ключів в операторі вибору **Select Case**?
11. Відмінність роботи блокового оператору умовного переходу **If** та оператору вибору **Select Case**?

## ЛАБОРАТОРНА РОБОТА № 4

### ОРГАНІЗАЦІЯ АРИФМЕТИЧНИХ ЦИКЛІВ

**Мета та основні завдання роботи.** Дослідити роботу і можливості оператора цикла *For... Next*. Вивчити алгоритм роботи цикла з передумовою. Ознайомитися з правилами організації вкладених циклів.

**Вивчити та описати (стисло) в протоколі лабораторної роботи:**

- правила запису та роботу оператора цикла For– Next;
- алгоритм роботи цикла з передумовою;
- вкладені цикли.

#### 4.1. Основні теоретичні відомості

##### Програмування циклічних структур на мові VBA

Часто виникає необхідність виконувати одну і туж групу операторів (частину програми) декілька разів. Запрограмувати передачу управління на ці частини програми можна, використовуючи розглянуті оператори передачі управління If, Goto, Select Case.

Проте VBA має могутні і цікаві засоби для програмування таких ситуацій – оператори циклу.

**Цикл** – це фрагмент програми, який повторюється або задане число разів, або до тих пір, поки не буде виконано деяку умову. (Чи – це процес виконання фрагменту програми деяку кількість разів).

Розрізняють цикли, де кількість повторень відома заздалегідь (визначені або арифметичні) і цикли, де вона заздалегідь невідома (невизначені або ітераційні).

При побудові циклу слід забезпечити:

- етап підготовки (тут задають початкові значення змінних, які будуть використані в циклі). Ця частина виконується один раз;



- перевірка умови цикла – тут перевіряється умова, котра забезпечує роботу цикла і його закінчення (вихід з цикла) після досягнення мети, яка покладена і основу роботи цикла;
- виконання конструкції цикла (фрагменту програми);
- зміна параметру цикла.

Щоб виділити такий фрагмент програми, необхідно вказати:

1. Початок – заголовок циклу.
2. Кінець – кінець циклу.
3. “Тіло” циклу. Програмні рядки, які розташовані між заголовком і кінцем циклу називають “тілом” циклу.

В залежності від того, коли перевіряється умова циклу розрізняють два види циклів:

- цикли з передумовою – тут умова перевіряється на початку цикла;
- цикли з післяумовою – тут умова перевіряється на кінці цикла.

Блок–схеми циклів таких циклів наведені на рис.4.1, рис.4.2.

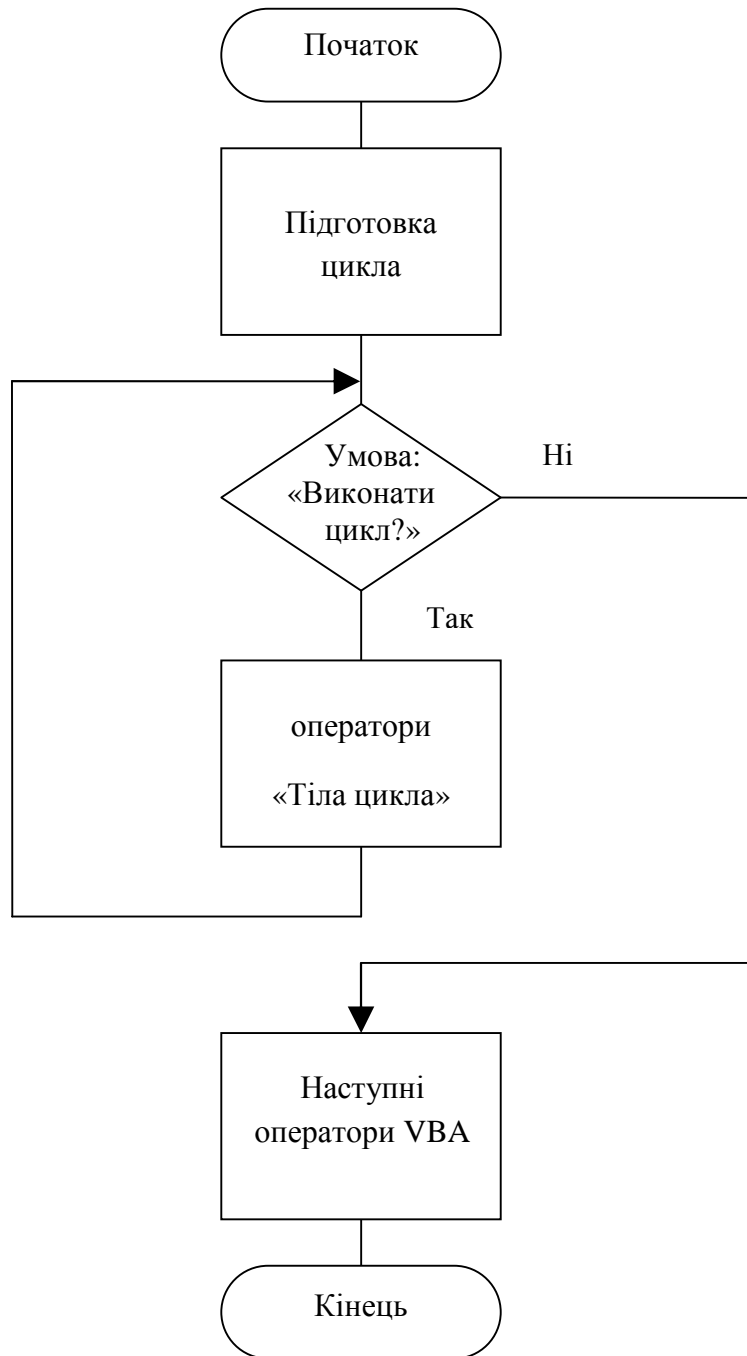


Рис.4.1 – Блок-схема алгоритму цикла, що перевіряється на початку цикла (цикл з передумовою)



Рис.4.2 – Блок-схема алгоритму циклу з умовою, що перевіряється в кінці циклу (цикл з післяумовою)

## Цикли For – Next

Це найбільше поширений варіант організації цикла. Тут “тіло” цикла повторюється задану кількість разів. При цьому параметр цикла змінюється за правилом арифметичної прогресії.

Формат:

```
For <P>=<Початкове значення> To <Кінцеве значення>[Step <Крок>]  
  «Тіло» циклу - оператори мови VBA  
[Exit For]  
Next [P]
```

Ключові (службові) слова **For–To–Step** утворюють заголовок цикла. А ключове слово **Next** – кінець цикла. Ці конструкції записують в окремих рядках.

**P** – параметр цикла ( лічильник цикла або змінна цикла) – числова змінна.

**Початкове значення** – вираз, змінна або константа, що визначає початкове значення параметра цикла.

**Кінцеве значення** – вираз, змінна або константа, що визначає кінцеве значення параметра цикла.

**Крок** – вираз, змінна або константа, що визначає величину зміни параметра цикла при повторному виконанні. Крок може бути позитивним і негативним. При позитивному значенні кроку <Кінцеве значення>, що записане в заголовку цикла повинно бути більшим за <Початкове значення >. У випадку негативного значення кроку <Кінцеве значення > повинно бути меншим за <Початкове значення>. Якщо крок дорівнює +1, то ключове слово **Step** можна не писати.

**Exit For** – інструкція для негайного завершення циклу, що забезпечує достроковий вихід з циклу. Операторів **Exit For** може бути будь-яка кількість і вони можуть розташовуватися в будь-якому місці “тіла” циклу. **Exit For** перериває виконання циклу і програма продовжується з оператора, що записаний за ключовим словом **Next**.

*Next* завершує цикл вказуючи місце кінця циклу. У *Next* бажано записувати параметр циклу, для того, щоб знати який цикл закривається. Це дуже важливо при організації вкладених циклів для уникнення плутанини.

Циклічні структури, в яких кількість повторів завчасно визначена, часто називають арифметичними циклами. Їх організація не представляє великої складності.

Блок-схема алгоритма цикла **For – Next**, що записаний в наступному вигляді:

**For** P = P<sub>поч</sub> **To** P<sub>кін</sub> **Step** ΔP

«Тіло» циклу – оператори мови VBA

**Next** P

зображена на рис.4.3.

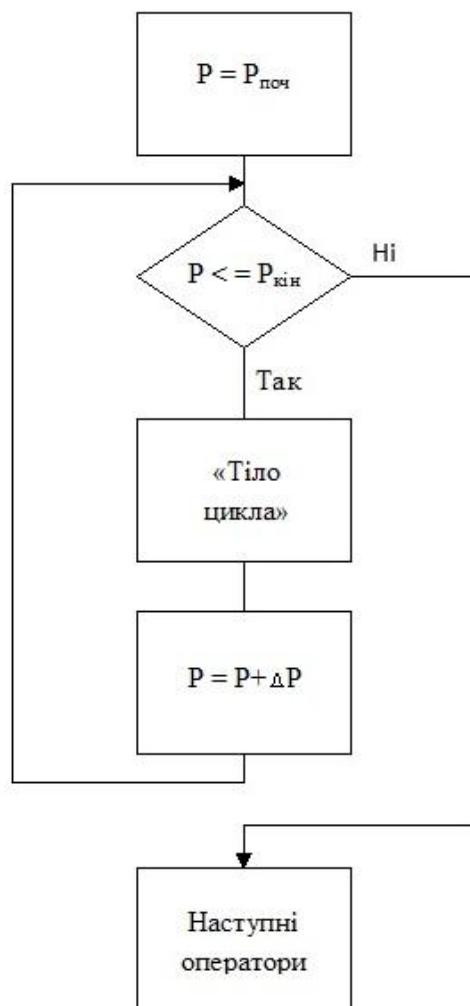


Рис.4.3 – Блок-схема алгоритма цикла For – Next

## Опис роботи циклу For – Next

Параметру циклу **P** присвоюється <Початкове значення> –  $P_{\text{поч}}$ . Далі іде перевірка умови циклу. Якщо значення параметру циклу **P** менше або дорівнює <Кінцевому значенню> –  $P_{\text{кін}}$ , то виконується «Тіло» циклу. Після виконання всіх операторів «тіла» циклу програма досягає ключового слова Next, яке вказує на кінець циклу. Далі значення параметру циклу **P** змінюється (наприклад, збільшується) на <Крок> –  $\Delta P$  ( $P = P + \Delta P$ , якщо крок додатній). Далі цикл повертається до заголовку, де знову іде перевірка умову циклу і т. д. Коли значення параметру циклу **P** стане більшим за <Кінцеве значення>, то цикл буде закінчено і виконання програми продовжується з програмного рядка, який записаний за оператором Next.

Цикли For – Next у VBA є циклами з передумовою, бо тут умова закінчення циклу перевіряється на початку циклу.

Приклад: Розрахувати  $y = \sin(x)$  в інтервалі  $[\pi/6, 2\pi]$  з кроком  $\pi/6$ .

```
Sub ccc()  
` Розрахунок в циклі  $y = \sin(x)$   
Dim x As Single  
Dim y As Single  
Dim PI As Single  
PI=3.14  
Debug.Print "Таблиця результатів"  
For x = PI/6 To 2* PI Step PI/6  
    y = SIN(X)  
    Debug.Print "При x="; x, "y="; y  
Next x  
End Sub
```

Immediate	
Таблиця результатів	
При x= 0,5233334	y= 0,4997701
При x= 1,046667	y= 0,8657598
При x= 1,57	y= 0,9999997
При x= 2,093333	y= 0,8665558
При x= 2,616667	y= 0,5011488
При x= 3,14	y= 1,592548E-03
При x= 3,663333	y=-0,49839
При x= 4,186667	y=-0,8649619
При x= 4,710001	y=-0,9999971
При x= 5,233334	y=-0,8673492
При x= 5,756668	y=-0,5025257

Рис.4.4 – Результат виконання програми

Як видно з наведених результатів розрахунку (рис.4.4), значення  $y$  при  $x=2\pi$  не було отримане. Ця особливість використання циклу *For – Next* виникає при використанні у якості параметру циклу не цілих чисел при проведенні розрахунків. Щоб її уникати, треба кінцеве значення параметру циклу, що записується у заголовку циклу *For – Next*, збільшувати на не невелике значення (наприклад, на крок).

Наприклад, в нашому випадку слід записати у програмі такий заголовок:

`For x = PI/6 To 2* PI+ PI/6 Step PI/6 , або`

`For x = PI/6 To 13* PI/6 Step PI/6`

Приклад: Надрукувати цілі числа від 1 до 100 (рис.4.5)

```
Public Sub ccc()
    'Друк чисел
    Dim x As Integer
    Debug.Print "Числа"
    For x = 1 To 100
        Debug.Print x;
    Next x
End Sub
```

Immediate	
Числа	
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 ...	97 98 99 100

Рис.4.5 – Результат виконання програми

При використанні циклу **For – Next** слід пам'ятати:

1. Передати управління в середину циклу не можна (як і в оператори **If** і **Select Case**) – це блокова конструкція. Передати управління можна тільки рядку програми, де записаний заголовок циклу.
2. Вирази, що визначають крок, кінцеве значення і початкове значення обчислюються один раз перед виконанням циклу.
3. Параметр циклу можна змінювати усередині циклу. Але це не слід робити, бо можливе виникнення таких помилок, які дуже важко буде усунути. Наприклад, такий цикл ніколи не закінчиться:

```
Dim A(1 To 5) As Integer
Dim i As Integer
For i = 1 To 5 Step 1
    A(i) = i
    i = i - 1
Next i
```

4. Вийти з циклу можна у будь-який момент (за допомогою операторів **Exit For**, **If**, **GoTo**).
5. Після закінчення циклу лічильник (параметр) циклу зберігає своє значення у момент виходу і його можна використовувати.
6. Кількість повторень циклу  $n$  можна визначити за формулою:

$$n = \frac{P_k - P_1}{\Delta P}, \quad (4.1)$$

де  $P_k$ ,  $P_1$  – кінцеве і початкове значення параметру циклу;  $\Delta P$  – крок зміни параметру циклу.

### Вкладені цикли

Всередині блокової структури програми можуть бути розміщені один чи декілька інших або таких же блоків. Аналогічним чином всередині циклу можна розмістити необхідну кількість інших циклів. Такі конструкції називаються вкладеними циклами. При цьому, одні цикли будуть називатися



зовнішніми, а інші внутрішніми. Внутрішній цикл може містити в собі інші цикли.

Правила вкладання:

- параметри зовнішніх і внутрішніх циклів повинні бути різними і не змінюватися одночасно;
- вкладені цикли не повинні перетинатися один з іншим.

Приклади:

Варіанти правильного вкладення				Неправильне вкладення
<pre> For I=...   For K=...   Next K Next I           </pre>				<pre> For I=...   For K=...   Next I Next K           </pre>

Приклад:

```

For I=1 To 3
  For K=1 To 5
    Блок операторів мови VBA
  Next K
Next I
          
```

Записана конструкція вкладених циклів працює наступним чином. Спочатку починається зовнішній цикл. Параметр зовнішнього циклу *I* отримує початкове значення ( $I=1$ ) і починає роботу внутрішній цикл. Блок операторів мови VBA виконується при всіх значеннях параметра *K* (від 1 до 5 включно) і  $I=1$ . Коли всі значення *K* перебрані і значення *K* стало більше за 5, внутрішній цикл закінчує свою роботу. Здійснюється вихід із внутрішнього цикла знову в зовнішній цикл. Коли програма доходить до ключового слова *Next I*, значення параметра змінюється –  $I=2$ . При новому

значенні параметра I знову повністю виконуються внутрішній цикл. I так далі. Таким чином, внутрішній цикл виконується при всіх значеннях параметра зовнішнього циклу I.

Приклад :

Розрахувати значення функції  $y = a \cdot x^2 + k \cdot b \cdot \sin k$  при наступних значеннях змінних:  $a = 5.8$  ;  $b = 12.37$  ;

x змінюється від 2 до 10 з кроком 2;

k змінюється від 0.5 до 1.3 з кроком 0.1.

Для програмування цієї задачі доцільно використати конструкцію вкладених циклів.

Програма.

```
Public Sub sr()  
' Табуляція функції  
Dim a As Single, b As Single  
Dim k As Single, y As Single  
Dim x As Integer  
a = Val(TextBox("Введіть a", "Вікно введення", "5.8"))  
b = Val(TextBox("Введіть b", "Вікно введення", "12.37"))  
Debug.Print "Результати розрахунків"  
For x = 2 To 10 Step 2  
    Debug.Print "При x="; x  
    For k = 0.5 To 1.4 Step 0.1  
        y = a * x ^ 2 + k * b * Sin(k)  
        Debug.Print " k="; k; Tab(20); "y="; y  
    Next k  
Next x  
End Sub
```

```

Public Sub sr()
' Табуляція функції
Dim a As Single, b As Single
Dim k As Single, y As Single
Dim x As Integer
a = Val(InputBox("Введіть a", "Вікно введення", "5.8"))
b = Val(InputBox("Введіть b", "Вікно введення", "12.37"))
Debug.Print "Результати розрахунків"
For x = 2 To 10 Step 2
    Debug.Print "При x="; x
    For k = 0.5 To 1.4 Step 0.1
        y = a * x ^ 2 + k * b * Sin(k)
        Debug.Print " k="; k; Tab(20); "y="; y
    Next k
Next x
End Sub

```

Рис.4.6 – Код програми у VBA

Результати роботи програми наведені в таблиці 4.1.

Таблиця 4.1.

Результати виконання програми у вікні Immediate

При x= 2		При x= 8	
k= 0,5	y= 26,16525	k= 0,5	y= 374,1653
k= 0,6	y= 27,39078	k= 0,6	y= 375,3908
k= 0,7	y= 28,77828	k= 0,7	y= 376,7783
k= 0,8000001	y= 30,29896	k= 0,8000001	y= 378,299
k= 0,9000001	y= 31,92078	k= 0,9000001	y= 379,9208
k= 1	y= 33,609	k= 1	y= 381,609
k= 1,1	y= 35,32666	k= 1,1	y= 383,3267
k= 1,2	y= 37,03519	k= 1,2	y= 385,0352
k= 1,3	y= 38,69498	k= 1,3	y= 386,695
При x= 4		При x= 10	
k= 0,5	y= 95,76525	k= 0,5	y= 582,9653
k= 0,6	y= 96,99078	k= 0,6	y= 584,1908
k= 0,7	y= 98,37829	k= 0,7	y= 585,5783
k= 0,8000001	y= 99,89896	k= 0,8000001	y= 587,099
k= 0,9000001	y= 101,5208	k= 0,9000001	y= 588,7208
k= 1	y= 103,209	k= 1	y= 590,409
k= 1,1	y= 104,9267	k= 1,1	y= 592,1267
k= 1,2	y= 106,6352	k= 1,2	y= 593,8352
k= 1,3	y= 108,295	k= 1,3	y= 595,495
При x= 6			
k= 0,5	y= 211,7653		

k= 0,6	y= 212,9908	
k= 0,7	y= 214,3783	
k= 0,8000001	y= 215,899	
k= 0,9000001	y= 217,5208	
k= 1	y= 219,209	
k= 1,1	y= 220,9267	
k= 1,2	y= 222,6352	
k= 1,3	y= 224,295	

Як видно з наведених результатів розрахунку, в заголовку циклу For – Next по змінній k записано:

```
For k = 0.5 To 1.4 Step 0.1,
```

хоча за умовою задачі k змінюється від 0.5 до 1.3 з кроком 0.1.

Цей прийом використано для того (розглядали при вивченні циклу For – Next), щоб останнє значення змінної k = 1.3 було використане у розрахунках у (параметр циклу не ціле число і тому він збільшений на крок).

З метою більш поглибленого вивчення теоретичних основ даної лабораторної роботи рекомендується використати конспект лекцій з курсу та список рекомендованої літератури для даних методичних вказівок.

#### **4.2.Опис лабораторних засобів та обладнання**

Лабораторна робота виконується на персональному комп'ютері стандарту IBM PC під керуванням операційної системи MS Windows зі стандартним пакетом MS Office.

#### **Заходи безпеки під час виконання лабораторної роботи**

Заходи безпеки, яких треба дотримуватись при виконанні лабораторної роботи «Автоматизація обчислень в MS Excel за допомогою макросів» наведено у Додатку 5.

### **4.3. Послідовність виконання роботи**

Згідно із отриманим завданням виконати наступне:

#### ***Частина 1***

1. Вивчити правила запису та роботи оператора цикла For... Next.
2. Вивчити алгоритм роботи цикла з передумовою.
3. У відповідності до свого варіанту (Додаток 14) завдання розробити циклічний алгоритм та програму розрахунку заданої функції для вказаних значень аргумента. Виведення результатів організувати у вигляді таблиці.
4. Набрати, відредагувати та налаштувати програму в середовищі VBA.
5. Програми зберегти на диску в папці своєї групи (та іншому носії). Створити документ Microsoft Word, в який скопіювати текст програм і додати отримані результати роботи програм.
6. Продемонструвати роботу програми викладачу.
7. Оформити протокол лабораторної роботи. Алгоритм задачі навести у вигляді блок-схеми.

#### ***Частина 2***

##### ***“ Вкладені цикли. Табулювання функцій ”***

1. Вивчити організацію циклічних алгоритмів, правила запису та роботи вкладених циклів.
2. У відповідності до свого варіанту (Додаток 15) завдання розробити циклічний алгоритм та програму розрахунку заданої функції для вказаних значень аргументів. Виведення результатів організувати у вигляді таблиці.
3. Набрати, відредагувати та налаштувати програму в середовищі VBA.

4. Внести необхідні корективи в розроблену програму для організації введення даних з листа Microsoft Excel і виведення результатів в інші комірки того ж листа Microsoft Excel (див. Додаток 16).
5. Програми зберегти на диску в папці своєї групи (та іншому носії). Створити документ Microsoft Word, в який скопіювати текст програм і додати отримані результати роботи програм.
6. Алгоритм задачі навести в протоколі у вигляді блок-схеми.
7. Продемонструвати роботу програм викладачу.

#### **4.4.Оброблення та аналізування результатів. Оформлення звіту**

При оформленні звіту з лабораторної роботи до заздалегідь підготованого протоколу (див. завдання до самостійної роботи) додаються роздруковані аркуші з результатами виконаної роботи та рисунки блок-схем алгоритму.

#### **Контрольні запитання**

1. Що таке цикл і які цикли використовуються у VBA?
2. З яких фрагментів складається цикл?
3. Формат і робота оператора циклу For– Next?
4. Як виконується робота циклу з передумовою?
5. Як достроково вийти з циклу For– Next?
6. Як визначити кількість повторень циклу For– Next?
7. Яке значення має лічильник (параметр) циклу після закінчення циклу For– Next?
8. Що таке вкладені цикли і які правила вкладання циклів?
9. Як виконуються вкладені цикли?

### Список рекомендованой літератури

1. Харрис М. Освой самостоятельно программирование MS Excel 2000 для за 21 день. – М.: Вильямс (SAMS), 2000. – 880 с.
2. Орвис В. Excel для ученых, инженеров и студентов. – К.: Юниор, 1999. – 528 с.
3. Гарнаев А. Самоучитель VBA. Технология создания пользовательских приложений. – СПб.: BHV, 1999. – 512 с.
4. Microsoft Visual Basic 6.0 для профессионалов. Шаг за шагом: Практик. пособие./Пер. с англ. – М.: Издательство ЭКОМ, 2001. – 720 с.
5. Уокенбах, Джон Профессиональное программирование на VBA в Excel 2002.: Пер.сангл. – М.:Издательский дом «Вильямс» , 2003. –784с.
6. Уокенбах, Джон Профессиональное программирование на VBA в Excel 2003.: Пер.сангл. – М.:Издательский дом «Вильямс» , 2005. –800с.
7. Культин Н. Б. Visual Basic. Освой на примерах. – СПб.: БХВ-Петербург, 2004. – 288 с.
8. VisualBasic.NET: учебный курс / В.Долженков, М.Мозговой. – СПб.: Питер, 2003. –464 с.
9. John Walkenbach, Excel 2010 Power Programming with VBA.–Published by Wiley Publishing, Inc., Indianapolis, Indiana Published simultaneously in Canada, 2010. – p. 1007.

### Додаток 1. Приклад тексту макросу

Програма запису макросу записує всі дії мовою VB. Цей запис називається текстом макросу або вихідним текстом. При запуску макросу на виконання програма VBA зчитує ці команди й виконує їх у порядку їхнього запису.

Приклад.

```
Sub form()  
,  
' form Макрос  
' Макрос записан 14.10.2007 (1)  
,  
    With Selection.Font  
        .Name = "Times New Roman"  
        .Size = 10  
        .Strikethrough = False  
        .Superscript = False  
        .Subscript = False  
        .OutlineFont = False  
        .Shadow = False  
        .Underline = xlUnderlineStyleNone  
        .ColorIndex = xlAutomatic  
    End With  
    Selection.Font.Bold = True  
    Selection.Font.Italic = True  
    Selection.Font.Underline = xlUnderlineStyleSingle  
    With Selection.Font  
        .Name = "Times New Roman"  
        .Size = 14  
        .Strikethrough = False  
        .Superscript = False  
        .Subscript = False  
        .OutlineFont = False  
        .Shadow = False  
        .Underline = xlUnderlineStyleSingle  
        .ColorIndex = xlAutomatic  
    End With  
End Sub
```

Кожний макрос являє собою підпрограму (процедуру), що починається зі службового слова SUB і закінчується службовим словом END SUB.



## Додаток 2. Команди меню редактора Visual Basic

**Меню File (Файл).** У меню File, як і у всіх додатках для Windows, зібрані команди, що пов'язані з відкриттям та збереженням файлів. У меню File редактора Visual Basic знаходяться команди, які дозволяють відкривати макроси, зберігати зміни і друкувати текст макросів. У табл. 2.1 наведено короткий опис команд, їх призначення та відповідні їм комбінації клавіш.

**Таблиця 2.1. Команди меню File**

Команда	Комбінація клавіш	Дія
Save (Зберегти)	<Ctrl+S>	Зберігає на диску поточний проект, включаючи всі модулі і форми
Import File (Імпорт файла)	<Ctrl+M>	Додає в існуючий модуль, форму або клас до поточного проекту. Додавати можна тільки ті модулі, форми чи класи, які були раніше збережені командою Export File з іншого проекту
Export File (Експорт файла)	<Ctrl+F>	Зберігає модуль, форму або клас з поточного проекту в текстовому форматі для подальшого імпортування або для архівування
Remove (Видалити)		Назавжди видаляє з поточного проекту зазначений модуль або форму. Ця команда не доступна, якщо жоден елемент не виділено
Print (Друк)	<Ctrl+P>	Друкує модуль або форму для звіту та архівування
Close and Return (Закрити повернутися)	<Alt+Q>	Закриває редактор Visual Basic і повертає до програми, з якої він був викликаний

Меню **Edit (Правка)**. У меню Edit зібрані команди, призначені для редагування тексту макросів у вікні Code. У табл. 2.2 зібрані команди меню Edit і наведено їх опис і вказані відповідні комбінації клавіш.

**Таблиця 2.2. Команди меню Edit**

Команда	Комбінація клавіш	Дія
<i>1</i>	<i>2</i>	<i>3</i>
Undo (Відмінити)	<Ctrl+Z>	Скасувати останню команду. (Скасувати можна не всяку команду).
Redo (Повторити)		Повторює останню виконану команду
Cut (Вирізати)	<Ctrl+X>	Вирізає виділений текст або об'єкт і поміщає його в буфер обміну. Виділений текст видаляється з модуля або форми
Copy (Копіювати)	<Ctrl+C>	Копіює виділений текст або об'єкт і поміщає його в буфер обміну. Виділений текст залишається в модулі або формі без змін
Paste (Вставити)	<Ctrl+V>	Вставляє текст або об'єкт з буфера обміну в поточний модуль або форму
Clear (Очистити)	<Ctrl+Del>	Видаляє виділений текст або об'єкт з модуля або форми
Select All (Виділити усе)	<Ctrl+A>	Виділяє весь текст у модулі чи об'єкти у формі. Виділений текст залишається в модулі або формі без змін
Find (Знайти)	<Ctrl+F>	Подібно команді Find у програмі Excel, дозволяє знайти потрібний фрагмент у тексті
Find Next (Знайти наступний)	<F3>	Повторює останню операцію пошуку
Replace (Замінити)	<Ctrl+H>	Подібно команді Replace у програмі Excel, дозволяє знайти потрібний фрагмент у тексті і замінити його новим
Indent (Збільшити відступ)	<Tab>	Збільшує відступ виділеного тексту на одну позицію табуляції

<i>1</i>	<i>2</i>	<i>3</i>
Outdent (Зменшити відступ)	<Shift+Tab>	Зменшує відступ виділеного тексту на одну позицію табуляції
List (Список властивостей/методов)	<Ctrl+J>	Розкриває список Properties / Methods у вікні Code, відображаючи властивості і методи об'єкта, ім'я якого ввели. Якщо курсор знаходиться на порожньому місці у вікні Code, виводиться список всіх властивостей і методів
List Constants (Список констант)	<Ctrl+Shift+J>	Розкриває список всіх можливих значень властивості, ім'я якого ввели перед знаком рівності
Quick Info (Відомості)	<Ctrl+I>	Виводить вікно підказки, що містить правильний синтаксис процедури, функції або методу, ім'я якого ввели у вікні Code
Parameter Info (Параметри)	<Ctrl+Z>	Виводить вікно підказки, що містить список параметрів (аргументів) процедури, функції або методу, ім'я якого ввели у вікні Code
Complete Word (Завершити слово)	<Ctrl+проміжок>	Завершує не до кінця набране ключове слово, якщо ввели досить символів для того, щоб Visual Basic зміг розпізнати це слово
Bookmarks (Закладки)		Розкриває додаткове меню, дозволяючі вставити, видалити або перейти на закладку, раніше вставлену у модулі. Закладки в Visual Basic не мають імен

Меню View (**Вид**). У меню View зібрані команди, що дозволяють вибирати, який об'єкт або елемент редактора Visual Basic треба подивитися, а також режими перегляду. У табл. 2.3 зібрані команди меню View, наведено їх опис і вказані відповідні комбінації клавіш.

**Таблиця 2.3 Команди меню View**

Команда	Комбінація клавіш	Дія
<i>1</i>	<i>2</i>	<i>3</i>
Code (Програма)	<F7>	Активізує вікно Code, що містить вихідний текст виділеного модуля або форми
Object (Об'єкт)	<Shift+F7>	Відображає об'єкт, виділений у вікні Project
Definition (Опис)	<Shift+F2>	Відображає текст процедури або функції, на імені якої стоїть курсор
Last Position (Повернення до останньої позиції)	<Ctrl+Shift+F2>	Повертається в останню позицію в модулі після використання команди Definition або після редагування тексту
Object Browser (Перегляд об'єктів)	<F2>	Розкриває вікно Object Browser, в якому можна бачити всі доступні в даний момент макроси.
Immediate Window (Вікно налагодження)	< Ctrl+G>	Розкриває вікно відладчика редактора Visual Basic
Locals Window (Вікно локальних змінних)		Розкриває вікно локальних змінних редактора Visual Basic
Watch Window (Вікно контрольних значень)		Розкриває вікно для перегляду значень змінних
Call Stack (Стек виклику)	<Ctrl+L>	Відображає стек викликів даної функції або процедури
Project Explorer (Вікно проекту)	<Ctrl+R>	Розкриває вікно Project
Properties Window (Вікно властивостей)	<F4>	Розкриває вікно Properties
Toolbox (Панель елементів)		Розкриває панель Toolbox. Панель Toolbox застосовується для створення керуючих елементів у діалогових вікнах.
Tab Order (Послідовність переходу)		Розкриває діалогове вікно Tab Order. Діалогове вікно Tab Order застосовується для створення керуючих елементів у діалогових вікнах.

<i>1</i>	<i>2</i>	<i>3</i>
Toolbars (Панелі інструментів)		Розкриває додаткове меню, за допомогою якого можна забирати з екрана або виводити на екран різні панелі інструментів або розкрити діалогове вікно Customize (Налаштування) для налаштування панелей інструментів
< Початковий додаток >	<Alt+F11>	Повертає в додаток, з якого викликали редактор Visual Basic, сам редактор при цьому залишається відкритим. Назва цього пункту меню залежить від того, з якого додатку викликали редактор Visual Basic

**Меню Insert (Вставка).** Команди меню Insert призначені для вставки різних об'єктів у ваш проект, таких як модулі або форми. У табл. 2.4 зібрані команди меню Insert і наведено їх опис. Командам цього меню не відповідають жодні комбінації клавіш.

Таблиця 2.4 Команди меню Insert

Команда	Дія
Procedure (Процедура)	Вставляє процедуру Sub, Function або Property в поточний модуль. (Процедура – це те ж саме, що й макрос.)
UserForm	Додає нову форму в проект. (Форми застосовуються для створення діалогових вікон.)
Module (Модуль)	Додає новий модуль у проект. Редактор Visual Basic дає цьому модулю ім'я відповідно до правил, обумовлених вище
Class Module (Модуль класу)	Додає новий модуль класу в проект. Модуль класу застосовується для створення нових об'єктів
File	Дозволяє вставити в модуль частину вихідного тексту з текстового файлу

**Меню Format (Формат).** Команди меню Format використовуються, коли створюють власні діалогові вікна та інші форми. Ці команди застосовуються для того, щоб вирівнювати об'єкти щодо один до одного, міняти розміри керуючих елементів і для багато чого іншого. В табл. 2.5 зібрані команди меню Format та приведено їх опис. Командам цього меню не відповідають ніякі комбінації клавіш.

**Таблиця 2.5. Команди меню Format**

Команда	Дія
<i>1</i>	<i>2</i>
Align (Вирівняти)	Розкриває додаткове меню, що містить команди для вирівнювання виділених об'єктів на формі один відносно одного. Можна вирівняти об'єкти по правому краю, по лівому, по верхньому або нижньому або по центру вибраного об'єкту
Make Same Size (Вирівняти розмір)	Розкриває додаткове меню, що містить команди для надання виділеним об'єктам потрібного розміру
Size to Fit (Підігнати розмір)	Автоматично підбирає висоту і ширину виділених об'єктів відповідно до розміру їх написів
Size to Grid (Вирівняти розмір за сіткою)	Автоматично підбирає висоту і ширину виділених об'єктів відповідно до розміру осередку. (При створенні форм редактор Visual Basic відображає сітку на формі, в осередках де розташовуються управляючі елементи.)
Horizontal Spacing (Інтервал по горизонталі)	Розкриває додаткове меню, що містить команди для підбору горизонтального інтервалу між виділеними елементами. Можна зробити горизонтальний інтервал рівномірним, збільшити його, зменшити або прибрати його зовсім
Vertical Spacing (Інтервал по вертикалі)	Розкриває додаткове меню, що містить команди для підбору вертикального інтервалу між виділеними елементами. Можна зробити вертикальний інтервал рівномірним, збільшити його, зменшити або відмовитися від нього зовсім
Center in Form (Розташувати в центрі форми)	Розкриває додаткове меню, що містить команди для центрування виділених об'єктів на формі по горизонталі або вертикалі правого краю форми
Arrange Buttons (Розташувати кнопки)	Розкриває додаткове меню, що містить команди для рівномірного вирівнювання кнопок на формі
Group (Згрупувати)	Об'єднує виділені об'єкти в групу для того, щоб їх можна було спільно переміщати, змінювати їх розміри, копіювати або видаляти як єдиний об'єкт

<i>1</i>	<i>2</i>
Ungroup (Розділити)	Роз'єднує групу, створену попередньою командою, на незалежні об'єкти
Order (Порядок)	Розкриває додаткове меню, що містить команди для зміни порядку, що перекривають об'єкти (цей порядок називають Z-порядком). За допомогою цих команд можна зробити, наприклад, так, щоб кнопка завжди відображалася зверху деякого графічного об'єкта

**Меню Debug (Налагодження).** Команди меню Debug використовуються для перевірки і налагодження макросів. (Налагодженням називається процес пошуку і виправлення помилок в програмі.) Команди меню Debug дозволяють строго контролювати виконання програми, зупиняючи її в зазначених точках, виконуючи програму у покроковому режимі, стежачи за значеннями вибраних змінних. У табл. 2.6 зібрані команди меню Debug, наведено їх опис і вказані відповідні комбінації клавіш.

**Таблиця 2.6. Команди меню Debug**

Команда	Комбінація клавіш	Дія
<i>1</i>	<i>2</i>	<i>3</i>
Compile (Компілювати)		Компілює проект, виділений у вікні Project
Step Into (Шаг макроса із заходом)	<F8>	Виконує одну інструкцію макроса із заходом в процедуру, якщо така зустрінеться
Step Over (Шаг з обходом)	<Shift+F8>	Виконує одну інструкцію макросу без заходу в процедуру
Step Out (Шаг з виходом)	<Ctrl+Shift+F8>	Виходить з процедури в місце її виклику
Run to Cursor (Виконати до поточної позиції)	<Ctrl+F8>	Виконує текст макросу від поточної позиції до курсору
Add Watch (Додати контрольне значення)		Дозволяє вказати змінну або вираз, значення яких треба контролювати
Edit Watch (Змінити контрольне значення)	<Ctrl+W>	Дозволяє змінити вираз, значення якого треба контролювати
Quick Watch (Контрольне значення)	<Shift+F9>	Показує значення виділеного виразу

<i>1</i>	<i>2</i>	<i>3</i>
Toggle Breakpoint (Точка зупину)	<F9>	Зазначає (або знімає зазначене) місце у тексті, в якому потрібно призупинити виконання програми (таке місце називають точкою зупинки)
Clear All Breakpoints (Зняти всі точки зупину)	<Ctrl+Shift+F9>	Знімає всі точки зупинки у модулі
Set Next Statement (Задати наступну інструкцію)	<Ctrl+F9>	Дозволяє змінити порядок виконання тексту програми, вказавши наступну інструкцію для виконання
Show Next Statement (Показати наступну інструкцію)		Підсвічує (виділяє кольором) інструкцію, яка буде виконана при наступному кроці

**Меню Run (Запуск).** Команди меню Run слугують для запуску макросу на виконання, для переривання і відновлення роботи макросу, для відновлення початкових умов перед повторним виконанням. У табл. 2.7 зібрані команди меню Run, наведено їх опис і вказані відповідні комбінації клавіш.

**Таблиця 2.7 Команди меню Run**

Команда	Комбінація клавіш	Дія
Run Sub/User Form (Запуск підпрограми/User Form)	<F5>	Редактор Visual Basic запускає макрос, в тексті якого знаходиться текстовий курсор. Якщо активна форма, то Visual Basic запускає форму
Break (Прервати)	<Ctrl+Break>	Переривається виконання макросу, і редактор Visual Basic переходить в режим зупинки (Break mode).
Reset (Скинути)		Скидає значення всіх змінних модуля і очищає стек викликів.
Design Mode (Конструктор)		Включає або відключає режим конструктора для всього проекту. У режимі конструктора не виконуються ніякі макроси і не обробляються ніякі системні події



**Меню Tools (Сервіс).** Команди меню Tools дозволяють вибрати макрос для виконання і, крім того, отримати доступ до зовнішніх бібліотек макросів і форм. На додаток до цього, з меню Tools можна викликати діалогове вікно Options для налаштування параметрів проекту, виділеного у вікні Project. У табл. 2.8 зібрані команди меню Tools і наведено їх опис та відповідні комбінації клавіш.

**Таблиця 2.8. Команди меню Tools**

Команда	Дія
References (Посилання)	Розкриває діалогове вікно References, в якому можна встановити посилання на бібліотеки об'єктів, бібліотеки типів і на інший VBA-проект. Після встановлення посилань об'єкти, методи, властивості, процедури і функції стають доступними у вікні Object Browser
Additional Controls (Додаткові елементи)	Розкриває діалогове вікно Additional Controls, за допомогою якого можна додати на панель інструментів додаткові кнопки, призначені для включення в форму нових керуючих елементів. За допомогою таких же кнопок можна у форму вставити об'єкти, такі як лист Excel або документ Word
Macros (Макроси)	Розкриває діалогове вікно Macro, за допомогою якого можна створювати, редагувати, запускати або видаляти макроси
Options (Параметри)	Розкриває діалогове вікно Options, за допомогою якого можна встановлювати різні параметри редактора Visual Basic, такі як ширина інтервалу табуляції, перевірка синтаксису інструкцій, розміри осередку у формі і інші.
Properties (Властивості)	Розкриває діалогове вікно Object Properties, за допомогою якого можна встановлювати різні властивості проекту, такі як його ім'я, опис, файл довідки і т.д. У цьому ж вікні можна встановити захист свого проекту за допомогою пароля, заборонивши його редагування
Digital Signature (Цифровий підпис)	Розкриває діалогове вікно Digital Signature, за допомогою якого можна подивитися цифровий підпис проекту або встановити його

**Примітка 1.** Список додаткових керівних елементів, які можна встановити за допомогою діалогового вікна Additional Controls, залежить від того, які програми встановлені на комп'ютері. Звичайно в цьому списку вам доступні керуючі елементи ActiveX, отримані від незалежних виробників або входять до складу будь-якої програми.

**Примітка 2.** Цифровий підпис і сертифікат – це нові засоби, включені до Microsoft Office, за допомогою яких розробник може захистити свій програмний продукт, а користувач – перевірити справжність продукту і запускати на виконання тільки програми і макроси, отримані з надійних джерел.

### **Інші пункти меню редактора Visual Basic.**

Є ще три пункти меню в редакторі Visual Basic:

- Add-Ins (Надбудови);
- Window (Вікно);
- Help (Довідка).

У меню **Add-Ins** є тільки один пункт – Add-In Manager. Ця команда розкриває діалогове вікно Add-In Manager. За допомогою цього вікна можна додавати або видаляти програми-надбудови редактора Visual Basic. Подібно програмі Excel, редактор Visual Basic у складі Microsoft Office дозволяє використовувати програми-надбудови, що розширюють і поліпшують властивості Visual Basic. Такі програми-надбудови можна отримати у незалежних розробників програмного забезпечення.

Меню **Window** і **Help** абсолютно аналогічні відповідним меню програм Windows. Команди меню **Window** дозволяють вибрати активне вікно, розділити поточне вікно на частини, розташувати дочірні вікна горизонтально, вертикально чи каскадом або розташувати значки згорнутих вікон уздовж границі головного вікна. Меню Window, працює абсолютно так само, як у програмах Word, Excel і в інших додатках Windows. Меню **Help** також мало чим відрізняється від аналогічного меню програм Word, Excel та інших. Працюючи з редактором Visual Basic, можна скористатися помічником Office або просто отримати довідку по роботі з Visual Basic для того додатка, з якого його викликали. При підключення до Internet, можна через меню Help-MSDN on the Web отримати доступ до багатьох Web-сторінок, які містять важливу та корисну інформацію про роботу з продуктами Microsoft, в тому числі і про програмування на VBA. Остання команда в меню довідки – це About Microsoft Visual Basic (Про програму). За цією командою розкривається діалогове вікно, що містить інформацію про авторські права на програму Visual Basic. Крім того, в цьому вікні є кнопка з написом System Info (Про систему). Натиснувши на неї, розкриється вікно, що містить інформацію про комп'ютер, версію операційної системи, тип відеосистеми, підтримку звуку, підключені принтери, встановлені програми і інше.

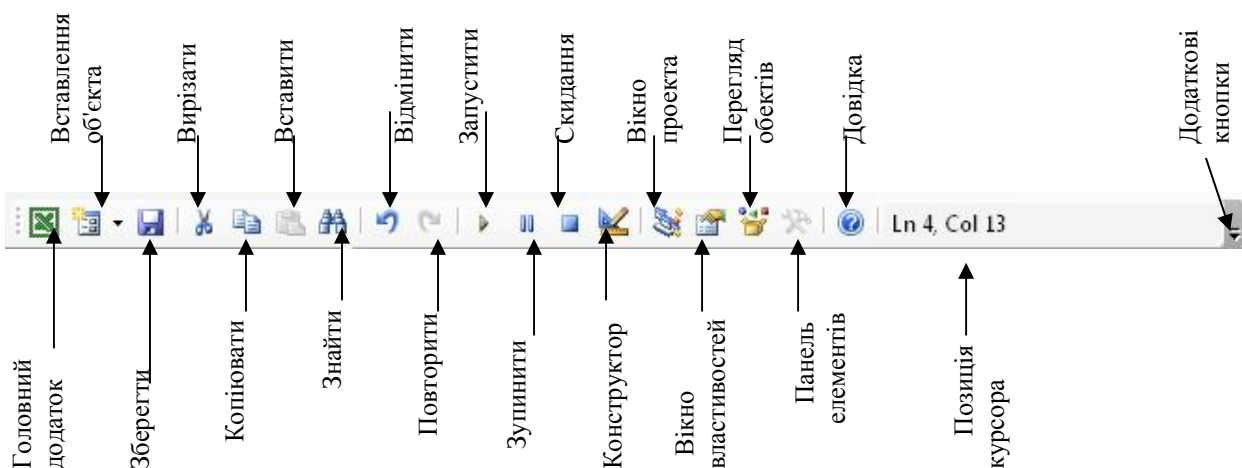
### Додаток 3. Панелі інструментів редактора Visual Basic

Більшості користувачів зручніше використовувати кнопки панелі інструментів, ніж обирати команди з меню. В редакторі Visual Basic для найбільш часто використовуваних команд меню передбачені кнопки на панелі інструментів, які прискорюють роботу.

Зазвичай редактор Visual Basic відображає стандартну панель інструментів притиснутою до верхньої межі вікна. Але можливо розташувати потрібну панель інструментів в будь-якому місці вікна. Таку панель інструментів називають плаваючою. Вона виглядає як повноцінне вікно, має межі і рядок заголовка. У редакторі Visual Basic з плаваючими панелями поводяться точно так само, як у програмі Excel або в будь-який інший – її можна перетягувати, міняти розміри і форму.

#### Стандартна панель інструментів

На стандартній панелі інструментів у редакторі Visual Basic є 18 кнопок. Кожній кнопці відповідає команда меню.





У табл. 3.1 наведено опис команд, пов'язаних з кнопками стандартної панелі інструментів. команди

**Таблиця 3.1. Кнопки стандартної панелі інструментів**

Кнопка	Дія
<i>1</i>	<i>2</i>
View (Вид) <додаток>	Перемикає на додаток, з якого був викликаний редактор Visual Basic. Зображення цієї кнопки залежить від того, з якого додатку був запущений Visual Basic. На рисунку панелі Visual Basic був викликаний з програми Excel
Insert (Вставити) <об'єкт>	Якщо клацнути на стрілці праворуч від кнопки, то буде виведено список об'єктів, які можна вставити у проект. Це об'єкти UserForm, Module, Class Module (Модуль класу) або Procedure. Діє аналогічно меню Insert
Save (Зберегти)	Зберігає поточний проект. Діє аналогічно меню File–Save
Cut (Вирізати)	Вирізає виділений текст і поміщає його в буфер обміну. Діє аналогічно меню Edit–Cut
Copy (Копіювати)	Копіює виділений текст і поміщає його в буфер обміну. Діє аналогічно меню Edit–Copy
Paste (Вставити)	Вставляє текст з буфера обміну в позицію курсору. Діє аналогічно меню Edit–Paste
Find (Знайти)	Розкриває діалогове вікно для пошуку тексту. Діє аналогічно меню Edit–Find
Undo(Відмінити)	Скасує останню команду. Скасувати можна не всяку команду. Діє аналогічно меню Edit–Undo
Redo(Повторити)	Повторює останню виконану команду. Діє аналогічно меню Edit–Redo
Run (Запуск підпрограми)	Запускає макрос, в тексті якого знаходиться текстовий курсор. Діє аналогічно меню Run–Sub/UserForm
Break (Переривання)	Переривається виконання макросу. Діє аналогічно меню Run–Break
Reset (Скидання)	Скидає значення всіх змінних модуля та оштщає стек викликів. Діє аналогічно меню Run– Reset
Design Mode (Конструктор)	Включає або відключає режим конструктора. Діє аналогічно меню Run–Design Mode
Project Explorer (Вікно проекту)	Розкриває вікно Project. Діє аналогічно меню View–Project Explorer
Properties Window (Вікно властивостей)	Розкриває вікно Properties. Діє аналогічно меню View–Properties Window
Object Browser (Перегляд об'єктів)	Розкриває вікно Object Browser. Діє аналогічно меню View–Object Browser
Toolbox (Панель елементів)	Розкриває панель інструментів Toolbox. Діє аналогічно меню View–Toolbox

<i>1</i>	<i>2</i>
Office Assistant (Помічник по Office)	Викликає вікно довідки. Діє аналогічно меню Help–Microsoft Visual Basic Help
Cursor Position (Позиція курсора)	Ця область на панелі не є командною кнопкою. Тут просто вказується позиція курсору в текстовому вікні Code, а саме номер рядка та стовпця.
More Buttons (Дополнительные кнопки)	Клацнувши на стрілці праворуч від кнопки, можна вивести додаткове меню з командами, для яких можна створити кнопки на панелі інструментів

Вигляд панелей інструментів відладки – **Debug** і форм користувача – **UserForm**

Панель інструментів <b>UserForm</b>	Панель інструментів <b>Debug</b>
	

### Панель інструментів Edit (Правка)

Панель інструментів Edit значно економить час і сили, створюючи доступ до часто використовуваних команд. Редактор Visual Basic не показує цю панель автоматично, її можна вивести її на екран при необхідності.

У табл. 3.2. наведено опис команд, пов'язаних з кнопками панелі інструментів Edit.

**Таблиця 3.2. Команди панелі інструментів Edit**

Кнопка	Дія
<i>1</i>	<i>2</i>
List Properties/Methods (Список властивостей/методів)	Розкриває список Properties / Methods. Діє аналогічно меню Edit–List Properties / Methods
List Constants (Список констант)	Розкриває список всіх можливих значень властивостей. Діє аналогічно меню Edit–List Constants

<i>1</i>	<i>2</i>
Quick Info (Відомості)	Виводить вікно підказки, що містить правильний синтаксис процедури. Діє аналогічно меню Edit–Quick Info
Parameter Info (Параметри)	Виводить список параметрів (аргументів) процедури, функції або методу. Діє аналогічно меню Edit–Parameter Info
Complete Word (Завершити слово)	Завершує не до кінця набране ключове слово. Діє аналогічно меню Edit–Complete Word
Indent (Збільшити відступ)	Збільшує відступ виділеного тексту на одну позицію табуляції. Діє аналогічно меню Edit–Indent
Outdent (Зменшити відступ)	Зменшує відступ виділеного тексту на одну позицію табуляції. Діє аналогічно меню Edit–Outdent
Toggle Breakpoint (Точка зупинки)	Зазначає точку зупинку у тексті. Діє аналогічно меню Debug–Toggle Breakpoint
Comment Block (Закоментувати блок)	Перетворює виділений текст у вікні Code в коментарі, додаючи на початку кожного рядка знак коментаря. Аналогічною команди немає в жодному іншому меню.
Uncomment Block (Розкоментувати блок)	Видаляє знак коментарю на початку кожного рядка виділеного тексту. Аналогічної команди немає ні в якому меню.
Toggle Bookmark (Закладка)	Встановлює або знімає закладку в тексті макросу. Діє аналогічно меню Edit–Bookmarks–Toggle Bookmark
Next Bookmark (Наступна закладка)	Переміщує курсор у вікні Code до наступної закладки. Діє аналогічно меню Edit–Bookmarks–Next Bookmark
Previous Bookmark (Попередня закладка)	Переміщує курсор у вікні Code до попередньої закладки. Діє аналогічно меню Edit–Bookmarks–Previous Bookmark
Clear All Bookmarks (Зняти всі закладки)	Видаляє всі закладки в модулі. Діє аналогічно меню Edit–Bookmarks–Clear All Bookmarks

## Додаток 4.Області видимості Private і Public

Змінні й константи локальної видимості, тобто описані на рівні процедури або функції, доступні лише в цій програмній одиниці. Змінні й константи, описані на рівні модуля, доступні всім його процедурам і функціям.

Хоча змінні й константи, описані на рівні модуля, доступні всім його процедурам і функціям, вони недоступні процедурам і функціям іншого модуля. Їхня область видимості обмежена модулем, у якому вони були визначені.

В VBA про змінні і константи, чия область видимості обмежена одним модулем, говорять, що вони мають область видимості **private**, тому що вони використовуються тільки усередині модуля з інструкціями, у яких вони були описані. Всі змінні й константи, описані на рівні модуля в розділі описів (за допомогою інструкції Dim), мають область видимості private, а це означає, що вони обмежені цим модулем.

В VBA те, що доступно всім модулям всіх проектів, має область видимості **public** (загальний), тому що загальнодоступні всім цим модулям. Всі процедури, що перебувають у модулі, функції або визначення користувальницьких типів мають таку область видимості й доступні всім модулям у всіх проектах.

Іноді змінні, які мають область видимості **public**, називають глобальними змінними, тому що вони доступні глобально, тобто всій програмі. Можна почути від програмістів, що працюють не на VBA, що змінні й константи з областю видимості на рівні модуля є глобальними, тому що глобально доступні усередині окремого модуля. Щоб не було плутанини, в VBA термін private використовується для предметів з областю видимості на рівні модуля, а public – на рівні декількох модулів.

У списку перераховані три різних рівні видимості.

- Область видимості local. Змінні й константи описуються усередині процедури або функції. Вони доступні тільки усередині тієї процедури або функції, де були описані.
- Область видимості private. Змінні й константи описуються на рівні модуля. Вони доступні будь-якій процедурі або функції в межах того МОДУЛЯ, де були описані, але не поза нього.
- Область видимості public. Процедури, функції й обумовлені користувачем типи. Доступні в модулі, де були описані, а також будь-якому іншому модулю того ж проекту або інших проектів, за умови, що на проекті модулем, де вони описані, є посилання.

## **Додаток 5. Заходи безпеки під час виконання лабораторних робіт**

Цикл лабораторних робіт з дисципліни «Інформаційні технології» виконуються в комп'ютерному класі кафедри кібернетики хіміко-технологічних процесів хіміко-технологічного факультету, де розміщені персональні комп'ютери. Обладнання живиться електричним струмом напругою 220 В. Тому при виконанні лабораторних робіт слід дотримуватися заходів безпеки наступних інструкцій.

### **ІНСТРУКЦІЯ**

#### **з техніки безпеки при навчанні студентів на ПЕОМ в учбових лабораторіях кафедри кібернетики хіміко-технологічних процесів хіміко-технологічного факультету**

1. Знання і суворе дотримання цих правил є обов'язковим для всіх осіб, допущених до роботи на ПЕОМ. Доведення їх до кожного зі студентів підтверджується особистим підписом кожного з них у контрольному листі з техніки безпеки. Особи, які не одержали такого інструктажу та не поставили підпис у контрольному листі з техніки безпеки, до роботи на ПЕОМ не допускаються.
2. Всі роботи в учбових лабораторіях кафедри кібернетики ХТП проводяться лише з дозволу викладача або співробітника кафедри.
3. Під час проведення занять в учбовій лабораторії не повинні знаходитися сторонні особи, в тому числі студенти інших груп. Студенти не повинні самовільно залишати учбову лабораторію під час занять.
4. При роботі на ПЕОМ треба пам'ятати, що в них використовується напруга, небезпечна для життя.
5. Всі особи, працюючі в учбових лабораторіях кафедри КХТП повинні бути ознайомлені з правилами надання першої медичної допомоги при ураженні електричним струмом.
6. Перед вмиканням ПЕОМ кожен з працюючих повинен отримати дозвіл викладача або співробітника кафедри.
7. У випадках виникнення короткого замикання, горіння, диму, вогню в апаратурі, пристрій необхідно негайно вимкнути з мережі та доповісти викладачеві або співробітникові кафедри. Самостійні дії по усуненню пошкодження забороняються.
8. У випадку виходу з ладу обладнання або програмного забезпечення, що зумовлені іншими причинами, доповісти викладачеві або співробітникові кафедри. Вимкати апаратуру при цьому не дозволяється. Самостійні дії по усуненню пошкодження забороняються.
9. Працюючі в учбових лабораторіях кафедри кібернетики ХТП несуть майнову та адміністративну відповідальність за збереження та використання обладнання, наданого для їх праці.



10. Категорично забороняється:

- § самостійно вмикати та вимикати тумблери на щитку електроживлення;
- § несанкціоновано вмикати електрообладнання;
- § приносити та вмикати своє обладнання та пристрої, встановлювати власне програмне забезпечення;
- § залишати без нагляду увімкнені пристрої та лабораторію;
- § пересувати обладнання та комплектуючі;
- § підключати та відключати інформаційні кабелі та кабелі живлення;
- § використовувати власні носії інформації без дозволу викладачів або співробітників кафедри;
- § знаходитись в учбовій лабораторії у верхньому одязі.

Після закінчення занять обладнання не вимикається. Робоче місце має бути прибрано працюючим та перевірене викладачем чи співробітником кафедри

## ІНСТРУКЦІЯ

### **про міри пожежної безпеки у лабораторіях, учбових та робочих приміщеннях кафедри кібернетики хіміко-технологічних процесів хіміко-технологічного факультету**


1. Всі студенти повинні знати та ретельно виконувати «Загальні правила пожежної безпеки в НТУУ «КПІ»».
2. Завідуючий кафедрою та завідуючий лабораторією відповідають за забезпечення пожежної безпеки всіх приміщень кафедри та за справність протипожежного обладнання та сигналізації.
3. Все електричне обладнання, яке знаходиться в лабораторіях та приміщеннях кафедри, повинно мати заземлення.
4. В усіх приміщеннях повинно дотримуватись чистоти, не займати приміщення непотрібними меблями, обладнанням та матеріалами.
5. Всі двері основних та додаткових виходів утримувати у стані швидкого відкривання.
6. Зберігання та використання горючих та легкоспалахуючих рідин у приміщеннях кафедри забороняється.
7. Ремонт електричного обладнання проводити у строгій відповідності з правилами пожежної безпеки.
8. Всі електрозахисти повинні знаходитися у закритому положенні, не займаними сторонніми предметами.
9. Коридори, проходи, тамбури, евакуаційні виходи та підходи до першочергових засобів пожежогасіння, а також комунікаційні ніші повинні бути постійно вільними, чистими та нічим не зайнятими.
10. Відповідальні особи перед закриттям приміщень повинні ретельно оглянути їх, забезпечити прибирання виробничих відходів, перевірити якість перекриття води, газу, відключити напругу електромережі, перевірити стан пожежної сигналізації та засобів пожежогасіння.
11. Від усіх приміщень мати два комплекти ключів. Один комплект здавати черговому, а інший – зберігати в певному місці, яке відомо обслуговуючому персоналу.

Студенти повинні знати та ретельно виконувати «Загальні правила техніки безпеки в НТУУ «КПІ»», про що вони ставлять свій підпис у відповідному контрольному листі з техніки безпеки перед початком проведення циклу лабораторних робіт. Студенти, які не пройшли інструктаж і не поставили підпис у контрольному листі, до роботи не допускаються.

Додаток 6.Схема єдиного ділення

Хід	Етапи	Коефіцієнти			Вільні члени	Контрольні суми
		$x_1$	$x_2$	$x_3$		
Прямий	I	$a_{11}$	$a_{12}$	$a_{13}$	$b_1$	$c_1 = a_{11} + a_{12} + a_{13} + b_1$
		$a_{21}$	$a_{22}$	$a_{23}$	$b_2$	$c_2 = a_{21} + a_{22} + a_{23} + b_2$
	$a_{31}$	$a_{32}$	$a_{33}$	$b_3$	$c_3 = a_{31} + a_{32} + a_{33} + b_3$	
	$1 = \frac{a_{11}}{a_{11}}$	$a_{12} = \frac{a_{12}}{a_{11}}$	$a_{13} = \frac{a_{13}}{a_{11}}$	$b_1 = \frac{b_1}{a_{11}}$	$g_1 = \frac{c_1}{a_{11}} = 1 + a_{12} + a_{13} + b_1$	
	II	$a_{22}^{(1)} = a_{22} - a_{21}a_{12}$	$a_{23}^{(1)} = a_{23} - a_{21}a_{13}$	$b_2^{(1)} = b_2 - a_{21}b_1$	$c_2^{(1)} = c_2 - a_{21}g_1 = a_{22}^{(1)} + a_{23}^{(1)} + b_2^{(1)}$	
		$a_{32}^{(1)} = a_{32} - a_{31}a_{12}$	$a_{33}^{(1)} = a_{33} - a_{31}a_{13}$	$b_3^{(1)} = b_3 - a_{31}b_1$	$c_3^{(1)} = c_3 - a_{31}g_1 = a_{32}^{(1)} + a_{33}^{(1)} + b_3^{(1)}$	
III	$1 = \frac{a_{22}^{(1)}}{a_{22}^{(1)}}$	$a_{23} = \frac{a_{23}^{(1)}}{a_{22}^{(1)}}$	$b_2 = \frac{b_2^{(1)}}{a_{22}^{(1)}}$	$g_2 = \frac{c_2^{(1)}}{a_{22}^{(1)}} = 1 + a_{23} + b_2$		
		$a_{33}^{(2)} = a_{33}^{(1)} - a_{32}^{(1)}a_{23}$	$b_3^{(2)} = b_3^{(1)} - a_{32}^{(1)}b_2$	$c_3^{(2)} = c_3^{(1)} - a_{32}^{(1)}g_2 = a_{33}^{(2)} + b_3^{(2)}$		
Зворотній	IV			$1 = \frac{a_{33}^{(2)}}{a_{33}^{(2)}}$	$b_3 = \frac{b_3^{(2)}}{a_{33}^{(2)}}$	$g_3 = \frac{c_3^{(2)}}{a_{33}^{(2)}} = 1 + b_3$
				1	$x_3 = b_3$	$\overline{x_3} = g_3 = 1 + x_3$
			1		$x_2 = b_2 - a_{23}x_3$	$\overline{x_2} = g_2 - a_{23}\overline{x_3} = 1 + x_2$
	1			$x_1 = b_1 - a_{13}x_3 - a_{12}x_2$	$\overline{x_1} = g_1 - a_{13}\overline{x_3} - a_{12}\overline{x_2} = 1 + x_1$	

Періодична система Д.І. Менделєєва

		Периодическая система химических элементов Д. И. Менделеева						VII (H)		VIII		
1	1	<b>H</b> 1 1,00794 водород								2 <b>He</b> 4,002602 гелий	 Периодический закон открыт Д.И. Менделеевым в 1869 г.	
2	2	<b>Li</b> 3 6,941 литий	<b>Be</b> 4 9,01218 бериллий	5 <b>B</b> 10,811 бор	6 <b>C</b> 12,011 углерод	7 <b>N</b> 14,0067 азот	8 <b>O</b> 15,9994 кислород	9 <b>F</b> 18,998403 фтор	10 <b>Ne</b> 20,179 неон			
3	3	<b>Na</b> 11 22,98977 натрий	<b>Mg</b> 12 24,305 магний	13 <b>Al</b> 26,98154 алюминий	14 <b>Si</b> 28,0855 кремний	15 <b>P</b> 30,97376 фосфор	16 <b>S</b> 32,066 сера	17 <b>Cl</b> 35,453 хлор	18 <b>Ar</b> 39,948 аргон			
4	4	<b>K</b> 19 39,0983 калий	<b>Ca</b> 20 40,078 кальций	<b>Sc</b> 21 44,95591 скандий	<b>Ti</b> 22 47,88 титан	<b>V</b> 23 50,9415 ванадий	<b>Cr</b> 24 51,9961 хром	<b>Mn</b> 25 54,9380 марганец	<b>Fe</b> 26 55,847 железо	<b>Co</b> 27 58,9332 кобальт	<b>Ni</b> 28 58,69 никель	
	5	29 <b>Cu</b> 63,546 медь	30 <b>Zn</b> 65,39 цинк	31 <b>Ga</b> 69,723 галлий	32 <b>Ge</b> 72,59 германий	33 <b>As</b> 74,9216 мышьяк	34 <b>Se</b> 78,96 селен	35 <b>Br</b> 79,904 бром	36 <b>Kr</b> 83,80 криптон			
5	6	<b>Rb</b> 37 85,4678 рубидий	<b>Sr</b> 38 87,62 стронций	<b>Y</b> 39 88,9059 иттрий	<b>Zr</b> 40 91,224 цирконий	<b>Nb</b> 41 92,9064 ниобий	<b>Mo</b> 42 95,94 молибден	<b>Tc</b> 43 [98] технеций	<b>Ru</b> 44 101,07 рутений	<b>Rh</b> 45 102,9055 родий	<b>Pd</b> 46 106,42 палладий	
	7	47 <b>Ag</b> 107,8682 серебро	48 <b>Cd</b> 112,41 кадмий	49 <b>In</b> 114,82 индий	50 <b>Sn</b> 118,710 олово	51 <b>Sb</b> 121,75 сурьма	52 <b>Te</b> 127,60 теллур	53 <b>I</b> 126,9045 йод	54 <b>Xe</b> 131,29 ксенон			
6	8	<b>Cs</b> 55 132,9054 цезий	<b>Ba</b> 56 137,33 барий	<b>La*</b> 57 138,9055 лантан	<b>Hf</b> 72 178,49 гафний	<b>Ta</b> 73 180,9479 тантал	<b>W</b> 74 183,85 вольфрам	<b>Re</b> 75 186,207 рений	<b>Os</b> 76 190,2 осмий	<b>Ir</b> 77 192,22 иридий	<b>Pt</b> 78 195,08 платина	
	9	79 <b>Au</b> 196,9665 золото	80 <b>Hg</b> 200,59 ртуть	81 <b>Tl</b> 204,383 галлий	82 <b>Pb</b> 207,2 свинец	83 <b>Bi</b> 208,9804 висмут	84 <b>Po</b> [209] полоний	85 <b>At</b> [210] астат	86 <b>Rn</b> [222] радон			
7	10	<b>Fr</b> 87 [223] франций	<b>Ra</b> 88 [226] радий	<b>Ac**</b> 89 [227] актиний	<b>Rf</b> 104 [261] резерфордий	<b>Db</b> 105 [262] дубний	<b>Sg</b> 106 [263] сиборгий	<b>Bh</b> 107 [262] борий	<b>Hs</b> 108 [265] гасий	<b>Mt</b> 109 [266] майтнерий	<b>Ds</b> 110 [271] дармштадтий	
	11	111 <b>Rg</b> [272] рентгений	112 <b>Uub</b> [285] унубий	113 <b>(Uut)</b> [ ] унунтрий	114 <b>Uuq</b> [287] унунквадий	115 <b>(Uup)</b> [ ] унунпентий	116 <b>Uuh</b> [292] унунгексий	117 <b>(Uus)</b> [ ] унунсептий	118 <b>Uuo</b> [293] унуноктий			

\* Лантаноиды

<b>Ce</b> 58 140,12 церий	<b>Pr</b> 59 140,9077 празеодим	<b>Nd</b> 60 144,24 неодим	<b>Pm</b> 61 [145] прометий	<b>Sm</b> 62 150,36 самарий	<b>Eu</b> 63 151,96 европий	<b>Gd</b> 64 157,25 гадолиний	<b>Tb</b> 65 158,9254 тербий	<b>Dy</b> 66 162,50 диспрозий	<b>Ho</b> 67 164,9304 гольмий	<b>Er</b> 68 167,26 эрбий	<b>Tm</b> 69 168,9342 тулий	<b>Yb</b> 70 173,04 иттербий	<b>Lu</b> 71 174,967 лютеций
---------------------------------	---------------------------------------	----------------------------------	-----------------------------------	-----------------------------------	-----------------------------------	-------------------------------------	------------------------------------	-------------------------------------	-------------------------------------	---------------------------------	-----------------------------------	------------------------------------	------------------------------------

\*\* Актиноиды

<b>Th</b> 90 232,0381 торий	<b>Pa</b> 91 [231] протактиний	<b>U</b> 92 238,0289 уран	<b>Np</b> 93 [237] нептуний	<b>Pu</b> 94 [244] плутоний	<b>Am</b> 95 [243] амерций	<b>Cm</b> 96 [247] курий	<b>Bk</b> 97 [247] берклий	<b>Cf</b> 98 [251] калифорний	<b>Es</b> 99 [252] эйнштейний	<b>Fm</b> 100 [257] фермий	<b>Md</b> 101 [258] менделевий	<b>No</b> 102 [259] нобелий	<b>Lr</b> 103 [260] лоуренсий
-----------------------------------	--------------------------------------	---------------------------------	-----------------------------------	-----------------------------------	----------------------------------	--------------------------------	----------------------------------	-------------------------------------	-------------------------------------	----------------------------------	--------------------------------------	-----------------------------------	-------------------------------------

Целое число в скобках – массовое число наиболее устойчивого изотопа

Скласти макрос у відповідності до свого варіанту.

Розробити таблицю в середовищі Excel. Інформація про назви полів таблиць вказана в завданнях за варіантами. Заповнити поля та відформатувати таблицю наступним чином: зовнішню границю таблиці виконати подвійною лінією, назви стовпчиків відокремити лінією товщиною 2,25 пт, заголовки стовпчиків затисати шрифтом “**Arial-14**”, “**жирный**”, вирівнювання в комірках таблиці “**по центру**”, фон сірий.

### **Варіанти завдань до лабораторної роботи №1**

1. Інформація про абітурієнтах, які подали заявки для вступу в «НТУУ» КПІ, представлена такими даними: реєстраційний номер, дата реєстрації, прізвище, та ініціали, дата народження, стать, адреса, спеціальність.

2. Інформація про працюючих на факультеті ХТФ співробітників представлена такими даними: особистий номер, прізвище та ініціали, дата народження, адреса, освіта, кафедра, дата прийому на роботу.

3. Інформація про продані магазином хімічного обладнання та реактивів товари представлена такими даними: обліковий номер продажу, дата продажу, найменування товару, прізвище та ініціали покупця, ціна товару, знижка (може бути відсутня).

4. Інформація про продані фірмою комп'ютери і їх комплектуючі представлена такими даними: обліковий номер продажу, прізвище та ініціали покупця, дата продажу, найменування, кількість, ціна за одну одиницю, загальна ціна.

5. Інформація про послуги, що надані компанією з ремонту комп'ютерів своїм клієнтам, представлена такими даними: обліковий номер наданої послуги, дата виконання роботи, прізвище, та ініціали клієнта, серійний номер пристрою, прізвище та ініціали майстра, що виконував роботу, вартість роботи, вартість матеріалів.

6. Інформація про рейси, що виконуються компанією, представлена такими даними: дата рейсу, номер рейсу, тип транспортного засобу, пункт призначення, кількість пасажирів, ціна квитка, загальні витрати на рейс.

7. Інформація про виклики бригад швидкої допомоги, представлена такими даними: дата виклику, номер бригади, адреса, прізвище та ініціали пацієнта, дата народження, стать, дії.

8. Інформація про перевезення пасажирів даним автопідприємством представлена такими даними: дата роботи, марка автобуса, бортовий номер автобуса, номер маршруту, прізвище та ініціали водія, час початку роботи, час закінчення роботи.

9. Інформація про книги, що знаходяться в особистій бібліотеці, представлена такими даними: жанр, автор, назва, рік видання, видавництво, обліковий номер книги, ціна.

10. Інформація про договори на експорт сировини представлена такими даними: дата укладення договору, номер договору, фірма-постачальник, вид сировини, країна, фірма-одержувач, ціна.

11. Інформація про успішність студентів представлена такими даними: факультет, курс, група, прізвище та ініціали, предмет, оцінка на іспиті, номер студентського квитка.

12. Інформація про послуги, що надаються клієнтам салону краси, представлена такими даними: дата надання послуги, прізвище та ініціали майстра, вид послуги, ціна послуги, знижка постійним клієнтам, частка майстра.

13. Інформація про клієнтів туристичної фірми представлена такими даними: рік надання послуги, країна призначення, місце призначення, дата початку туру, дата закінчення туру, вартість туру, прізвище та ініціали клієнта, транспортний засіб доставки.

14. Інформація про товари, що є на складі, представлена такими даними: ідентифікаційний номер товару, категорія товару, найменування товару, одиниця виміру (наприклад, ящик, пачка, мішок і т.д.), кількість одиниць на складі, ціна за одиницю товару.

15. Інформація про продажі товарів зі складу представлена такими даними: дата продажу, найменування товару, кількість одиниць, ціна за одиницю, загальна ціна, прізвище та ініціали покупця (назва фірми-покупця).

16. Інформація про договори на продаж автомобілів представлена такими даними: дата укладення договору, номер договору, прізвище, ім'я та по батькові клієнта, телефон клієнта, марка автомобіля, рік випуску, ціна.

17. Інформація про облік автомобілів представлена такими даними: марка автомобіля, рік випуску, колір, реєстраційний номер, прізвище та ініціали власника, дата постановки на облік.

18. Інформація про абонентів телефонної станції представлена такими даними: номер телефону, дата підключення, прізвище, ім'я та по батькові абонента, адреса, поверх, наявність пільг.

19. Інформація про пацієнтів стоматологічної клініки представлена такими даними: номер картки, прізвище та ініціали пацієнта, дата народження, телефон, дата надання послуг, вид послуги, вартість матеріалів, вартість роботи.

20. Інформація видавничого підприємства представлена такими даними: реєстраційний номер, дата реєстрації, прізвище та ініціали автора (першого, якщо є співавтори), назва, прізвище та ініціали рецензента, номер журналу, рік видання.

21. Інформація про продаж та прокат відеотеки представлена такими даними: реєстраційний номер, тип носія назва, дата продажу або прокату, прізвище та ініціали клієнта, кількість, ціна за одиницю.

22. Інформація про продажі магазину представлена такими даними: дата продажу, найменування товару, обліковий номер товару, кількість одиниць, вартість одиниці товару, загальна вартість, прізвище та ініціали продавця.

23. Інформація про договори, що укладені компанією мобільного зв'язку, представлена такими даними: реєстраційний номер, дата укладення договору, прізвище та ініціали абонента, початок дії послуги, кінець дії послуги, тариф, виконавець.

24. Інформація про фільми представлена такими даними: жанр, назва, код фільму, режисер, країна виробник, тривалість показу, вікові обмеження.

25. Інформація про продажі взуттєвої фірми, представлена такими даними: назва товару, країна-виробник, рік випуску, обліковий номер товару, дата продажу, ціна за одиницю товару (пару), прізвище та ініціали продавця.

26. Інформація про продані магазином матеріали представлена такими даними: дата продажу, найменування, обліковий номер товару, ціна за одиницю, кількість одиниць, загальна вартість, прізвище та ініціали продавця.

27. Інформація про облік виконання лабораторних робіт представлена такими даними: дата, прізвище та ініціали студента, назва лабораторної роботи, оцінка за підготовку, оцінка за виконання, оцінка за захист, штрафні та заохочувальні бали.

28. Інформація про студентів групи представлена такими даними: назва групи, прізвище та ініціали, рік народження, адреса, домашній телефон, мобільний телефон, електронна пошта.

29. Інформація до нарахування квартплати представлена такими даними: платник, загальна площа квартири, житлова площа, кількість кімнат, поверх, площа прибудинкової території, пільги.

30. Інформація до сплати за спожиту електроенергію представлена такими даними: платник, адреса, початкові показання, кінцеві показання, різниця, тариф, пільги.

31. Інформація про відрядження представлена такими даними: прізвище та ініціали, країна, місто, мета відрядження, дата вибуття, термін відрядження, сума витрат.

32. Інформація про операції служби таксі представлена такими даними: дата поїздки, адреса подачі машини, адреса призначення, приблизний кілометраж, тариф, вартість, прізвище та ініціали оператора.

33. Інформація про змагання з фігурного катання представлена такими даними: учасник, країна, порядковий номер виступу, оцінка за техніку виконання, оцінка за художнє враження, сумарна оцінка, загальне місце у списку учасників.

34. Інформація на автомобільній стоянці представлена такими даними: реєстраційний номер автомобіля, марка (модель), дата в'їзду, дата виїзду, денний тариф, вартість.

Для елементів з таблиці Менделєєва, згідно варіантів індивідуального завдання створити таблицю в середовищі Excel засобами макрорекодера. В стовпцях таблиці представити наступну інформацію: номер по порядку, назва елемента, його символічне позначення, порядковий номер елемента в таблиці Менделєєва, його атомна вага, властивості елемента (метал/неметал). Заповнити рядки даними згідно варіантів індивідуального завдання. Відформатувати таблицю наступним чином: зовнішню границю таблиці виконати подвійною лінією, назви стовпчиків відокремити лінією товщиною 2,25 пт, заголовки стовпчиків затисати шрифтом “Arial-14”, “жирний”, вирівнювання в комірках таблиці “по центру”, фон сірий.

1. Заповнити таблицю елементами групи **I**.
2. Заповнити таблицю елементами групи **II**.
3. Заповнити таблицю елементами групи **III**.
4. Заповнити таблицю елементами групи **IV**.



5. Заповнити таблицю елементами групи **V**.
6. Заповнити таблицю елементами групи **VI**.
7. Заповнити таблицю елементами групи **VII**.
8. Заповнити таблицю елементами групи **VIII**.
9. Заповнити таблицю елементами **1** періоду.
10. Заповнити таблицю елементами **2** періоду.
11. Заповнити таблицю елементами **3** періоду.
12. Заповнити таблицю елементами **4(4)** періоду.
13. Заповнити таблицю елементами **4(5)** періоду.
14. Заповнити таблицю елементами **5(6)** періоду.
16. Заповнити таблицю елементами **5(7)** періоду.
17. Заповнити таблицю елементами **6(8)** періоду.
18. Заповнити таблицю елементами **6(9)** періоду.
19. Заповнити таблицю елементами **7(10)** періоду.
20. Заповнити таблицю елементами **7(11)** періоду.
21. Заповнити таблицю елементами лантаноїдів.
22. Заповнити таблицю елементами актиноїдів.

## Додаток 7. Типи даних VBA

Тип даних	Розмір (байт)	Діапазон значень
Integer	2	Цілі числа від – 32 768 до 32 767
Long	4	Цілі числа від – 2 147 483 648 до 2 147 483 647
Single	4	Дійсні числа від –3,402823E38 до –1,401298E-45 (від’ємні) і від 1,401298E-45 до 3,402823E38 (додатні)
Double	8	Дійсні числа від –1,79769313486232E308 до –4,94065645841247E-324 (від’ємні) і від 4,94065645841247E-324 до 1,79769313486232E308 (додатні)
Byte	1	Додатні цілі числа від 0 до 255
Boolean	2	Логічні значення: Істина (True) або Хибність (False)
Currency	8	Від – 922 337 203 685 477,5808 до 922 337 203 685 477,5807
Date	8	Значення дат від 1 січня 100 року до 31 грудня 9999 року й значення часу від 0:00:00 до 23:59:59.
Decimal	14	±79 228 162 514 264 337 593 543 950 335 (без десяткових знаків) або ±7,9228162514264337593543950335 (з 28 десятковими розрядами)
Object	4	Посилання на будь-який об’єкт (зберігає адресу об’єкта)
String (змінної довжини)	10 + 1*n	Від 0 до приблизно 2 млн. символів
String (фіксованої довжини)	Довжина рядка	Від 0 до приблизно 65 400. символів
Variant (універсальний числовий)	16	Будь-яке число з діапазону Double
Variant (універсальний рядковий)	22 + 1*n	Той же діапазон, що й для рядкового типу зі змінною довжиною

Тип даних **Byte** – указує на цілі числа від 0 до 255. У числах типу Byte не можна зберігати від'ємні величини.

Тип даних **Currency** – з типом з фіксованою комою (праворуч від коми розташовані чотири значущі цифри). Математичні обчислення із числами цього типу мають малу помилку округлення або не мають помилки взагалі. Тут точність вище, ніж при використанні чисел із плаваючою комою, і тому цей тип використовують для грошових обчислень.

Тип даних – **Boolean** (логічний тип) використовується для зберігання результатів перевірки деяких умов. Можуть приймати тільки два значення – **Істина (True)** або **Хибність (False)**.

Дані типу **Variant**

Variant – це спеціальний тип даних, у якому можна зберігати всі типи даних, перераховані в таблиці. Коли тип явно не зазначений, VBA використовує тип даних Variant.

Дані типу Variant мають властивості тих даних, які фактично в них записані.

Будь-яка змінна, для якої явно не вказали тип, стає змінною типу Variant. Наприклад, якщо в даних типу Variant записаний текстовий рядок, вони мають властивості даних типу String, а якщо там записане число, то вони мають властивості числового типу, найчастіше типу Double, хоча це може бути й Integer, і Long, і Single й Currency.

Дані типу Variant використовують саме компактне з можливих подань. Наприклад, якщо в даних типу Variant записане ціле число, то ці дані будуть трактуватися як Integer або як Long, залежно від розміру цього числа. Число 15 у змінній типу Variant буде трактуватися як Integer, а число 1000000 – як Long.

VBA зберігає числа із плаваючою комою в змінних типу Variant, як **Double**.

Хоча використання змінних типу Variant зручно й звільняє від багатьох турбот при написанні процедур, але такі змінні вимагають більше пам'яті й більшість операцій виконується на них повільніше, ніж з іншими типами. Варто уникати невиправданого використання змінних типу Variant, тому що це може зробити програми повільними й утруднити їхнє читання. Крім того, це може привести до появи помилок, що важко виявити.

Важливість оголошення змінних полягає в наступному:

- використання змінних різних типів зменшує пам'ять, займану змінними, і скорочує час виконуваних обчислень, завдяки чому програми стають ефективнішими (якщо програма точно знає тип даних, вона не виконує додаткову перевірку даних і резервує рівно стільки пам'яті, скільки необхідно для зберігання кінцевих даних);
- оголошення змінних виключає перетворення значень змінних, яке відбувається при обчисленні виразів, з типу **Variant** в інший тип;
- оголошення змінних дозволяє створювати змінні, які можна використовувати в декількох процедурах або модулях (локальні і

глобальні змінні). Можна визначати також змінні, що зберігають свої значення після виходу з процедури;

- якщо змінна оголошена, то **Visual Basic** перевіряє, чи використовується правильний тип даних в операціях з цією змінною.

Якщо тип не вказаний – за замовченням система приймає тип **Variant**.

Оголошувати змінні треба обов'язково. І при цьому бажано явно вказувати потрібний тип даних, бо:

- скорочується кількість помилок: програма із самого початку відмовиться приймати в змінну значення неправильного типу (наприклад, строкове замість числового);
- при роботі з об'єктами підказка по властивостям і методам діє тільки тоді, коли ми з самого початку оголосили об'єктну змінну з потрібним типом.

Деякі моменти, пов'язані з вибором типів даних для змінних:

- загальний принцип – вибирайте найменший тип даних, який може вмістити вибрані вами значення. Якщо є якісь сумніви – обирайте більший тип щоб уникнути виникнення помилок;
- якщо є можливість, краще не використовувати типи даних з плаваючою комою (**Single** і **Double**). Робота з ними проводиться повільніше, крім того, можуть бути проблеми при порівняннях за рахунок округлень.

### Вказування типу змінної за допомогою спеціальних символів

Повідомляючи змінну неявно, можна вказати її тип, додаючи наприкінці імені змінної спеціальний символ. Такі символи називаються символами визначення типу. Символи визначення типу змінної наведені в таблиці.

Символ	%	&	!	#	@	\$
<i>Tun</i>	<i>Integer</i>	<i>Long</i>	<i>Single</i>	<i>Double</i>	<i>Currency</i>	<i>String</i>

Таких символів шість. Не існує символів визначення типу для таких типів, як **Byte**, **Boolean**, **Date**, **Object** (или **Array**). Символи визначення типу можуть знаходитись тільки в кінці імені змінної.

Наприклад, *авс%*

Символи визначення типу – це дань традиції VBA діалектам мови **Basic** – це був єдиний спосіб вказати тип змінної. Насправді користуватись ними нема необхідності. Інструкція **Dim** – набагато зручніший і зрозуміліший спосіб. Більшість сучасних програмістів не користуються символами визначення типу.

## Додаток 8. Аргумент **buttons** функції **MsgBox**

Аргумент **buttons** дозволяє управляти наступними параметрами вікна повідомлення:

- Кількість кнопок у вікні;
- Типи кнопок і їх розміщення у вікні;
- Піктограма, що відображається у вікні;
- Яка кнопка призначається за умовчанням;
- Режим (модальність) вікна повідомлення.

Допустимі значення аргументу **buttons**:

<i>Константа</i>	Значення	Опис
<b>vbOKOnly</b>	<b>0</b>	Відображається тільки кнопка "ОК".
<b>VbOKCancel</b>	<b>1</b>	Відображаються кнопки "ОК" і "Відміна" (Cancel).
<b>VbAbortRetryIgnore</b>	<b>2</b>	Відображаються кнопки "Перервати" (Abort), "Повторити" (Retry) і "Пропустити" (Ignore).
<b>VbYesNoCancel</b>	<b>3</b>	Відображаються кнопки "Так" (Yes), "Ні" (No) і "Відміна" (Cancel).
<b>VbYesNo</b>	<b>4</b>	Відображаються кнопки "Так" (Yes) і "Ні" (No).
<b>VbRetryCancel</b>	<b>5</b>	Відображаються кнопки "Повторити" (Retry) і "Відміна" (Cancel).
<b>VbCritical</b>	<b>16</b>	Використовується значок "Критичне повідомлення".
<b>VbQuestion</b>	<b>32</b>	Використовується значок "Застережливий запит".
<b>VbExclamation</b>	<b>48</b>	Використовується значок "Попередження".
<b>VbInformation</b>	<b>64</b>	Використовується значок "Інформаційне повідомлення".
<b>VbDefaultButton1</b>	<b>0</b>	Основною є перша кнопка.
<b>VbDefaultButton2</b>	<b>256</b>	Основною є друга кнопка.
<b>VbDefaultButton3</b>	<b>512</b>	Основною є третя кнопка.
<b>VbDefaultButton4</b>	<b>768</b>	Основною є четверта кнопка.
<b>VbApplicationModal</b>	<b>0</b>	Модальне вікно на рівні додатку: щоб продовжити роботу з поточним застосуванням, необхідно відповісти на дане повідомлення.
<b>VbSystemModal</b>	<b>4096</b>	Модальне вікно на рівні системи: всі застосування будуть недоступні до тих пір, поки користувач не відповість на дане повідомлення.

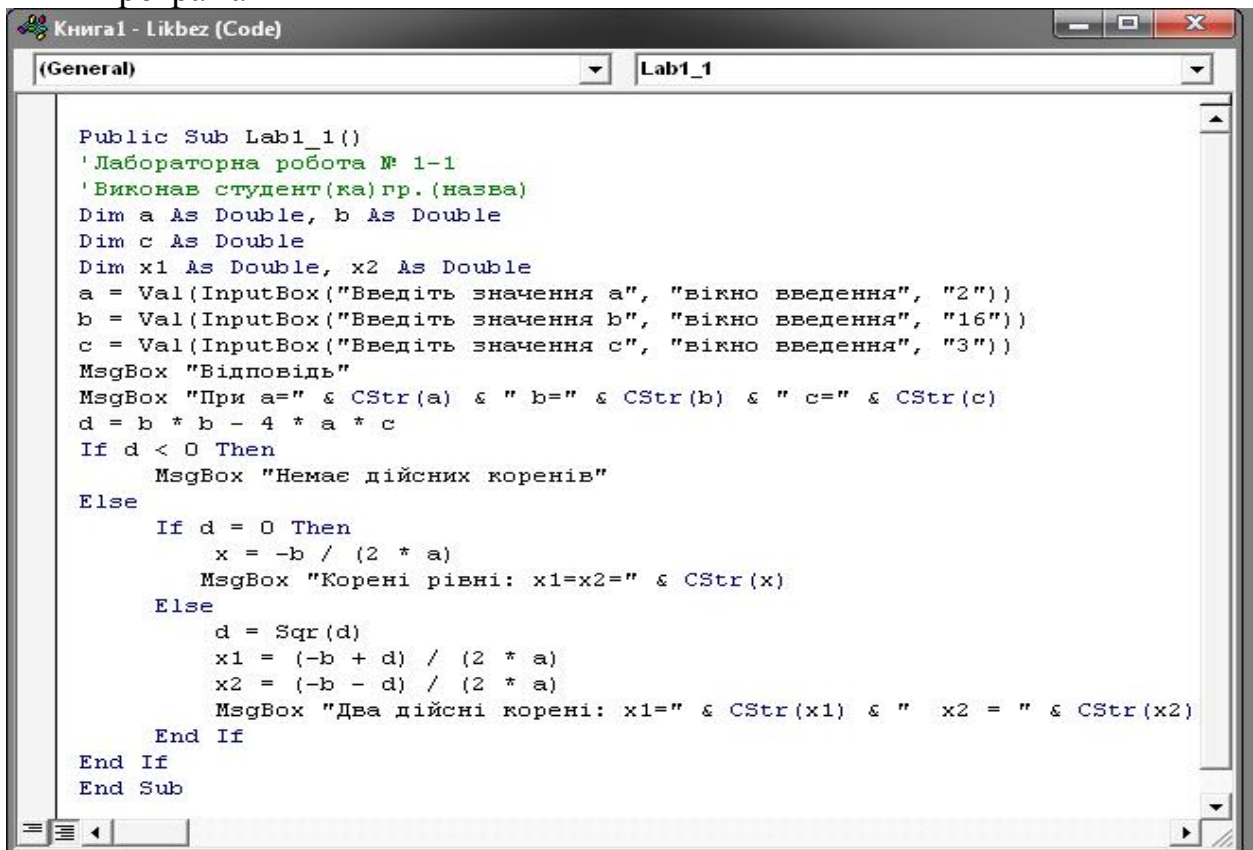
Перша група значень (0–5) указує число і тип кнопок, що відображаються у вікні діалогу, друга група (16, 32, 48, 64) задає тип використовуваного значка, третя група (0, 256, 512) визначає кнопку, яка є основною, а четверта група (0, 4096) – модальність вікна повідомлення. При визначенні значення аргументу **buttons** слід підсумовувати не більш за одне значення з кожної групи.

#### Значення, повертані функцією MsgBox

Константа	Значення	Натиснута кнопка
<b>vbOK</b>	<b>1</b>	<b>ОК</b>
<b>vbCancel</b>	<b>2</b>	<b>Відміна (Cancel)</b>
<b>vbAbort</b>	<b>3</b>	<b>Перервати (Abort)</b>
<b>vbRetry</b>	<b>4</b>	<b>Повторити (Retry)</b>
<b>vbIgnore</b>	<b>5</b>	<b>Пропустити (Ignore)</b>
<b>vbYes</b>	<b>6</b>	<b>Так (Yes)</b>
<b>vbNo</b>	<b>7</b>	<b>Немає (No)</b>

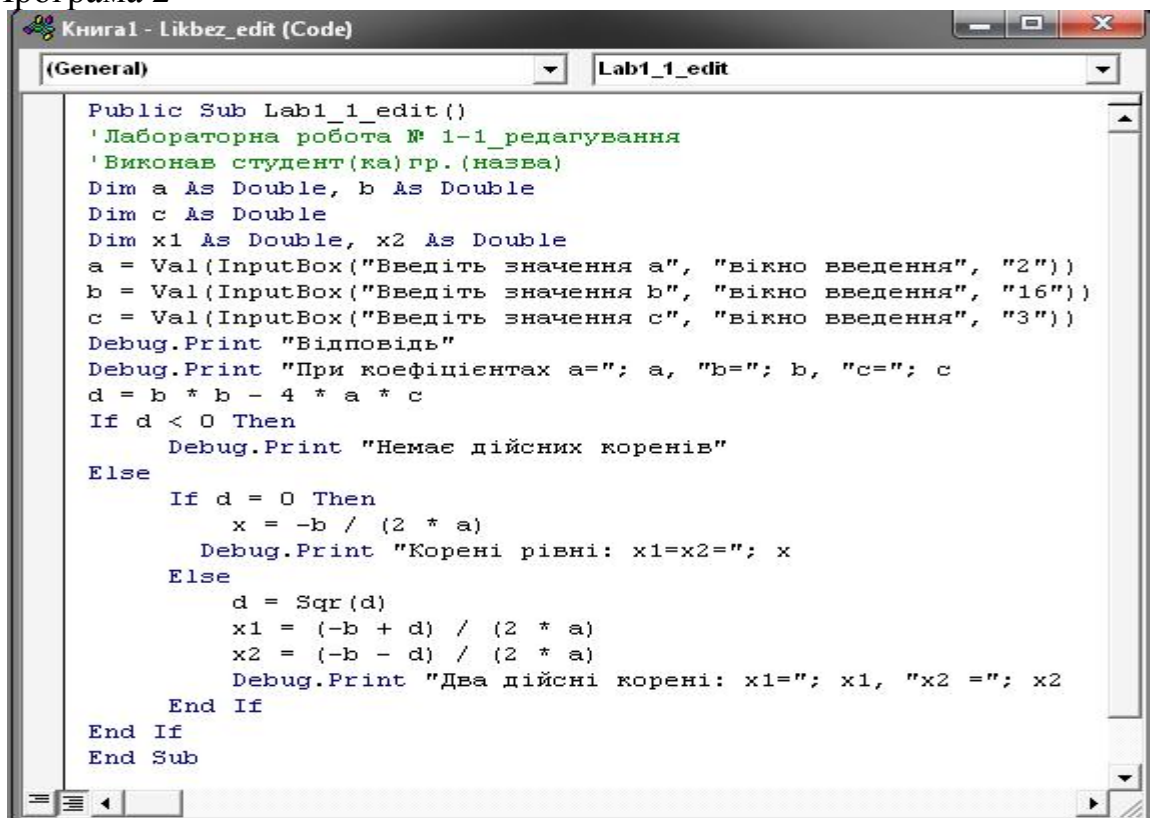
## Додаток 9.Програми рішення квадратного рівняння

### Програма 1



```
Public Sub Lab1_1()  
'Лабораторна робота № 1-1  
'Виконав студент(ка) гр. (назва)  
Dim a As Double, b As Double  
Dim c As Double  
Dim x1 As Double, x2 As Double  
a = Val(InputBox("Введіть значення а", "вікно введення", "2"))  
b = Val(InputBox("Введіть значення b", "вікно введення", "16"))  
c = Val(InputBox("Введіть значення с", "вікно введення", "3"))  
MsgBox "Відповідь"  
MsgBox "При а=" & CStr(a) & " b=" & CStr(b) & " c=" & CStr(c)  
d = b * b - 4 * a * c  
If d < 0 Then  
    MsgBox "Немає дійсних коренів"  
Else  
    If d = 0 Then  
        x = -b / (2 * a)  
        MsgBox "Корені рівні: x1=x2=" & CStr(x)  
    Else  
        d = Sqr(d)  
        x1 = (-b + d) / (2 * a)  
        x2 = (-b - d) / (2 * a)  
        MsgBox "Два дійсні корені: x1=" & CStr(x1) & " x2 = " & CStr(x2)  
    End If  
End If  
End Sub
```

### Програма 2



```
Public Sub Lab1_1_edit()  
'Лабораторна робота № 1-1_редагування  
'Виконав студент(ка) гр. (назва)  
Dim a As Double, b As Double  
Dim c As Double  
Dim x1 As Double, x2 As Double  
a = Val(InputBox("Введіть значення а", "вікно введення", "2"))  
b = Val(InputBox("Введіть значення b", "вікно введення", "16"))  
c = Val(InputBox("Введіть значення с", "вікно введення", "3"))  
Debug.Print "Відповідь"  
Debug.Print "При коефіцієнтах а="; a, "b="; b, "c="; c  
d = b * b - 4 * a * c  
If d < 0 Then  
    Debug.Print "Немає дійсних коренів"  
Else  
    If d = 0 Then  
        x = -b / (2 * a)  
        Debug.Print "Корені рівні: x1=x2="; x  
    Else  
        d = Sqr(d)  
        x1 = (-b + d) / (2 * a)  
        x2 = (-b - d) / (2 * a)  
        Debug.Print "Два дійсні корені: x1="; x1, "x2 ="; x2  
    End If  
End If  
End Sub
```

### Додаток 10.Завдання до лабораторної роботи № 2 (частина 2)

Розробити програму для обчислення арифметичного виразу при заданих значеннях змінних.

#### I рівень

№ вар.	$y$	$x$	$a$	$b$	$c$	$d$
1	2	3	4	5	6	7
1	$y = d \cdot \ln\left(\frac{a}{x}\right) + b \cdot \lg\left(\frac{x^2}{c}\right)$	0,1	7,1	-0,8	0,5	0,2
2	$y = a \cdot \sin^2 x + \frac{\cos(b \cdot x^2)}{c \cdot \operatorname{tg}\left \frac{x}{d}\right }$	2	0,5	-0,4	-4,4	-1
3	$y = \operatorname{tg}(b \cdot x^3) + \frac{\sin(2a \cdot x)}{c - d}$	-5	0,85	2,2	1	1,2
4	$y = c \cdot e^{\sin a \cdot x - \cos b \cdot x} + \frac{x^3}{d \cdot a}$	-1	-1,1	0,8	0,33	0,05
5	$y = c \cdot 10^{a \cdot x} + \frac{10^{d \cdot x}}{b \cdot x^3}$	6,2	0,2	-2,1	1,1	0,1
6	$y = \frac{d \cdot x^2}{1 + c \cdot \sqrt{a \cdot \sin^2 x - b^3 x^2}}$	6	1,1	0,1	9,2	-0,3
7	$y = \frac{\log_a x^c}{\sqrt{ \log_5(b \cdot x) } + 1} + d$	9,6	8	0,5	-2	-0,65
8	$y = \operatorname{arctg}(b^2 x) + \frac{a \cdot \operatorname{tg}^c x}{d \cdot x}$	2	0,6	-0,1	4	-0,6
9	$y = \sqrt{ a \cdot x + b \cdot x^2 } - \frac{1}{\lg x^c + d}$	0,95	-2,5	0,1	3	0,15
10	$y = \frac{a \cdot e^x + b \cdot e^{-x}}{c \cdot e^d}$	-10	1,2	4,3	-1	2
11	$y = \lg^2 \frac{x^c}{a} - d \cdot e^{-b \cdot x}$	0,2	0,95	-1	2	0,8
12	$y = \operatorname{tg}^2(b \cdot x^3) - \frac{\operatorname{ctg}(2a \cdot x)}{c + d}$	3	7	-0,4	6	-0,2
13	$y = \frac{a \cdot \cos^b x + b \cdot \sin x^a}{c \cdot d}$	-2,5	5	3	25	5
14	$y = \frac{a}{c \cdot 10^x} + b \cdot \lg \frac{x^d}{d}$	4	-3	0,2	2	4
15	$y = \frac{a^3 \sqrt{c \cdot x^2 - d}}{ b + x } + 1$	0,6	0,1	0,5	4	5



1	2	3	4	5	6	7
16	$y = \frac{a \sin(x^2) - \cos^2(bx)}{c \cdot \lg x-d }$	2,5	-3	0,25	0,5	1,1
17	$y = c \cdot \frac{e^{ax} + e^{x/b}}{\ln d-x }$	-4,1	0,43	-2,5	0,5	4
18	$y = \sqrt{\frac{b \cdot \operatorname{tg}(x+a)}{(c \cdot x - d)^2}}$	0,6	6,2	0,5	0,2	-4
19	$y = \frac{\sqrt{\left  \operatorname{tg}(x-a) + \operatorname{ctg}\left(\frac{b}{x}\right) \right }}{c \cdot (x-d)^2}$	-2	-0,8	25	1	3
20	$y = \frac{a(x-b)^2}{\sqrt{ \ln(x^2) + \lg^2(d \cdot x) }} \cdot c$	2	-0,8	3,5	1,1	3
21	$y = \frac{a(c \cdot x - b)^3}{\sqrt{ \cos(x^2) + \sin^2(d+x) }}$	-15	0,12	-1,3	1,5	0,3
22	$y = \frac{a^2x + bx^3}{c \cdot \sqrt{\operatorname{tg}(x^2) + \operatorname{arctg}^2(d-x)}}$	5	-2,4	4	-1,5	6
23	$y = \frac{\cos(ax^2) - b \sin^2(x)}{c \cdot \sqrt[3]{ x-d }}$	2,5	-3	0,25	0,1	1,1
24	$y = \frac{e^{x^a} + e^{x/b}}{\log_c d-x }$	3	2	-1	5	7
25	$y = b \left( \frac{a+1}{c \cdot 10^x} + b \cdot \lg \frac{x^d}{d} \right)$	4	-3	0,2	2	4

## II рівень

№ вар.	у	Значення змінних
1	2	3
1	$y = \frac{\lg^4 -x  - \arctg(p/3a)}{2\sqrt{x^3}} - \frac{ctg^2(a) + \sqrt[3]{5x-1.5}}{\sin(a-p) - \sqrt{e}} + 0.5x^a$	x=0.71 a=1.43
2	$y = \frac{\sqrt[5]{e} - \cos^3(2.7-p)}{\ln( a-x ^3) + tg(2x^2)} + \frac{\sqrt{ x-a/3p }}{1/2 - e^{2x}} - \sin(x+p/6)^2$	x=0.95 a=1.25
3	$y = -\frac{a^e - \sin(x-p)^2}{\lg(a-2.3x)^2 + \arctg(p+3/px)} + \frac{\cos^2(1.5x^3) - tg(0.7-p)}{7.8\sqrt{x^5}} + \sqrt[5]{\ln^2(x)}$	x=0.63 a=2.01
4	$y = \frac{\arctg(p-0.1a) + \lg^2(x+a)}{4.5 \cos(x-2p)^2 - 7x} + \frac{e^{2x} tg^2(x+p)/3a}{2 \sin(x^2)} - 0.6\sqrt[3]{x}$	x=1.28 a=0.44
5	$y = -\frac{\sin(a-p)^2 + \sqrt{ x-a }}{5.3 \cos^2(x-p)} + \frac{\lg^3 x=a  - ctg(8.4a)}{2a\sqrt{(x-a)^5} -  a-x } + \arctg(x+p)$	x=1.41 a=0.99
6	$y = -\frac{e^{0.2x} - \lg tg(x)-a }{\sin^2(x-p) - 7.3} + \frac{\arctg(xp) - \sqrt[3]{\ln(x)}}{2ctg(x-p)} + \cos(x^2+p)$	x=0.87 a=2.98
7	$y = -\frac{ctg^3(x-p) + \sqrt[3]{\sin(2x^3) - 7.2}}{2.5x} + \frac{\cos(x^2+p/2) - 9.6}{e^{3.5a-x} + tg^2(x)} + \lg( x=a )$	x=4.06 a=5.17
8	$y = \frac{tg(x-p/3)^2 - \ln^2( 7.2x-a )}{8.45ctg^2(x)} - \frac{3.4a - \sqrt[3]{(x-a)^2}}{\cos(x-p/4)^2 - \sin(x-2p)} + 2.2e^{2x}$	x=3.01 a=0.55
9	$y = \frac{\arctg^2(x-p/6) + 2e^{a-2x}}{3.7 \ln( x^2 - \sin(x)+1 )} + \frac{\sin(x-p/6) - a}{\cos^2(2x) + \sqrt[3]{xa} - 6.5x} - ctg(x-p/3)^2$	x=3.66 a=1.77
10	$y = \frac{\lg^2(\cos(x-p)^3) - tg(x-p/6)}{\arctg^2(x+p/3) - 4.6a} + \frac{ctg(a+p/2) + \sqrt[3]{x^2a-4.3}}{2.3a} - e^{ax}$	x=1.34 a=2.65
11	$y = \frac{\cos^2(x-p/3) + \sin(x^2)}{2e^{0.25x}} + \frac{\sqrt{ x-a } - 8.3}{\arctg(x^3) + \lg(xa)}$	x=2.61 a=0.51
12	$y = \frac{\sqrt[3]{ a-x^2 } + ctg^2(x+p/4)}{3.3 \lg^2(x)} + \arctg(x^2-p) - e^{2x}$	x=1.23 a=0.67
13	$y = \frac{\lg^3( x-a ) + \cos(x^2-p/3)}{4.3 \arctg^2(a+p/2)} + \sqrt[3]{x+a} + ctg(x^2+p/4)$	x=3.78 a=2.45
14	$y = \frac{\sin^2(x+p/6) + \sqrt[3]{3.4+x-a^2}}{\cos(x^2-6.7) + \ln( x-a )} - \arctg(x^2)e^{3x}$	x=2.75 a=5.23

<b>1</b>	<b>2</b>	<b>3</b>
15	$y = \frac{e^{2x} - \sqrt[3]{\lg x-a } + ctg^2(x-p)}{\cos(x^2+p/6)\sin(x^2+p/6)} + arctg^2(x+p/3)$	x=6.08 a=3.72
16	$y = \frac{arctg(p/3a) - \sqrt[3]{e}}{ctg^2(a) + \sqrt[3]{6x-2.5}} - \ln(x^2-2a) + e^{4x}$	x=1.72 a=2.41
17	$y = \frac{\sqrt[3]{\lg(x)} - \cos^2(2.1-p/2)}{\sqrt{ tg(2x^2) } - (\ln a-x ^3)} + e^{3x-1} arctg(x+p/6)^3$	x=2.95 a=0.25
18	$y = -\frac{arctg(p+3/px) - \cos(x-p)^2}{\lg(a-.3x)^2 + e^{3x}} + ctg^2(2.5x^3) + \sqrt[3]{\ln^2(x)}$	x=0.53 a=4.01
19	$y = \frac{e^{2x} ctg^2(x+p)/3a - 2.5 \cos(x-2p)^2}{2 \sin(x^2)} - 0.4 \sqrt[3]{x+a^3} - \lg^2(x+a)$	x=1.37 a=0.24
20	$y = \frac{\lg^4 x-2a  - ctg(2.4a)}{3a\sqrt{(x-a)^5} -  a-3.1x } + arctg(x+p)^2 - 2.3 \cos^2(x-p)$	x=2.31 a=0.99
21	$y = \frac{e^{0.2x} - \lg ctg(x)+2a }{2 \cos^2(x-p) - 7.3} - arctg^2(xp) - \sqrt[3]{\ln(x)} - \sin(x^2+p)$	x=0.57 a=3.28
22	$y = \frac{\cos(x^2+p/2) - 5.6}{e^{a-0.1x} + ctg^2(x)} + \sqrt[3]{ \sin(2x^3) - 8.3 } - \lg( x-a ) \lg^3(x-p)$	x=2.16 a=6.17
23	$y = -\frac{5.4a - \sqrt[3]{(x^3-a)^2}}{\sin(x-p/4)^2 - \cos(x^2-2p)} + 3.1e^{2x} - ctg(x-p/3)^2 - \ln^2(4.1x-a)$	x=2.31 a=0.65
24	$y = \frac{\sin(x-p/6) - a}{tg^2(2x) + \sqrt[3]{xa} - 2.5x} - arctg(x-p/3)^2 - \sqrt[3]{\ln(x-a)} + 2.1e^{3x}$	x=5.66 a=1.77
25	$y = \frac{tg(a+p/2) + \sqrt[5]{x^2a-2.3}}{4.3a} - e^{ax} - \lg^2( \sin(x-p)^3 ) - ctg(x-p/6)$	x=1.34 a=2.65
26	$y = \frac{\sqrt{ x^2-a } - 4.3}{arctg(x^3) + \ln(x-a)} - \frac{\sin^2(x-p/3) + \cos(x^2)}{4ctg(x-a^2)} - 2e^{0.25x}$	x=2.61 a=0.51
27	$y = \frac{ctg^2(x+p/4) - \sqrt[3]{ 2a-x^2 }}{3.2 \ln^2(x)} + arctg(x^2-p) - 3.4e^{2x}$	x=2.23 a=0.37
28	$y = \frac{\ln^3( x-a ) + arctg(x^2-p/3)}{\sqrt[3]{x+a} + ctg(x^2+p/4)} + 2.1e^{ax} - 4.3 \cos^2(a+p/2)$	x=4.56 a=1.45
29	$y = \frac{\cos^2(x-p/3) + \sqrt[3]{4.4+2x-a^2}}{arctg(x^2)e^{3x}} - ctg^2(x-p/6) + \sqrt[3]{\lg( x-a^2 )}$	x=1.75 a=4.83
30	$y = \frac{\cos(x^2+p/6) - \sin(x^2+p/6)}{\sqrt[3]{\lg x-a } + ctg^2(x-p)} + arctg^2(x+p/3) - e^{2x}$	x=5.18 a=1.62

**III рівень**

<b>№ вар.</b>	<b>арифметичний вираз 1</b>	<b>арифметичний вираз 2</b>
<b>1</b>	<b>2</b>	<b>3</b>
1	$f = x^{e^{(2x^3-3y^2)}} \cdot \left( \frac{10 \lg(x^{(y+4x^{-2})})}{\cos\left(\frac{2x^4}{y^{-3}}\right)} + \operatorname{tg}^3(3x^{-2}) \right) - 12 \cdot  \sin(3x - b) - q^6 $	$e = \sqrt[3]{\lg\left x + \frac{a^2}{y}\right } \cdot \frac{ \cos^2(y) \cdot z^{\sqrt{x-3y}} }{\operatorname{tg}(y/x)}$
2	$f = x^{1/7} \cdot \left( \frac{1.5 \cdot a \cdot \lg(x^{2y})}{e^{ax}} + \operatorname{tg}^3(3x^{l-2}) \right)$	$b = 12 \cdot \sqrt{c(\sqrt{y} + x^2)} \cdot  \sin(3x - b) - q^6 $
3	$c = \operatorname{arctg}(x) - \frac{3}{5} \cdot e^{x \cdot y} + 0.5 \frac{ x + y }{(x + y)^b}$	$g = (a^2 - b^3) \sqrt{\frac{\sin(q^4)}{(a + y^3) - 3}} + b^5(1 - c^2)$
4	$m = \sqrt[3]{1 - \frac{a \cdot \operatorname{tg}(x) \cdot \lg(x^4 + y)}{e^{-x}} + 1,81 \cdot z^2}$	$d = \frac{e^{ x-y } \cdot \operatorname{tg}(z)}{\operatorname{arctg}(y) + \sqrt{x}} + \ln(x)$
5	$e = \frac{(\cos(a^2 \cdot x) - \sin^3(w \cdot y))^3}{j \sqrt{ \operatorname{tg}(z) }} + \ln^2(z)$	$r = \sqrt[3]{\left(\frac{e}{\sqrt{a^2 + b^2}}\right)^2} + \left(\frac{1}{3} \cdot x^3 + x \cdot y\right)^l \cdot e^{-0.6 \cdot z}$
6	$f = y^x + \sqrt{ x  + e^y} - \frac{j^3 \cdot \sin^2(2y)}{y + z^2/(a - x)}$	$p = \frac{\ln^3(2 + b^2) \cdot (x + z)^2}{q - (x + y^2)} + \sqrt[5]{(x \cdot z^l + y^b)^2}$
7	$g = \frac{1 + j \cdot \cos(x - y)^3}{ e^{a \cdot z} - 2y/(1 + y^2) } \cdot x^3 + x \cdot \sin(z)$	$u = \frac{(x - z)^3 - (y + z)^2 + \sqrt{ x + y + z }}{b \cdot \lg(\operatorname{tg}(a))}$
<b>1</b>	<b>2</b>	<b>3</b>

8	$s = \left( (j + y^d) \cdot \sqrt[3]{\sin^2(a \cdot z) - \frac{ y-x }{5 \cdot t}} \right)^3$	$m = \left( \sin^2(g \cdot x) + \frac{1}{e^{-0.3 \cdot z}} \right)^{\frac{1}{z^2}} + \operatorname{tg}^k(a - z)$
9	$h = 2 + \frac{x^2}{\sqrt[3]{q^2}} + \frac{ y^3  \cdot (\ln(b \cdot x) + 1) \cdot \sqrt{j}}{\sqrt{3}}$	$p = \frac{\sqrt{ \lg(x^4 \cdot x^2) + \cos^2(y + a) }}{2 \cdot \ln (x-1)^5 }$
10	$k = \ln \left  (y - l \cdot \sqrt{ x }) \cdot \left( x - \frac{a}{z + x^2/4} \right) \right $	$g = \frac{\cos^3 \left( \frac{7p}{2} + x \right)}{1 + \cos(y^d)} + \operatorname{tg}^3 \left( \frac{3p}{2} - y \right)$
11	$l = 0.5x^5 + 3 \cdot \cos^2(x + y) + e^{-0.1y} - \sqrt{ x \cdot y }$	$p = \frac{\lg(a \cdot x^3) - e^{x+y}}{\sqrt[5]{y^q} +  x^3 - \ln(y) }$
12	$k = \left( a^d \left( b^{(a-b)^3} + c^{\sqrt{b-ac^{-2}}} \cdot \ln(c^{-3a}) \right) + 2 \sin^2(c^3) \right)$	$e = \frac{4 \sin^2(g \cdot x)}{1 + \operatorname{tg}^2(y)} + \ln(h \cdot x \cdot y^3)$
13	$m = \sqrt{ -w \cdot \operatorname{tg}(x^3) \cdot \lg^2(a \cdot x^4) / e^{-x} + 1 }$	$f = \operatorname{tg}^2(x) + \frac{\sin^2(2x^4) - 4 \sin^2(a \cdot y)}{2 \sin^2(b \cdot x - 4)} - 1$
14	$n = \sqrt{e^x + \operatorname{tg}(x) + 1} \cdot (\lg(y) + \cos(x \cdot y) + \sqrt[3]{x})$	$l = \left( e^{-0.6 \cdot y + \sqrt{ z }} + \frac{\operatorname{tg}^3(z^j)}{b \cdot \sqrt[4]{ x \cdot z }} \right)^3$
15	$r = \lg^2  1 + 3x^2 - 4x^3  + \frac{\sqrt[4]{ \sin^3(d \cdot x^2) }}{3z}$	$v = \frac{x^2 / (b \cdot (y + z)^3) - \sin(a - y^2)}{z^3 + 3 \cdot \cos^2(z/x)}$

<b>1</b>	<b>2</b>	<b>3</b>
16	$q = \frac{\sqrt[3]{12x^4 - 3x^3 + 6}}{3w^2} - \lg^2(a \cdot z^3)$	$w = \frac{(x/y) \cdot e^{ x-y } + \ln^3(a+z)}{\sin^2(j \cdot y) - \sqrt[5]{ \sin(x^3) }}$
17	$s = \frac{2 \cdot \cos(x-1/j)}{0.85 \sin(d \cdot y)} - \frac{1}{ x^2/(y+x^1) }$	$r = \frac{1}{ x } \cdot \frac{\sqrt[4]{x^3/(x+q)}}{\lg(x \cdot y) - 1 \cdot 10^s} - e^{ y^4/2x }$
18	$t = \frac{a \cdot x \cdot z - y \cdot  x + \sqrt{z} }{d \cdot 2,3 \cdot 10^7 + \sqrt[4]{\lg(4)}}$	$q = \lg \left  (x-1 \cdot \sqrt[4]{ y }) \cdot \left( x - \frac{a}{m+x^3/4} \right) \right $
19	$u = \frac{(x+2y-z+e^{-b})^3 + \sqrt[3]{ x-3z }}{b \cdot \lg(\operatorname{tg}^2(2a))}$	$s = \frac{2 \cdot \cos(x-1/b)}{1,5 \sin^2(j \cdot y)} - \frac{e^{z-a}}{ x^2/(y+x^3) }$
20	$w = \frac{(x/y) \cdot e^{ x-y } + \ln^3(1+x)}{\sin^2(b \cdot y) - (\sin(a \cdot x^2) \cdot \sin(g \cdot y))^2}$	$h = x^y + \sqrt{ z +w^y} - \frac{j^3 \cdot \sin^3(2z)}{x+z^2/(a-y)}$
21	$d = \frac{w^3/(j \cdot (y+z)^2) - \sin^2(a \cdot x) + 5,2^t}{x^3 + 3 \cdot \cos^2(z/x)}$	$k = \ln^3 \left( x^{-\sqrt{ y }} \right) \cdot \frac{(g \cdot \sin(z^2) + e^{2u})}{q \cdot \sqrt[3]{ \sin(b \cdot x) }}$
22	$f = \frac{(z \cdot y + x)^2 - a \cdot \cos(y^2 \cdot z)/e^{z-y}}{e^{x-z} + (\sin^2(3x) - \sin(4y)^2)}$	$u = \frac{\sqrt[4]{-w \cdot \ln(x^3) \cdot \operatorname{tg}^2(b \cdot x^2)/e^{-z} + y}}{2\sqrt{3}}$
23	$b = \frac{\lg^2(2+b^2) \cdot (a-z)^3}{r - ((x+z)^2/2y)} + \sqrt[3]{(x \cdot z^1 + y)^2}$	$j = \frac{e^{ a-y } \cdot \ln(z)}{a \cdot \sin^2(b \cdot y) + \sqrt{ x \cdot y }} + \operatorname{tg}(a \cdot x^3)$
24	$a = \sqrt[3]{1 - \frac{\operatorname{tg}^2(x) \cdot \lg(x^3 + y)}{e^{-z}}} + 2,3 \cdot x^{d+0,5}$	$t = x^y + \sqrt{ z +e^{-y}} - \frac{w \cdot \sin^2(b \cdot y)}{x+z^2/(a-y)}$

<b>1</b>	<b>2</b>	<b>3</b>
25	$v = \sqrt[5]{\frac{q \cdot x^2}{\sqrt{a^2 + b^2}}} - \left(\frac{1}{7} \cdot x^3 + z \cdot y\right)^g \cdot e^{-0.6 \cdot x}$	$g = \frac{e^{ x-y } \cdot \sin(a \cdot z^2)}{\operatorname{tg}^3(l \cdot y) + \sqrt[3]{z^2}} + \ln(x)$
26	$p = \frac{4,5 \cdot \lg^3(y + g^2) \cdot (z + x)^2}{j - ((x + z)^3 / (y + z))} + \sqrt[5]{(y^q + z^b)^3}$	$s = \frac{z + q \cdot \sin(x + y)^3}{ e^j - 3y / (1 + z^2) } \cdot y^2 + \cos^2(a \cdot z)$
27	$m = \frac{e^{x \cdot z}}{a \cdot y^2} + 0,25 \frac{ x - y }{(x + z)^d}$	$q = \sqrt{e^{-z} + \operatorname{tg}^2(a \cdot x^3)} \cdot \left( \lg^2(x) + \frac{\cos(x^2 \cdot z^2)}{2\sqrt{5}} \right)$
28	$w = e^{(y^3 \cdot z)} + \frac{(\operatorname{tg}(q^2) - \sin^3(b \cdot y))^3}{l \sqrt{ \lg(x) }}$	$f = 2 \frac{ x^3 }{\sqrt{2}} + \frac{g}{\sqrt[3]{q^2}} + \frac{y^2 \cdot (\ln(l \cdot z) + 1)}{\sqrt{5}}$
29	$b = \frac{\sqrt[5]{y^2 / (x + z \cdot g)}}{\lg(d \cdot y) - l \cdot 10^s} - e^{ z^3 / (x - y) }$	$s = \ln^3 2 - 5x^3  + \frac{\sqrt[5]{ \cos^2(b \cdot z^2) }}{3y}$
30	$r_k = \frac{\cos^2(a \cdot x)}{3\sqrt{2}} + \frac{4 \operatorname{tg}(j \cdot z^3)}{1 + \ln(q \cdot z + y^2)}$	$d = \left( \frac{\sqrt[5]{\sin^2(l \cdot z)}}{\sqrt{3}} \cdot x^g - \frac{ y - x }{3 \cdot k} \right)^3$

## Додаток 11. Поняття про алгоритм і етапи розробки програм

При вирішенні задач на комп'ютері програміст виконує постановку задачі, розробляє алгоритм її рішення, програмує і проводить аналіз отриманих результатів, а персональний комп'ютер – обробляє інформацію відповідно до розробленого алгоритму.

При постановці задачі необхідно добре представляти наочну область завдання і чітко визначити мету завдання, дати словесний опис змісту завдання і запропонувати загальний підхід до її рішення. Після цього необхідно обрати метод її розв'язання – створити таку модель вирішуваного завдання, яка могла би бути реалізована в комп'ютері. Проте можливо, що може бути декілька методів рішення, і тоді необхідно вибрати кращий. Далі виконується алгоритмізація завдання. На основі математичного опису необхідно розробити алгоритм рішення.

**Алгоритм** – це чітка, однозначна послідовність кінцевого числа дій, що необхідна для вирішення будь-якого завдання даного типу (класу) і приводить до шуканого результату.

При складанні і записі алгоритму необхідно забезпечити, щоб він задовільняв наступним **вимогам**:

- однозначність алгоритму – єдиність тлумачення виконавцем правил виконання дій і порядку їх виконання (Наприклад, з'їсти морозиво, а не – чи з'їсти морозиво, чи випити води);
- скінченність алгоритму – обов'язковість завершення кожної з дій, складових алгоритма, і завершеність алгоритму в цілому;
- результативність алгоритму – передбачає, що виконання алгоритму повинне завершитися отриманням шуканих результатів. При цьому, іноді в якості вимоги включають і таку властивість алгоритму, як його правильність – це здатність алгоритму давати правильні результати вирішення поставленої задачі;
- масовість – можливість застосування даного алгоритму для вирішення цілого класу задач, що відповідають загальній постановці задачі.
- Правильність алгоритму – здатність алгоритму давати правильні результати вирішення поставлених задач.

### **Форми запису алгоритму:**

1. Текстуальна – алгоритми записують за допомогою загальноприйнятих математичних символів і виразів, що супроводжуються пояснюючим текстом. Можливий будь-який ступінь деталізації алгоритму, але такі алгоритми не наочні і погано формалізуються. Як приклади таких алгоритмів можна навести інструкції по використанню технічних пристроїв.

2. Графічна – алгоритм представляється у вигляді графічних фігур (блоків), кожний з яких має певне смислове значення і є частиною рішення задачі. Таку схему називають блок-схемою. Вигляд графічних блоків (ГОСТ 19003.80) наведений в таблиці 1.

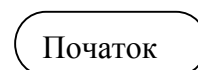
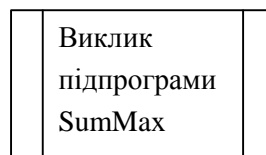
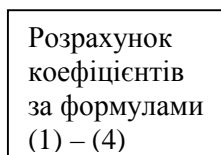
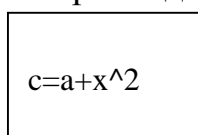


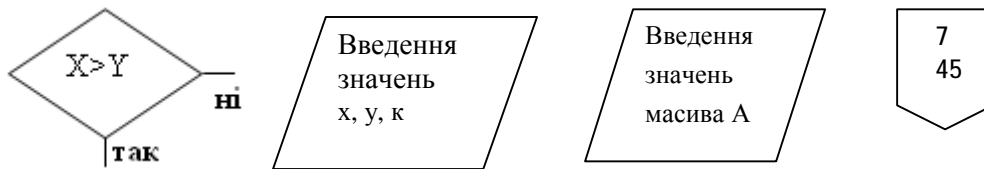
Таблиця 11.1  
Основні блоки для побудови блок-схем

№	Позначення	Функції блока
1		Обчислювальний блок для позначення виконання операції або групи операцій. (Виконання операції або групи операцій, в результаті яких змінюється значення(я); форма уявлення або розташування даних (розрахунковий блок ).
2		Блок введення-виведення. (Введення інформації (даних) і виведення інформації (даних)).
3		Блок виведення інформації на паперовий носій
4		Логічний блок (вибір напрямку виконання алгоритму залежно від деяких умов
5		Блок виклику підпрограми (Використання раніше створених і окремо описаних програм)
6		Блок початку, кінця програми
7		З'єднувач (зв'язок між перерваними лініями в межах однієї сторінки)
8		Міжсторінковий з'єднувач (Зв'язок між частинами алгоритму, розташованими на різних сторінках).
9	-----[	Коментар (Зв'язок між елементом схеми і поясненням).

Розміри **a** і **b** зв'язані співвідношенням  $b = 1,5a$  (розмір **a** починається зі значення  $a = 10\text{мм}$ )

Приклади:





На прикладі міжсторінковий з'єднувач призначений для з'єднання блоку, після якого він розташований з блоком, що знаходиться на сторінці 7 і має номер 45.

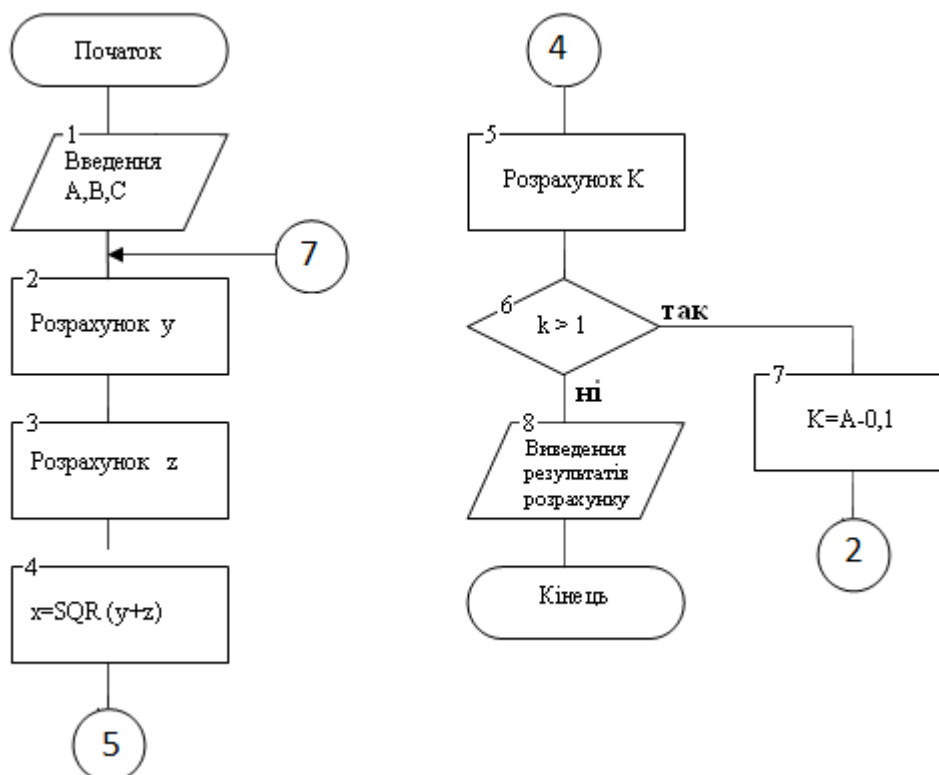
Блоки розташовуються в блок-схемі згідно логіці алгоритму. Блок-схема починається з блоку із словом “Початок” і закінчується блоком “Кінець”. Всі блоки послідовно нумеруються. Номер проставляється в розриві лінії в лівому верхньому куті кожного блоку (окрім блоків “Початок” і “Кінець”).

Блоки з'єднуються між собою горизонтальними і/або вертикальними лініями (лінії, розташовані під кутом, не допускаються) із стрілками на кінці, що відображають напрямок обчислювального процесу. При цьому стрілки, направлені “вниз” і “вправо” можуть бути опущені. Перетину стрілок бажано не допускати. При неможливості провести нерозривну стрілку між двома блоками, використовуються або “з'єднувач”, – якщо блоки розташовані на одній сторінці, або “міжсторінковий з'єднувач”, – якщо блоки розташовані на різних сторінках. При використанні “з'єднувача” на одному кінці розриву в “крузі” проставляється номер наступного блоку (на який передається виконання), а на іншому – номер попереднього блоку (з якого передається управління). Коли необхідно з'єднати стрілкою блоки на різних сторінках, в “міжсторінковому з'єднувачі” аналогічним чином проставляються не тільки номери подальшого і попереднього блоків, але і номери сторінок, на яких вони розташовані (як показано у прикладах – номер сторінки зверху і номер блока знизу).

У кожному “розрахунковому блоці” (див. приклади) допускається написання не більш однієї розрахункової формули, але можливе “об'єднання” декількох формул в одному блоці під загальною назвою, якщо вони об'єднані загальним сенсом або однаковим призначенням.

Алгоритм, що реалізується у вигляді окремого програмного модуля, є автономним і зображається у вигляді окремої блок-схеми. Зв'язок між різними модулями однієї програми і, відповідно, – між блок-схемами алгоритмів, які в них реалізовані, здійснюється за допомогою “блоку підпрограм” (див. таблицю і приклади), який відповідає операторові звернення до підпрограми в “головному модулі”.

Приклад.



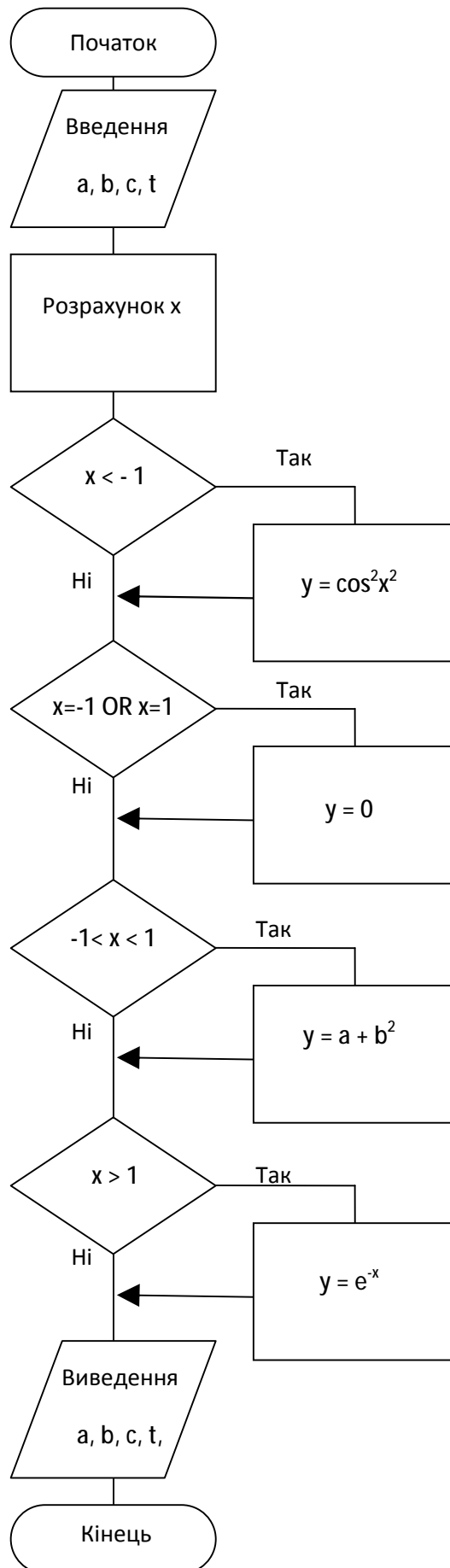
Після розробки алгоритма рішення виконується програмування задачі. Розробка програми дозволяє реалізувати розроблений алгоритм на обраній мові програмування.

Після розробки програми виконується її введення за допомогою редактора і запис на диск. Також підбирають дані для контрольних розрахунків.

Наступний етап – це тестування і відладка програми (правильне виконання алгоритму на ЕОМ, пошук і виключення помилок). Для складних програм цей етап часто вимагає значно більше часу і сил, чим написання первинного тексту програми. Відладка програми – складний і нестандартний процес, який полягає в тому, щоб протестувати програму на контрольних прикладах. Контрольні приклади прагнуть вибрати так, щоб при роботі з ними програма пройшла всі основні шляхи алгоритму, оскільки на кожному з шляхів можуть зустрітися свої помилки (програма може «зациклитися», може дати безглуздий результат тощо). Дуже складні програми відладжують окремими фрагментами.

І тільки потім настає використання відладженої програми. Але на початковому етапі її використання робиться детальний аналіз результатів, що отримані для початкових даних, потрібних по умові задачі. Отримані результати аналізуються постановником задачі, і на підставі цього аналізу виробляються відповідні рішення, рекомендації, висновки.

## Додаток 12. Блок - схема алгоритма визначення Y



### Додаток 13. Завдання до лабораторної роботи №3

#### І рівень

Розробити програму для обчислення значення арифметичного виразу при заданих значеннях змінних з використанням розгалуженої алгоритмічної структури.

№ вар.	y	x	a	b	c	t
1	2	3	4	5	6	7
1	$y = \begin{cases}  x  & \text{при } x < 0 \\ -x & \text{при } 0 \leq x < 2 \\ \sqrt{x} & \text{при } x \geq 2 \end{cases}$	$x = c \cdot \sqrt{\ln \frac{t-b}{a} + 1}$	2,5	0,1	3	1,6
2	$y = \begin{cases} -x & \text{при } x < 0 \\ x^2 & \text{при } 0 \leq x < 0,5 \\ \sqrt[3]{x} & \text{при } x \geq 0,5 \end{cases}$	$x = b \cdot \sqrt{\ln \frac{t-a}{c} + 1}$	1,2	0,43	1,5	2
3	$y = \begin{cases} x^2 & \text{при } x < 1 \\ \sqrt[3]{x} & \text{при } 1 \leq x < 2 \\  x  & \text{при } x \geq 2 \end{cases}$	$x = a \cdot \sqrt{\ln \frac{t-c}{b} + 1}$	0,95	-1	2	0,8
4	$y = \begin{cases}  x  & \text{при } x < -5 \\ -x & \text{при } -5 \leq x < 0 \\ \sqrt{x} & \text{при } x \geq 0 \end{cases}$	$x = c \cdot \log_a t - b$	7	-0,4	6	0,2
5	$y = \begin{cases} -x & \text{при } x < 22 \\ x^2 & \text{при } x = 22 \\ \sqrt[3]{x} & \text{при } x > 22 \end{cases}$	$x = b \cdot \log_a t - c$	5	3	25	5
6	$y = \begin{cases} x^2 & \text{при } x < 0,5 \\ \sqrt[3]{x} & \text{при } 0,5 \leq x < 1 \\  x  & \text{при } x \geq 1 \end{cases}$	$x = a \cdot \log_b t - c$	-3	0,2	2	4
7	$y = \begin{cases} -x & \text{при } x < 0 \\ x^2 & \text{при } 0 \leq x < 0,5 \\ \sqrt[3]{x} & \text{при } x \geq 0,5 \end{cases}$	$x = b \cdot \lg^2 \frac{c-a}{t}$	0,1	25	4	5

1	2	3	4	5	6	7
8	$y = \begin{cases} -x & \text{при } x < 2 \\ x^2 & \text{при } 2 \leq x < 3 \\ \sqrt[3]{x} & \text{при } x \geq 3 \end{cases}$	$x = a \cdot \lg^2 \frac{b}{t} + c$	7,1	0,8	-0,5	0,2
9	$y = \begin{cases} x^2 & \text{при } x < -1 \\ \sqrt[3]{x} & \text{при } -1 \leq x < 0 \\  x  & \text{при } x \geq 0 \end{cases}$	$x = c \cdot \lg^2 \frac{t}{a} + b$	-0,5	-0,4	-4,4	-1
10	$y = \begin{cases} -x & \text{при } x < 0 \\ x^2 & \text{при } 0 \leq x < 0,5 \\ \sqrt[3]{x} & \text{при } x \geq 0,5 \end{cases}$	$x = \frac{c \cdot e^{-t}}{b-a}$	0,85	2,2	1	1,2
11	$y = \begin{cases} x^2 & \text{при } x < -1 \\ \sqrt[3]{x} & \text{при } -1 \leq x < 0 \\  x  & \text{при } x \geq 0 \end{cases}$	$x = \frac{b \cdot e^{-t}}{a-c}$	-1,1	0,8	0,33	0,05
12	$y = \begin{cases} x^2 & \text{при } x < -0,5 \\ \sqrt[3]{x} & \text{при } -0,5 \leq x < 0 \\  x  & \text{при } x \geq 0 \end{cases}$	$x = \frac{a \cdot e^{-t}}{b-c}$	0,2	-1,1	0,1	0,1
13	$y = \begin{cases}  x  & \text{при } x < 5 \\ -x & \text{при } 5 \leq x < 10 \\ \sqrt{x} & \text{при } x \geq 10 \end{cases}$	$x = \frac{c \cdot t g^2 t}{b \cdot a}$	1,1	0,1	9,2	-0,3
14	$y = \begin{cases} -x & \text{при } x < 0 \\ x^2 & \text{при } 0 \leq x < 0,5 \\ \sqrt[3]{x} & \text{при } x \geq 0,5 \end{cases}$	$x = \frac{b \cdot \sin^2 t}{c \cdot a}$	0,8	0,5	-2	-0,65
15	$y = \begin{cases}  x  & \text{при } x < -2 \\ -x & \text{при } -2 \leq x < -1 \\ \sqrt{x} & \text{при } x \geq -1 \end{cases}$	$x = \frac{a \cdot \cos^2 t}{b \cdot c}$	0,6	-0,1	4	-0,6
16	$y = \begin{cases}  x  & \text{при } x < 0 \\ -x & \text{при } 0 \leq x < 2 \\ \sqrt{x} & \text{при } x \geq 2 \end{cases}$	$x = c \sqrt{\ln \frac{t-b}{a} + 1}$	2,5	0,1	3	1,6
17	$y = \begin{cases}  x  & \text{при } x < 0 \\ -x & \text{при } 0 \leq x < 2 \\ \sqrt{x} & \text{при } x \geq 2 \end{cases}$	$x = c \sqrt{\ln \frac{t-b}{a} + 1}$	2,5	0,1	3	1,6

<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>
18	$y = \begin{cases} -x & \text{при } x < 0 \\ x^2 & \text{при } 0 \leq x < 0,5 \\ \sqrt[3]{x} & \text{при } x \geq 0,5 \end{cases}$	$x = b \sqrt{\ln \frac{t-a}{c} + 1}$	1,2	0,43	1,5	2
19	$y = \begin{cases} x^2 & \text{при } x < 1 \\ \sqrt[3]{x} & \text{при } 1 \leq x < 2 \\  x  & \text{при } x \geq 2 \end{cases}$	$x = a \sqrt{\ln \frac{t-c}{b} + 1}$	0,95	-1	2	0,8
20	$y = \begin{cases}  x  & \text{при } x < -5 \\ -x & \text{при } -5 \leq x < 0 \\ \sqrt{x} & \text{при } x \geq 0 \end{cases}$	$x = c \cdot \log_a t - b$	7	-0,4	6	0,2

## II рівень

1. Розробити алгоритм і програму для визначення залежності:

$$\text{де } x = \frac{\operatorname{tg} t}{\sin t + \cos t} + a * b; \quad b = 4; \\ a = 0,2; \quad t = 1,2.$$

$$Y = \begin{cases} \sin^4(x-5) & \text{при } x < 0 \\ x^3 - \sqrt{x} & \text{при } 0 \leq x < 10 \\ e^{-x} + |x| & \text{при } x \geq 10 \end{cases}$$

2. Розробити алгоритм і програму для визначення залежності:

$$\text{де } x = a * \cos t^2 - b * t; \\ a = 1; \quad b = 0,2; \quad t = 0,4.$$

$$Y = \begin{cases} 1/x + 1/(3 * x^2) & \text{при } x < 5 \\ 7x^{\ln x} & \text{при } 5 \leq x < 7 \\ 2\sin x^4 + \cos x & \text{при } x \geq 7 \end{cases}$$

3. Розробити алгоритм і програму для визначення залежності:

$$\text{де } x = b * t^2 + a; \quad a = -3; \\ b = 4; \quad t = 3,5.$$

$$Y = \begin{cases} e^{-2x} + x & \text{при } x < 0 \\ x^4 + 5x & \text{при } x = 0 \\ |\sin x + \operatorname{tg} x| & \text{при } x > 0 \end{cases}$$

4. Розробити алгоритм і програму для визначення залежності:

$$x = 2,5t^2 - a / t; \quad a = 5; \\ t = 7,8.$$

$$Y = \begin{cases} 5^{\operatorname{tg} x} - 4 & \text{при } x < 2 \\ 1 & \text{при } 2 \leq x < 3 \\ \frac{1}{(2\sin x)^x} & \text{при } x \geq 3 \end{cases}$$

5. Розробити алгоритм і програму для визначення залежності:

$$\text{де } x = a * \sin t^b - a; \\ a = 0,2; \quad b = 0,4; \quad t = 0,3.$$

$$Y = \begin{cases} \cos 5x - x^{2,3} & \text{при } x < -2 \\ |\ln x^4 - 5| & \text{при } -2 \leq x < 2 \\ \frac{1}{\operatorname{tg} x + \operatorname{ctg} 2x} & \text{при } x \geq 2 \end{cases}$$

6. Розробити алгоритм і програму для визначення залежності:

$$\text{де } x = \ln t^a + b; \quad a = 3; \\ b = 0,3; \quad t = 2,7.$$

$$Y = \begin{cases} 10^{\sin x - \cos 2x} & \text{при } x < 3 \\ 5 & \text{при } 3 \leq x < 9 \\ |x^6 - 7\sqrt{x}| & \text{при } x \geq 9 \end{cases}$$

7. Розробити алгоритм і програму для визначення залежності:

$$\text{де } x = t^a + b * t^b; \quad a = 0,3; \\ b = 0,4; \quad t = 0,2.$$

$$Y = \begin{cases} x^2 + 4 & \text{при } x < -2 \\ |\sin x - 2^{-x}| & \text{при } -2 \leq x < 10 \\ \log_5 x^2 & \text{при } x \geq 10 \end{cases}$$



8. Розробити алгоритм і програму для визначення залежності:  
де  $x = t^{(a * b - b)}$ ;  $a = 0,3$ ;  
 $b = 1,1$ ;  $t = 0,7$ .
- $$Y = \begin{cases} \lg(3x - x^2) & \text{при } 0 < x < 2 \\ \sin(3x^3 + 4) & \text{при } 2 \leq x < 3 \\ \frac{1}{x^4 + 5} & \text{при } x \geq 3 \end{cases}$$
9. Розробити алгоритм і програму для визначення залежності:  
де  $x = a * \sin^2 t - b * \cos t^3$ ;  
 $a = 1,3$ ;  $b = 1,5$ ;  $t = 1,8$ .
- $$y = \begin{cases} |\sin x^2 - b * \cos^2 x^3| & \text{при } x < 0 \\ \lg|x^2 - 5.1| + l^{-x^2+1} & \text{при } 0 \leq x < 1 \\ \operatorname{tg}^2(x-2) - 2.5 & \text{при } x \geq 1 \end{cases}$$
10. Розробити алгоритм і програму для визначення залежності:  
де  $x = a * \ln t$ ;  $a = 1.2$ ;  
 $b = 2,1$ ;  $t = 2.8$ .
- $$y = \begin{cases} \ln|(x-1.5) * \sin x| + 2 * ax^2 & \text{при } x < 1 \\ \operatorname{tg} x - b * \cos^2 x^3 & \text{при } 1 \leq x \leq 2.1 \\ l^{-x^2+1.5x} + \ln x & \text{при } x > 2.1 \end{cases}$$
11. Розробити алгоритм і програму для визначення залежності:  
де  $x = a * \sin t - b * \cos(2t)$ ;  
 $a = 2,5$ ;  $b = 0,8$ ;  $t = 2,93$ .
- $$y = \begin{cases} \ln|x-3| + 1,4 & \text{при } x < 0 \\ \operatorname{tg}(x-1) + a * \cos^2 x^3 & \text{при } 0 \leq x < 1 \\ \ln(x^2 + 1) - e^{-2x} & \text{при } x \geq 1 \end{cases}$$
12. Розробити алгоритм і програму для визначення залежності:  
де  $x = e^{-\sin t} + \cos t$ ;  
 $t = 1,37$ .
- $$Z = \begin{cases} 3 & \text{при } x = 1,5 \\ \ln x & \text{при } x > 2 \\ \sqrt{|x|} & \text{при } x \leq -3 \\ |x + \operatorname{tg} x^2| & \text{при } -3 < x \leq 2 \end{cases}$$
13. Розробити алгоритм і програму для визначення залежності:  
де  $x = a * \sin 3 t + 1.5$ ;  
 $a = 1.2$ ;  $b = 2.3$ ;  $t = 0.8$ .
- $$Z = \begin{cases} e^{-x} + \operatorname{tg} x & \text{при } 2 \leq x \leq 2,5 \\ \ln x^2 + b * \ln x^4 & \text{при } x > 2 \\ \cos^2 x + b * \cos^3 x & \text{при } x < 2 \end{cases}$$
14. Розробити алгоритм і програму для визначення залежності:  
де  $x = a * t - \operatorname{ctg}(b * t)$ ;  
 $a = 0,2$ ;  $b = -0,4$ ;  $t = 4,7$ .
- $$Y = \begin{cases} \ln|\sin^3 x^3| - 1 & \text{при } x \leq 2 \\ 10^{-2x^2 + \sin x} & \text{при } 2 < x \leq 2,3 \\ \cos x + 2 \sin x^2 & \text{при } x > 2,3 \end{cases}$$
15. Розробити алгоритм і програму для визначення залежності:  
де  $x = a * \cos^2 t - b * t$ ;
- $$Y = \begin{cases} \lg(x^2 - 2x) & \text{при } x < 0 \\ \ln(x + 4) & \text{при } 0 \leq x < 10 \\ \lg(5 * x) - \log_5 x & \text{при } x \geq 10 \end{cases}$$

$$a = 0,5; \quad b = 6; \quad t = 1.$$

16. Розробити алгоритм і програму для визначення залежності:

$$\text{де } x = a * t + b; \quad a = 5; \\ b = 0,1; \quad t = 4.$$

$$Y = \begin{cases} |tgx - \cos^3 2x| & \text{при } x < 0 \\ x^2 + x^3 - 5x^4 & \text{при } 0 \leq x < 5 \\ x + \lg x & \text{при } x > 5 \end{cases}$$

17. Розробити алгоритм і програму для визначення залежності:

$$\text{де } x = \exp(t^4 - a*t + 1) - 2*b; \\ a = 4,5; \quad b = 6; \quad t = 4.$$

$$Y = \begin{cases} |x| & \text{при } x < 0 \\ 0 & \text{при } x = 0 \\ -x & \text{при } x > 0 \end{cases}$$

18. Розробити алгоритм і програму для визначення залежності:

$$\text{де } x = t^2 + a * \cos t; \quad a = 2; \\ t = 0.$$

$$Y = \begin{cases} |\sin x^2 - \cos^2 x| & \text{при } x < 1 \\ \cos 2x + \sin 2x & \text{при } 1 \leq x < 2 \\ |x - 5| & \text{при } x \geq 2 \end{cases}$$

19. Розробити алгоритм і програму для визначення залежності:

$$\text{де } x = t^{\frac{a}{b}} - 2*a/(b-0.8); \quad t = 4; \\ a = 3; \quad b = 2.$$

$$Y = \begin{cases} |\log_7 7x| & \text{при } x < 0 \\ 5x + 4 & \text{при } 0 \leq x < 15 \\ |-x^{\sin x + 3}| & \text{при } x \geq 15 \end{cases}$$

20. Розробити алгоритм і програму для визначення залежності:

$$\text{де } x = t^3 - a * t + b; \quad a = 2; \\ b = 4; \quad t = 1.$$

$$Y = \begin{cases} e^{x^2} & \text{при } x < -3 \\ |x| & \text{при } -3 \leq x < 3 \\ \sin x - \cos 2x & \text{при } x \geq 3 \end{cases}$$

21. Розробити алгоритм і програму для визначення залежності:

$$\text{де } x = \cos(\ln t^a + b); \quad a = 0,3; \\ b = 2,1; \quad t = 1,7.$$

$$Y = \begin{cases} -e^{-x^2 + 2x} & \text{при } x < 3 \\ \sin x^4 + ctg 2x & \text{при } 3 \leq x < 4 \\ 5 & \text{при } x \geq 4 \end{cases}$$

22. Розробити алгоритм і програму для визначення залежності:

$$\text{де } x = a * \sin t^b - a * t - b; \\ a = 2; \quad b = 3; \quad t = 1,5.$$

$$Y = \begin{cases} \sin|x| & \text{при } x < 10 \\ 10^x + \lg 3x^2 & \text{при } x = 10 \\ \cos|x| & \text{при } x > 10 \end{cases}$$

23. Розробити алгоритм і програму для визначення залежності:  
де  $x = 4t^3 - a * t + t^{-b/a}$ ;  
 $a = 1,5$ ;  $b = -2$ ;  $t = -0,5$ .
- $$Y = \begin{cases} \operatorname{tg}^2|x^2 - 4x + 3| & \text{при } x < 2 \\ |2x + 5| & \text{при } 2 \leq x < 5 \\ 4 & \text{при } x \geq 5 \end{cases}$$
24. Розробити алгоритм і програму для визначення залежності:  
де  $x = a * t^3 - b * t$ ;  $a = 1,5$ ;  
 $b = 3$ ;  $t = 2,3$ .
- $$Y = \begin{cases} \cos^2(x + 4) & \text{при } x < 2 \\ |\sin x - e^x| & \text{при } 2 \leq x < 4 \\ \ln 3x & \text{при } x \geq 4 \end{cases}$$
25. Розробити алгоритм і програму для визначення залежності:  
де  $x = a * t - b * t^2$ ;  $a = 1,5$ ;  
 $b = 3$ ;  $t = 2,3$ .
- $$Y = \begin{cases} e^{-x} + \operatorname{ctgx} \cdot \operatorname{tg} 2x & \text{при } x < -10 \\ |x^3| & \text{при } -10 \leq x < -1 \\ x^4 + 10^x & \text{при } x \geq -1 \end{cases}$$

26. Визначити коефіцієнт тепловіддачі  $a$  (Вт/м<sup>2</sup>·град) між стінкою реактора і шаром твердих частинок каталізатора (нерухомих шар):

$$a = \begin{cases} \exp\left[\ln\left(0,813 \cdot \frac{K_g}{D_t}\right) + 0,9 \ln\left(\frac{D_p \cdot G}{m}\right) - \frac{G \cdot D_p}{D_t}\right] & \text{при } \frac{D_p}{D_t} < 0,35, \\ \exp\left[\ln\left(0,125 \cdot \frac{K_g}{D_t}\right) + 0,75 \ln\left(\frac{D_p \cdot G}{m}\right)\right] & \text{при } \frac{D_p}{D_t} \geq 0,35. \end{cases}$$

при наступних значеннях вихідних даних:

$D_t = 0,14$  м – діаметр трубчатого реактора;  $D_p = 0,177 \cdot 10^{-3}$  м – діаметр частинки каталізатора;  $G = 771$  кг/(м<sup>2</sup>·ч) – масова швидкість потоку газу крізь шар каталізатора;  $m = 0,0647$  кг/(м·ч) – в'язкість;  $K_g = 0,0947$  Вт/(м<sup>2</sup>·град) – коефіцієнт теплопровідності.

27. Визначити коефіцієнт тепловіддачі від  $\alpha$  метилового спирту до стінки труби при поперечному обтіканні пучка гладких труб:  $a = \frac{Nu \cdot I}{H}$

де критерій Нуссельта визначається згідно залежності:

$$Nu = \begin{cases} 0,56 \operatorname{Re}^{0,5} \operatorname{Pr}^{0,36} \left(\frac{\operatorname{Pr}}{\operatorname{Pr}_{CT}}\right)^{0,25} & \text{при } \operatorname{Re} < 1000 \\ 0,22 \operatorname{Re}^{0,65} \operatorname{Pr}^{0,36} \left(\frac{\operatorname{Pr}}{\operatorname{Pr}_{CT}}\right)^{0,25} & \text{при } \operatorname{Re} \geq 1000 \end{cases},$$

$\operatorname{Re} = \frac{w d_{\exists} r}{m}$  – критерій Рейнольдса;  $\operatorname{Pr} = 4$  – критерій Прандтля;  $H = 2,5$  м –

висота теплообмінника;  $r = 984$  кг/м<sup>3</sup>;  $I = 0,679$  Вт/(м·град);

$m = 0,54 \cdot 10^{-3}$  кг/(м·с);  $d_3 = 0,015$  м – еквівалентний діаметр;  $w = 0,25$  м/с – швидкість течії рідини.

28. Визначити коефіцієнт тепловіддачі  $a$  від чотирьоххлористого вуглецю до стінки труби при стіканні його тонкою плівкою по внутрішній поверхні труб в вертикальному кожухотрубному теплообміннику:

$a = \frac{Nu \cdot \lambda}{H}$ , де критерій Нуссельта визначається згідно залежності:

$$Nu = \begin{cases} 0,01(Ga \cdot Pr \cdot Re)^{1/3} & \text{при } Re > 2000, \\ 0,67(Ga^2 \cdot Pr^3 \cdot Re)^{1/9} & \text{при } Re \leq 2000; \end{cases}$$

$$Ga = \frac{H^3 r^2 g}{m^2} \text{ – критерій Галілея, } Re = \frac{w d_3 r}{m} \text{ – критерій Рейнольдса; } Pr$$

$= 6$  – критерій Прандтля;  $\lambda = 0,109$  Вт/(м·град);

$m = 0,84 \cdot 10^{-3}$  кг/(м·с);  $r = 1630$  кг/м<sup>3</sup>;  $H = 1,25$  м – висота теплообмінника;  $g = 9,81$  м/с<sup>2</sup>;  $d_3 = 0,027$  м – еквівалентний діаметр плівки;  $w = 0,103$  м/с – швидкість течії рідини.

29. Розробити алгоритм і програму для визначення втрат напору  $\Delta r_\Gamma$  (Н/м<sup>2</sup>) при проходженні газу крізь суху насадку:

$$\Delta r_\Gamma = I \frac{l}{d_{екв}} \times \frac{w^2 r_\Gamma}{2e^2}, \text{ де } I \text{ – коефіцієнт тертя, що в залежності від критерію}$$

Рейнольдса визначається з співвідношень:

$$I = \begin{cases} \exp(\ln 70 - 0,45 \times \ln Re); & \text{при } Re < 400 \\ \exp(\ln 16,5 - 0,2 \times \ln Re); & \text{при } Re \geq 400 \end{cases}$$

Визначити втрати потужності при проходженні газу крізь шар каталізатора при наступних значеннях:

$$Re = \frac{4 \times w \times r_\Gamma}{s \times m_\Gamma}; \quad G = 771; \quad m_\Gamma = 0,0647; \quad l = 5; \quad d = 10;$$

$$w = \frac{4 \times G}{p \times d^2 \times 3600}; \quad d_{екв} = \frac{4 \cdot e}{s}; \quad r_\Gamma = 1; \quad s = 15; \quad e = 0,667.$$

30. Визначити значення критерію Рейнольдса для частинок кулястої форми, що осаджуються в полі сили тяжіння:

$$\begin{cases} Re = 0,056 Ar & \text{при } Ar \leq 36 \\ Re = 0,152 Ar^{0,714} & \text{при } 36 < Ar \leq 84000, \\ Re = 1,740 Ar^{0,5} & \text{при } Ar > 8400 \end{cases}$$

$$\text{де } Ar = \frac{g d_{min}^3}{n^2} \left( \frac{r_\Gamma - r_2}{r_2} \right) \text{ – критерій Архімеда; } r_\Gamma = 1300 \text{ кг/м}^3; \quad r_2 = 0,571 \text{ кг/м}^3;$$

$$g = 9,81 \text{ м/с}^2; \quad d_{min} = 0,012 \text{ мм}; \quad n_\Gamma = 53,1 \cdot 10^{-6} \text{ м}^2/\text{с}.$$

### III рівень

Розробити програму для обчислення значення арифметичного виразу при заданих значеннях змінних з використанням розгалуженої алгоритмічної структури. Підібрати значення змінних, які забезпечують виконання всіх гілок структури.

$$\begin{array}{l}
 1 \quad y = \begin{cases} \operatorname{arctg}(x - p / 6), & bx^2 > 4 \\ x \lg(|b|), & 1 < bx^2 \leq 4 \\ |b - x^3|, & bx^2 \leq 1 \end{cases} \\
 2 \quad y = \begin{cases} c^3 + x, & |cx| \leq 1.5 \\ \cos(c - p), & 1.5 < |cx| \leq p \\ |x|e^c, & |cx| > p \end{cases} \\
 3 \quad y = \begin{cases} \sin(t^2 x - p), & xt^2 > 4.3 \\ e^{-2x}, & 1 < xt^2 \leq 4.3 \\ t \cos^2(x + p), & xt^2 \leq 1 \end{cases} \\
 4 \quad y = \begin{cases} r^3 \sqrt{|x|}, & r - x \leq 2.1 \\ \cos(rx - p), & 2.1 < r - x \leq 5 \\ x \operatorname{ctg}(rx + p), & r - x > 5 \end{cases} \\
 5 \quad y = \begin{cases} k + e^{2x}, & kx^2 \leq 0.5 \\ \operatorname{ctg}(kx - p), & 0.5 < kx^2 < p \\ x/2|k|, & kx^2 \geq p \end{cases} \\
 6 \quad y = \begin{cases} \sqrt[3]{|qx|}, & qx < e \\ \cos^2(q - x), & e \leq qx \leq p \\ \operatorname{arctg}(q/2p), & qx > p \end{cases} \\
 7 \quad y = \begin{cases} b \sin(x + p), & bx^3 \leq -5 \\ \operatorname{ctg}(b/2x), & -5 < bx^3 < 5 \\ \ln^2(|b - x|), & bx^3 \geq 5 \end{cases} \\
 8 \quad y = \begin{cases} \sqrt{x} + |p|, & p/x > 1 \\ p/x, & -1 < p/x \leq 1 \\ \operatorname{tg}(|x| - p), & p/x \leq -1 \end{cases} \\
 9 \quad y = \begin{cases} \lg(|zx|), & zx \leq p \\ -ze^{3x}, & p < zx \leq 9 \\ \operatorname{ctg}(x/2p), & zx > 9 \end{cases} \\
 10 \quad y = \begin{cases} \ln(|x^2 - b|), & |bx| \leq 2 \\ b \operatorname{ctg}^3(x - p), & 2 < |bx| \leq 6 \\ \operatorname{arctg}(x) - e, & |bx| > 6 \end{cases}
 \end{array}$$

$$\begin{array}{l}
11 \quad y = \begin{cases} \lg(nx) + e^{3x}, & n^2 x^3 > 7 \\ n^2 + 2x, & 2 < n^2 x^3 \leq 7 \\ \cos(nx - p), & n^2 x^3 \leq 2 \end{cases} \\
12 \quad y = \begin{cases} \sin^2(x - p), & |ux| > 4 \\ |u + x|/2x, & 1 < |ux| \leq 4 \\ e^{u-x}, & |ux| \leq 1 \end{cases} \\
13 \quad y = \begin{cases} |x|/2t^2, & tx^2 > 4 \\ \text{ctg}(x - p), & 1 < tx^2 \leq 4 \\ e^{3x}, & tx^2 \leq 1 \end{cases} \\
14 \quad y = \begin{cases} \lg^2(|ax|), & a/2 + 1 \leq 1 \\ \cos(x^2 - p), & 1 < a/2 + 1 \leq 6 \\ 2a \cdot \text{arctg}(x), & a/2 + 1 > 6 \end{cases} \\
15 \quad y = \begin{cases} \text{tg}(x + p/2d), & \sqrt[3]{dx} \leq 1 \\ \lg(|x^3|), & 1 < \sqrt[3]{dx} \leq 7 \\ d \sin^2(x - p/6), & \sqrt[3]{dx} > 7 \end{cases} \\
16 \quad y = \begin{cases} rx^4 - \text{ctg}(x), & rx \leq -2 \\ r \ln^2(|x|), & -2 < rx \leq 2 \\ e - r\sqrt{x}, & rx > 2 \end{cases} \\
17 \quad y = \begin{cases} b|x| - \sqrt[3]{b}, & bx \leq 2 \\ b/(2x), & 2 < bx \leq 6 \\ be^{3x}, & bx > 6 \end{cases} \\
18 \quad y = \begin{cases} \lg^2(|x - a|), & a \ln(|x|) > 7 \\ ax - \text{ctg}(x - p), & 1 < a \ln(|x|) \leq 7 \\ ae^{2x}, & a \ln(|x|) \leq 1 \end{cases} \\
19 \quad y = \begin{cases} q \cos^2(x - p), & qx \leq -1 \\ e^{-qx}, & -1 < qx \leq 2 \\ q \text{ctg}(x + p), & qx > 2 \end{cases} \\
20 \quad y = \begin{cases} \sqrt[3]{b^2} + \cos(x + p), & bx \leq p \\ x/(2|b|), & p < bx \leq 7 \\ e^{-bx}, & bx > 7 \end{cases} \\
21 \quad y = \begin{cases} e^{ax} - \lg(|x|), & a/x \leq 0.5 \\ a/\sin(x - p/6), & 0.5 < a/x \leq 1 \\ \text{ctg}^2(x + p/3), & a/x > 1 \end{cases} \\
22 \quad y = \begin{cases} (e + k)^x, & e^{kx} > 8 \\ p^{2x}, & 1 < e^{kx} \leq 8 \\ \text{ctg}(kx), & e^{kx} \leq 1 \end{cases}
\end{array}$$

$$\begin{array}{l}
23 \quad y = \begin{cases} \sqrt[3]{h} + hx^2, & \ln(|hx|) \leq e \\ h - 3/(5x), & e < \ln(|hx|) \leq 6 \\ xe^{-h}, & \ln(|hx|) > 6 \end{cases} \\
24 \quad y = \begin{cases} ctg(x^2 - p), & a/x \leq 1 \\ a|x| - \lg(x^2), & 1 < a/x \leq 4 \\ e^{ax} - \sqrt{|a|}, & a/x > 4 \end{cases} \\
25 \quad y = \begin{cases} ctg^2(b - p/6), & \sqrt[3]{xb} \leq 1 \\ be^{bx}, & 1 < \sqrt[3]{xb} \leq 4 \\ \sqrt{|bx - \sin(b)/2x|}, & \sqrt[3]{xb} > 4 \end{cases} \\
26 \quad y = \begin{cases} a \lg(|x|), & ax^3 > 5 \\ a^4 - ctg(x - p/6), & 1 < ax^3 \leq 5 \\ \sin(|x^2 - p|), & ax^3 \leq 1 \end{cases} \\
27 \quad y = \begin{cases} \begin{cases} c \cdot arctg^2(x - p/3), & cx > 3 \\ 1/(cx), & 0 < cx \leq 3 \\ \lg(|x^2 - c|), & cx \leq 0 \end{cases} \end{cases} \\
28 \quad y = \begin{cases} \begin{cases} e^{a+x} - tg^2(x - p), & a^2x > 1 \\ |a - (p/2) \cdot x|, & -p \leq a^2x \leq p \\ \cos(x^2 + p/3), & a^2x < -p \end{cases} \end{cases} \\
29 \quad y = \begin{cases} \sqrt[3]{x^2} + ctg(x + p), & px^2 > 8 \\ |x| - e^{px}, & e < px^2 \leq 8 \\ x \cos(x - p/p), & px^2 \leq e \end{cases} \\
30 \quad y = \begin{cases} \lg(|x \cdot a/2|), & x \ln(|a|) > 4 \\ arctg(ax^3 - p), & 1 < x \ln(|a|) \leq 4 \\ e^{-ax} + \sqrt[3]{|x - a|}, & x \ln(|a|) \leq 1 \end{cases}
\end{array}$$

#### Додаток 14. Завдання до лабораторної роботи №4 Частина 1

1. Розрахувати в'язкість аміаку в інтервалі температур від 12 до 23°C

$$N = N_0 \cdot \frac{273,15 + C}{T \cdot C} \cdot \left( \frac{T}{273,15} \right)^n$$

через 1°C за формулою Сатерленда: при умові, що показник  $n$  визначається таким чином:

$$n = A - B \cdot 10^4 \cdot (t - 273,15) + D \cdot 10^{-7} \cdot (t - 273,15)^2,$$

де  $A = 1,06$ ;  $B = 1,04$ ;  $D = 0$ ;  $C = 503$ ;  $N_0 = 91,6 \cdot 10^{-7} \text{ Па} \cdot \text{с}$ .

2. Розрахувати в'язкість аміаку в інтервалі температур від 10 до 20°C

$$N = N_0 \cdot \frac{273,15 + C}{T \cdot C} \cdot \left( \frac{T}{273,15} \right)^n$$

через 2°C за формулою Сатерленда: при умові, що показник  $n$  визначається таким чином:

$$n = A - B \cdot 10^4 \cdot (t - 273,15) + D \cdot 10^{-7} \cdot (t - 273,15)^2,$$

де  $A = 1,06$ ;  $B = 1,04$ ;  $D = 0$ ;  $C = 503$ ;  $N_0 = 91,6 \cdot 10^{-7} \text{ Па} \cdot \text{с}$ .

3. Розрахувати ізобарну теплоємність  $CO_2$  в інтервалі температур від 30 до 100°C через 5°C за формулою:

$$C = E + F \cdot \left( \frac{T}{100} \right) + G \cdot \left( \frac{T}{100} \right)^2 + H^2 \cdot \left( \frac{T}{100} \right)^3 + N^3 \cdot \left( \frac{T}{100} \right)^4,$$

де

$$E = 0,81513; \quad F = 9,8454 \cdot 10^{-2}; \quad G = -9,4747 \cdot 10^{-3}; \quad H = 36,006 \cdot 10^{-5}; \quad N = -0,0567.$$

4. Розрахувати ізобарну теплоємність  $CO_2$  в інтервалі температур від 30 до 150°C через 10°C за формулою:

$$C = E + F \cdot \left( \frac{T}{100} \right) + G \cdot \left( \frac{T}{100} \right)^2 + H^2 \cdot \left( \frac{T}{100} \right)^3 + N^3 \cdot \left( \frac{T}{100} \right)^4,$$

де

$$E = 0,81513; \quad F = 9,8454 \cdot 10^{-2}; \quad G = -9,4747 \cdot 10^{-3}; \quad H = 36,006 \cdot 10^{-5}; \quad N = -0,0567.$$

5. Розрахувати ізобарну теплоємність аміаку в інтервалі температур від 80 до 180°C через 10°C за формулою:

$$C = E + F \cdot \left( \frac{T}{100} \right) + G \cdot \left( \frac{T}{100} \right)^2 + H^2 \cdot \left( \frac{T}{100} \right)^3 + N^3 \cdot \left( \frac{T}{100} \right)^4,$$

де

$$E = 2,0183; \quad F = 158,072 \cdot 10^{-2}; \quad G = 273,61 \cdot 10^{-3}; \quad H = -9380,9 \cdot 10^{-5}; \quad N = -1,9986.$$

6. Розрахувати ізобарну теплоємність аміаку в інтервалі температур від 100 до 160°C через 5°C за формулою:

$$C = E + F \cdot \left( \frac{T}{100} \right) + G \cdot \left( \frac{T}{100} \right)^2 + H^2 \cdot \left( \frac{T}{100} \right)^3 + N^3 \cdot \left( \frac{T}{100} \right)^4,$$



де  
 $E = 2,0183; F = 158,072 \cdot 10^{-2}; G = 273,61 \cdot 10^{-3}; H = -9380,9 \cdot 10^{-5}; N = -1,9986$ .

7. Розрахувати ізохорну теплоємність  $CO_2 - C_V$  в інтервалі температур від 30 до 100°C через 5°C. Відомо, що ізобарна теплоємність  $CO_2$  розраховується за формулою:

$$C = E + F \cdot \left(\frac{T}{100}\right) + G \cdot \left(\frac{T}{100}\right)^2 + H^2 \cdot \left(\frac{T}{100}\right)^3 + N^3 \cdot \left(\frac{T}{100}\right)^4,$$

де  
 $E = 0,81513; F = 9,8454 \cdot 10^{-2}; G = -9,4747 \cdot 10^{-3}; H = 36,006 \cdot 10^{-5}; N = -0,0567$ .

А формула зв'язку між теплоємністями має вигляд:  $C_V = C - R/M$  (M – молекулярна маса газу; R – універсальна газова стала).

8. Розрахувати ізохорну теплоємність  $CO_2 - C_V$  в інтервалі температур від 30 до 150°C через 10°C. Відомо, що ізобарна теплоємність  $CO_2$  розраховується за формулою:

$$C = E + F \cdot \left(\frac{T}{100}\right) + G \cdot \left(\frac{T}{100}\right)^2 + H^2 \cdot \left(\frac{T}{100}\right)^3 + N^3 \cdot \left(\frac{T}{100}\right)^4,$$

де  
 $E = 0,81513; F = 9,8454 \cdot 10^{-2}; G = -9,4747 \cdot 10^{-3}; H = 36,006 \cdot 10^{-5}; N = -0,0567$ .

А формула зв'язку між теплоємністями має вигляд:  $C_V = C - R/M$  (M – молекулярна маса газу; R – універсальна газова стала).

9. Розрахувати ізохорну теплоємність аміаку –  $C_V$  в інтервалі температур від 80 до 180°C через 10°C. Відомо, що ізобарна теплоємність аміаку розраховується за формулою:

$$C = E + F \cdot \left(\frac{T}{100}\right) + G \cdot \left(\frac{T}{100}\right)^2 + H^2 \cdot \left(\frac{T}{100}\right)^3 + N^3 \cdot \left(\frac{T}{100}\right)^4,$$

де  
 $E = 2,0183; F = 158,072 \cdot 10^{-2}; G = 273,61 \cdot 10^{-3}; H = -9380,9 \cdot 10^{-5}; N = -1,9986$ .

А формула зв'язку між теплоємністями має вигляд:  $C_V = C - R/M$  (M – молекулярна маса газу; R – універсальна газова стала).

10. Розрахувати ізохорну теплоємність аміаку –  $C_V$  в інтервалі температур від 100 до 100°C через 5°C. Відомо, що ізобарна теплоємність аміаку розраховується за формулою:

$$C = E + F \cdot \left(\frac{T}{100}\right) + G \cdot \left(\frac{T}{100}\right)^2 + H^2 \cdot \left(\frac{T}{100}\right)^3 + N^3 \cdot \left(\frac{T}{100}\right)^4,$$

де  
 $E = 2,0183; F = 158,072 \cdot 10^{-2}; G = 273,61 \cdot 10^{-3}; H = -9380,9 \cdot 10^{-5}; N = -1,9986$ .

А формула зв'язку між теплоємністями має вигляд:  $C_V = C - R/M$  ( $M$  – молекулярна маса газу;  $R$  – універсальна газова стала).

11. Розрахувати в'язкість води в інтервалі температур від 16 до 27°C через 1°C за формулою:  $N_0 = A(B-t)^n$ , де  $A = 0,59849$ ;  $B = 43,252$ ;  $n = -1,5423$ .

12. Розрахувати в'язкість води в інтервалі температур від 12 до 32°C через 2°C за формулою:  $N_0 = A(B-t)^n$ , де  $A = 0,59849$ ;  $B = 43,252$ ;  $n = -1,5423$ .

13. Розрахувати теплопровідність аміаку в інтервалі температур від 30 до 50°C через 2°C за формулою:  $L = A \frac{(T_K + C)}{(T + C)} \cdot \frac{T^{3/2}}{T_K^{3/2} \cdot M^{5/6}}$ , де  $A = 4,3543 \cdot 10^{-3}$ ;  $C = 503$ ;  $T_K = 239,73$ ;  $M$  – молекулярна маса газу.

14. Розрахувати теплопровідність аміаку в інтервалі температур від 10 до 70°C через 5°C за формулою:  $L = A \frac{(T_K + C)}{(T + C)} \cdot \frac{T^{3/2}}{T_K^{3/2} \cdot M^{5/6}}$ , де  $A = 4,3543 \cdot 10^{-3}$ ;  $C = 503$ ;  $T_K = 239,73$ ;  $M$  – молекулярна маса газу.

15. Розрахувати густину повітря в інтервалі температур від 45 до 54°C через 1°C за формулою:  $R = \frac{1,293 \cdot P}{(1 + 0,00367 \cdot t) \cdot 760} \frac{\text{кг}}{\text{м}^3}$ , де  $P$  – тиск, який дорівнює 105,2 кПа.

16. Розрахувати густину повітря в інтервалі температур від 14 до 44°C через 2°C за формулою:  $R = \frac{1,293 \cdot P}{(1 + 0,00367 \cdot t) \cdot 760} \frac{\text{кг}}{\text{м}^3}$ , де  $P$  – тиск, який дорівнює 105,2 кПа.

17. Розрахувати швидкість осадження кулеподібної частинки радіуса  $r$  ( $r = 0,2 \cdot 10^{-3} \text{ м}$ ;  $0,3 \cdot 10^{-3} \text{ м}$ ;  $0,4 \cdot 10^{-3} \text{ м}$ ; ... ;  $0,9 \cdot 10^{-3} \text{ м}$ ) у газі. Використати формули:  $d = 2 \cdot r$  і

$$W_{oc} = \frac{d^2 \cdot R \cdot g}{18 \cdot M_c}, \text{ де } g = 9,81 \text{ м}^2/\text{с}; R = 1684 \text{ кг}/\text{м}^3; M = 5,2 \cdot 10^{-4} \text{ Н} \cdot \text{с}/\text{м}^2.$$

18. Розрахувати швидкість осадження кулеподібної частинки радіуса  $r$  ( $r = 0,25 \cdot 10^{-3} \text{ м}$ ;  $0,35 \cdot 10^{-3} \text{ м}$ ;  $0,45 \cdot 10^{-3} \text{ м}$ ; ... ;  $0,95 \cdot 10^{-3} \text{ м}$ ) у газі. Використати формули:  $d = 2 \cdot r$  і

$$W_{oc} = \frac{d^2 \cdot R \cdot g}{18 \cdot M_c}, \text{ де } g = 9,81 \text{ м}^2/\text{с}; R = 1684 \text{ кг}/\text{м}^3; M = 5,2 \cdot 10^{-4} \text{ Н} \cdot \text{с}/\text{м}^2.$$

19. Розрахувати час процесу фільтрування суспензії об'ємом  $V$  ( $V = 50 \cdot 10^{-6}$ ;  $100 \cdot 10^{-6}$ ;  $150 \cdot 10^{-6}$ ;  $200 \cdot 10^{-6}$ ;  $250 \cdot 10^{-6}$ ;  $\text{м}^3$ ) за формулою:

$$t = \frac{M \cdot r_0 \cdot x_0}{2P} \cdot \frac{V^2}{S^2} + \frac{M \cdot R}{P} \cdot \frac{V}{S}$$

де  $M = 5,17 \cdot 10^{-3} \text{ Н} \cdot \text{с}/\text{м}^3$ ;  $r_0 = 4,15 \cdot 10^{12} \text{ м}^{-2}$ ;  $R = 2,3 \cdot 10^{10} \text{ м}^{-1}$ ;  
 $x_0 = 0,178 \text{ м}^3/\text{м}^3$ ;  $P = 0,6 \cdot 10^5 \text{ Н}/\text{м}^2$ ;  $S = 1 \text{ м}$ .

20. Розрахувати час процесу фільтрування суспензії об'ємом  $V$  ( $V = 0,2 \text{ м}^3$ ;  $0,3 \text{ м}^3$ ;  $0,4 \text{ м}^3$ ; ...  $0,9 \text{ м}^3$ ) за формулою:

$$t = \frac{M \cdot r_0 \cdot x_0}{2P} \cdot \frac{V^2}{S^2} + \frac{M \cdot R}{P} \cdot \frac{V}{S}$$

де  $M = 5,17 \cdot 10^{-3} \text{ Н} \cdot \text{с}/\text{м}^3$ ;  $r_0 = 4,15 \cdot 10^{12} \text{ м}^{-2}$ ;  $R = 2,3 \cdot 10^{10} \text{ м}^{-1}$ ;  
 $x_0 = 0,178 \text{ м}^3/\text{м}^3$ ;  $P = 0,6 \cdot 10^5 \text{ Н}/\text{м}^2$ ;  $S = 1 \text{ м}$ .

21. Розрахувати витрату потужності на перемішування для пропелерної мішалки діаметру  $d = 2,0$ ;  $2,2$ ;  $2,4$ ; ...  $3,2 \text{ м}$ .

Використати формулу:  $N = K_N \cdot n^3 \cdot R_c \cdot d_m^5$ , де  
 $K_N = 0,32$ ;  $n = 3,96 \text{ об}/\text{с}$ ;  $R_c = 1200 \text{ кг}/\text{м}^3$ .

22. Розрахувати витрату потужності на перемішування для пропелерної мішалки діаметру  $d = 2,0$ ;  $2,1$ ;  $2,3$ ; ...  $3,1 \text{ м}$ .

Використати формулу:  $N = K_N \cdot n^3 \cdot R_c \cdot d_m^5$ , де  
 $K_N = 0,32$ ;  $n = 3,96 \text{ об}/\text{с}$ ;  $R_c = 1200 \text{ кг}/\text{м}^3$ .

23. Розрахувати коефіцієнт теплопровідності  $L_t$  нітробензолу в інтервалі температур від  $40$  до  $110^\circ\text{C}$  через  $10^\circ\text{C}$ . Використати формулу:

$$L_t = L_{30} \cdot [1 - E \cdot (t - 30)], \text{ де } L_{30} = A_1 \cdot c \cdot R \cdot \sqrt{\frac{R}{M}},$$

$A_1 = 4,22 \cdot 10^{-2}$ ;  $c = 1,38 \cdot 10^3 \text{ Дж}/(\text{кг} \cdot \text{град})$ ;  $R = 1200 \text{ кг}/\text{м}^3$ ;  
 $M = 123$ ;  $E = 1,0 \cdot 10^{-3} \text{ град}^{-1}$ .

24. Розрахувати коефіцієнт теплопровідності  $L_t$  нітробензолу в інтервалі температур від  $20$  до  $00^\circ\text{C}$  через  $5^\circ\text{C}$ . Використати формулу:

$$L_t = L_{30} \cdot [1 - E \cdot (t - 30)], \text{ де } L_{30} = A_1 \cdot c \cdot R \cdot \sqrt{\frac{R}{M}},$$

$$A_1 = 4,22 \cdot 10^{-2}; \quad c = 1,38 \cdot 10^3 \text{ Дж}/(\text{кг} \cdot \text{град}); \quad R = 1200 \text{ кг}/\text{м}^3;$$

$$M = 123; \quad E = 1,0 \cdot 10^{-3} \text{ град}^{-1}.$$

25. Розрахувати значення густини  $\rho$  (R) та в'язкості  $\mu$  (M) сірчаної кислоти в інтервалі температур від 10 до 80°C через 10°C. Використати формули:

$$R = 1894,8 - 0,909 \cdot t;$$

$$M = 1,406 \cdot 10^{-3} + 5,0087 \cdot 10^{-1} / t.$$

26. Розрахувати значення густини  $\rho$  (R) та в'язкості  $\mu$  (M) сірчаної кислоти в інтервалі температур від 20 до 70°C через 5°C. Використати формули:

$$R = 1894,8 - 0,909 \cdot t;$$

$$M = 1,406 \cdot 10^{-3} + 5,0087 \cdot 10^{-1} / t.$$

27. Розрахувати опір шару осадка при фільтруванні суспензії з діаметром частинок  $d_{cp} = (0,1; 0,12; 0,14; \dots; 0,28) \cdot 10^{-3} \text{ м}$ , за формулою:  $r_0 = A \cdot d_{cp}^B$ , де  $A = 9,8373 \cdot 10^{10}$ ;  $B = -0,3224$ .

28. Розрахувати опір шару осадка при фільтруванні суспензії з діаметром частинок  $d_{cp} = (0,1; 0,11; 0,12; \dots; 0,22) \cdot 10^{-3} \text{ м}$ , за формулою:  $r_0 = A \cdot d_{cp}^B$ , де  $A = 9,8373 \cdot 10^{10}$ ;  $B = -0,3224$ .

29. Розрахувати залежність продуктивності вакуум-фільтру  $W$  ( $\text{кг}/(\text{час} \cdot \text{м}^3)$ ) від величини вакууму  $P$  ( $\text{Н}/\text{м}^2$ ), який змінюється від 25 до 125 мм.рт.ст з кроком  $P=10$  мм.рт.ст. для розміру часток  $d=0,25$  мм, якщо:  $W = 7056 \cdot P^{0,5} \cdot d^2$ .

30. Розрахувати залежність продуктивності вакуум-фільтру  $W$  ( $\text{кг}/(\text{час} \cdot \text{м}^3)$ ) від величини вакууму  $P$  ( $\text{Н}/\text{м}^2$ ), який змінюється від 15 до 80 мм.рт.ст з кроком  $P=5$  мм.рт.ст. для розміру часток  $d=0,30$  мм, якщо:  $W = 7056 \cdot P^{0,5} \cdot d^2$ .

31. Розрахувати залежність продуктивності вакуум-фільтру  $W$  ( $\text{кг}/(\text{час} \cdot \text{м}^3)$ ) від величини вакууму  $P$  ( $\text{Н}/\text{м}^2$ ), який змінюється від 20 до 100 мм.рт.ст з кроком  $P=10$  мм.рт.ст. для розміру часток  $d=0,35$  мм, якщо:  $W = 7056 \cdot P^{0,5} \cdot d^2$ .

32. Розрахувати тиск пару хлористого метилена  $P$  (та  $\lg P$ ) в інтервалі температур 20...40°C через 5°C, якщо:

$$\lg P = -A/T + B \cdot \lg T + C,$$

де  $A = 1995,6$ ;  $B = -3,4426$ ;  $T = t + 273$ ;  $C = 8,321$ .

33. Розрахувати тиск пару хлористого метилену  $P$  (та  $\lg P$ ) в інтервалі температур  $10 \dots 90^\circ\text{C}$  через  $10^\circ\text{C}$ , якщо:

$$\lg P = -A/T + B \cdot \lg T + C,$$

де  $A = 1995,6$ ;  $B = -3,4426$ ;  $T = t + 273$ ;  $C = 8,621$ .

34. Розрахувати тиск пару хлористого метилену  $P$  (та  $\lg P$ ) в інтервалі температур  $15 \dots 55^\circ\text{C}$  через  $5^\circ\text{C}$ , якщо:

$$\lg P = -A/T + B \cdot \lg T + C,$$

де  $A = 1995,6$ ;  $B = -3,4426$ ;  $T = t + 273$ ;  $C = 8,921$ .

35. Розрахувати швидкість  $R$  хімічної реакції  $A \rightarrow B$  в діапазоні температур  $t = 200 \dots 400^\circ\text{K}$  кроком  $20^\circ\text{K}$ , якщо:

$$R_1 = K_0 \cdot \exp(-E/R \cdot T) \cdot C_0,$$

де  $K_0 = 2,05 \text{ c}^{-1}$ ;  $R = 1,98725 \text{ кал/моль} \cdot \text{град}$ ;  $E = 20100 \text{ кал/моль}$ ;  $C_0 = 2 \text{ моль/м}^3$ .

36. Розрахувати швидкість  $R$  хімічної реакції  $A \rightarrow B$  в діапазоні температур  $t = 340 \dots 400^\circ\text{K}$  кроком  $10^\circ\text{K}$ , якщо:

$$R_1 = K_0 \cdot \exp(-E/R \cdot T) \cdot C_0,$$

де  $K_0 = 2,05 \text{ c}^{-1}$ ;  $R = 1,98725 \text{ кал/моль} \cdot \text{град}$ ;  $E = 20100 \text{ кал/моль}$ ;  $C_0 = 4 \text{ моль/м}^3$ .

37. Розрахувати швидкість  $R$  хімічної реакції  $A \rightarrow B$  в діапазоні температур  $t = 315 \dots 490^\circ\text{K}$  кроком  $15^\circ\text{K}$ , якщо:

$$R_1 = K_0 \cdot \exp(-E/R \cdot T) \cdot C_0,$$

де  $K_0 = 2,05 \text{ c}^{-1}$ ;  $R = 1,98725 \text{ кал/моль} \cdot \text{град}$ ;  $E = 20100 \text{ кал/моль}$ ;  $C_0 = 6 \text{ моль/м}^3$ .

38. Розрахувати, який об'єм займе  $10 \text{ г}$  азоту при тиску  $P = 1,5 \text{ атм}$  і зміні температури від  $200$  до  $320^\circ\text{K}$  з кроком  $10^\circ\text{K}$ .

( Використати формулу :  $P \cdot V = \frac{m}{M} \cdot R \cdot T$  ; де  $P$  – тиск;  $V$  – об'єм;  $m$  – маса газу;  $M$  – молекулярна маса;  $T$  – абсолютна температура;  $R$  – універсальна газова стала ( $R = 0,082057 \text{ л} \cdot \text{атм/моль} \cdot \text{град}$ ).

39. Розрахувати, який об'єм займе  $20 \text{ г}$  азоту при тиску  $P = 3,0 \text{ атм}$  і зміні температури від  $150$  до

280 °K з кроком 10 °K.

( Використати формулу :  $P \cdot V = \frac{m}{M} \cdot R \cdot T$  ; де P – тиск; V – об'єм; m – маса газу; M – молекулярна маса; T – абсолютна температура; R – універсальна газова стала (R=0,082057 л\*атм/моль\*град).

40.Розрахувати, який об'єм займе 15 г азоту при тиску P=4,0 атм і зміні температури від 140 до 280 °K з кроком 10 °K.

( Використати формулу :  $P \cdot V = \frac{m}{M} \cdot R \cdot T$  ; де P – тиск; V – об'єм; m – маса газу; M – молекулярна маса; T – абсолютна температура; R – універсальна газова стала (R=0,082057 л\*атм/моль\*град).

41.Розрахувати значення константи швидкості хімічної реакції

$$K = A \cdot \exp\left(\frac{-5,29 \cdot 10^4}{R \cdot T}\right)$$

в інтервалі температур 550...850° через 25°С, якщо  $A = 7,5 \cdot 10^{-5}$ .

42.Розрахувати значення константи швидкості хімічної реакції

$$K = A \cdot \exp\left(\frac{-5,29 \cdot 10^4}{R \cdot T}\right)$$

в інтервалі температур 600...880° через 20°С, якщо  $A = 8,5 \cdot 10^{-5}$ .

43.Розрахувати значення константи швидкості хімічної реакції

$$K = A \cdot \exp\left(\frac{-5,29 \cdot 10^4}{R \cdot T}\right)$$

в інтервалі температур 520...900° через 40°С, якщо  $A = 8,0 \cdot 10^{-5}$ .

44.Розрахувати значення енергії Гіббса

$$G = -R \cdot T \cdot \ln(K)$$

хімічної реакції в інтервалі температур від 400 до 680 К з кроком 20°, якщо R=1,3143 Дж/моль\*К, K=9,137.

45.Розрахувати значення енергії Гіббса

$$G = -R \cdot T \cdot \ln(K)$$

хімічної реакції в інтервалі температур від 460 до 600 К з кроком 10°, якщо R=1,3143 Дж/моль\*К, K=9,237.

46.Розрахувати значення енергії Гіббса

$$G = -R \cdot T \cdot \ln(K)$$

хімічної реакції в інтервалі температур від 450 до 750 К з кроком 25°, якщо R=1,3143 Дж/моль\*К, K= 9,337.

47. Розрахувати об'єм кульового сегменту

$$V = \frac{\pi \cdot h}{6} \cdot (h^2 + 3 \cdot r^2)$$

якщо  $h$  змінюється від 0,1 до 0,7 м, з кроком 0,05 м, для  $r=0,5$  м.

48. Розрахувати об'єм кульового сегменту

$$V = \frac{\pi \cdot h}{6} \cdot (h^2 + 3 \cdot r^2)$$

якщо  $h$  змінюється від 0,1 до 1,9 м, з кроком 0,1 м, для  $r=1,2$  м.

49. Розрахувати об'єм кульового сегменту

$$V = \frac{\pi \cdot h}{6} \cdot (h^2 + 3 \cdot r^2)$$

якщо  $h$  змінюється від 0,15 до 0,85 м, з кроком 0,05 м, для  $r=1,5$  м.

50. Розрахувати значення тригонометричних функцій :

$$\sin(A), \cos(A) \text{ і } \operatorname{ctg}(A)$$

при зміні  $A$  від  $\pi/24$  до  $11 \cdot \pi/24$  з кроком  $\pi/24$ .

51. Розрахувати значення тригонометричних функцій :

$$\sin(A), \cos(A) \text{ і } \operatorname{ctg}(A)$$

при зміні  $A$  від  $30$  до  $150^\circ$  з кроком  $15^\circ$ .

52. Для десяти значень  $X$  перевірити, чи є рівняння

$$\sin^6 X + \cos^6 X = 1 - 3/4 \cdot \sin^2 2X$$

тотожністю. Рівняння вважати тотожністю, якщо

$$|(\text{значення лівої частини}) - (\text{значення правої частини})| \leq 0,001.$$

53. Для десяти значень  $X$  перевірити, чи є рівняння

$$\operatorname{tg} X + \operatorname{tg} 2X - \operatorname{tg} 3X = -\operatorname{tg} X \cdot \operatorname{tg} 2X \cdot \operatorname{tg} 3X.$$

тотожністю.

Рівняння вважати тотожністю, якщо

$$|(\text{значення лівої частини}) - (\text{значення правої частини})| \leq 0,001.$$

54. Для десяти значень  $X$  перевірити, чи є рівняння

$$\sin 2nA + \sin 2nB + \sin 2nG = (-1)^{n+1} \cdot 4 \cdot \sin nA \cdot \sin nB \cdot \sin nG, \text{ де } n = 1, 2, 3 \dots$$

тотожністю. Рівняння вважати тотожністю, якщо

$$|(\text{значення лівої частини}) - (\text{значення правої частини})| \leq 0,001.$$

55. Виконати 5-7 перевірок рівняння:

$$\sin A + \sin B + \sin G = 4 \cdot \sin \frac{A+B}{2} \cdot \sin \frac{B+G}{2} \cdot \sin \frac{G+A}{2} \quad \text{на тотожність,}$$

якщо  $A$ ,  $B$  та  $G$  – кути трикутника. Рівняння вважати тотожністю, якщо  
| (значення лівої частини) - (значення правої частини) |  $\leq 0,001$ .



### Додаток 15. Завдання до лабораторної роботи №4 Частина 2

№	Функція	Дані для розрахунків
1	$t = \frac{M \cdot r_0 \cdot x_0}{2P} \cdot \frac{V^2}{S^2} + \frac{M \cdot R}{P} \cdot \frac{V}{S}$	$M = 5,17 \cdot 10^{-3}$ ; $r_0 = 4,15 \cdot 10^{12}$ ; $R = 2,3 \cdot 10^{10}$ ; $x_0 = 0,178$ ; $S = 1$ ; $P = 0,3 \cdot 10^5, 0,4 \cdot 10^5, 0,5 \cdot 10^5, 0,6 \cdot 10^5$ ; $V = 0,2$ ; $0,3$ ; $0,4$ .
2	$R_1 = K_0 \cdot \exp(-E/(R \cdot T)) \cdot C_0$	$K_0 = 2,05$ ; $R = 1,98725$ ; $E = 20100$ ; $C_0 = 4, 5, 6$ ; $t = 340 \dots 400$ , крок 20.
3	$W_{oc} = \frac{d^2 \cdot R \cdot g}{18 \cdot M_c}$	$g = 9,81$ ; $R = 1684$ ; $M = 5,0 \cdot 10^{-4}; 5,2 \cdot 10^{-4}; 5,4 \cdot 10^{-4}$ ; $r = 0,2 \cdot 10^{-3}; 0,3 \cdot 10^{-3}; 0,4 \cdot 10^{-3}; 0,5 \cdot 10^{-3} \cdot (d=2r)$
4	$R = \frac{k_1 \cdot P}{760 \cdot (1 + k_2 \cdot t)}$	$k_1 = 1,293$ ; $k_2 = 0,00367$ ; $P = 105,2; 125,2; 145,2$ ; $t = 14 \dots 44$ , крок 10.

## Додаток 16. Приклади зчитування інформації з листів Excel в VBA і запису інформації з програми VBA в лист Excel

1. Використання одного з варіантів синтаксису властивості Cells:  
об'єкт. Cells ( номер\_рядка, номер\_стовпчика) ;

Приклад 1:

Масив з 10 елементів записано у 1 рядку. Ввести масив в програму, а потім без змін записати в 3 рядок.

```
Option Base 1
Public Sub prim1()
Dim A(10) As Integer, i As Integer
' зчитування цілочисельного масива з листа Excel
For i = 1 To 10
    A(i) = Cells(1, i)
Next i
' запис цілочисельного масива в лист Excel
For i = 1 To 10
    Cells(3, i) = A(i)
Next i
End Sub
```

Результат роботи програми:

	A	B	C	D	E	F	G	H	I	J
1	3	4	3	9	7	11	4	8	1	10
2										
3	3	4	3	9	7	11	4	8	1	10
4										

Приклад 2:

Масив з 5 елементів записано у 2 стовпчику. Ввести масив в програму, а потім без змін записати в 4 стовпчик.

```
Option Base 1
```

```
Public Sub prim1()
```

```
Dim A(5) As Integer, i As Integer
```

```
' зчитування цілочисельного масива з листа Excel
```

```
For i = 1 To 5
```

```
    A(i) = Cells(i, 2)
```

```
Next i
```

```
' запис цілочисельного масива в лист Excel
```

```
For i = 1 To 5
```

```
    Cells(i,4) = A(i)
```

```
Next i
```

```
End Sub
```

Результат роботи програми:

	A	B	C	D
1		2		2
2		8		8
3		11		11
4		75		75
5		9		9

### Приклад 3:

Розрахувати значення функції  $y = a \cdot x^2 + b \cdot k$ , при наступних значеннях змінних:  $a = 5.8$ ;  $b = 12.37$ ;  $x$  змінюється від 2 до 6 з кроком 2;  $k$  змінюється від 0.5 до 0.7 з кроком 0.1. Для програмування задачі використати конструкцію вкладених циклів. Ввести дані в програму з листа Excel, а потім в лист Excel вивести і результати роботи програми.

Option Base 1

Public Sub prim1()

' Табуляція функції

Dim a As Single, b As Single

Dim k As Single, y As Single

Dim x As Integer, m As Integer

a = Cells(2,1)

b = Cells(2,2)

Cells(3,1) = "Результати розрахунків"

m = 4

For x = 2 To 6 Step 2

Cells(m,1) = "При x=" : Cells(m,2) = x

For k = 0.5 To 0.8 Step 0.1

m = m+1

y = a \* x ^ 2 + b \* k

Cells(m,1) = "k=" : Cells(m,2) = k

Cells(m,3) = "y=" : Cells(m,4) = y

Next k

m = m + 1

Next x

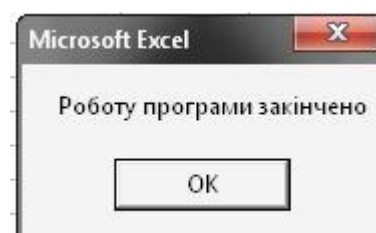
' Повідомлення про закінчення роботи

MsgBox ("Роботу програми закінчено")

End Sub

Результат роботи програми:

	A	B	C	D
1	a	b		
2	5,8	12,37		
3	Результати розрахунків			
4	При x=	2		
5	k=	0,5	y=	29,385
6	k=	0,6	y=	30,622
7	k=	0,7	y=	31,859
8	При x=	4		
9	k=	0,5	y=	98,985
10	k=	0,6	y=	100,222
11	k=	0,7	y=	101,459
12	При x=	6		
13	k=	0,5	y=	214,985
14	k=	0,6	y=	216,222
15	k=	0,7	y=	217,459



Приклад 4:

Двовимірний масив А розміром 4 на 4 записано у листі Excel. Ввести масив в програму, а потім без змін записати в той же лист Excel.

Option Base 1

Public Sub prim1()

Dim A(4,4) As Integer, i As Integer

Dim j As Integer, m As Integer

' зчитування цілочисельного масива з листа Excel

' перший елемент масива у 1 рядку і 1 стовпчику

For i = 1 To 4

For j = 1 To 4

A(i, j) = Cells(i, j)

Next j

Next i

' запис цілочисельного масива в лист Excel

' перший елемент виводимо у 6 рядок і 1 стовпчик

m=6

For i = 1 To 4

For j = 1 To 4

Cells(m, j) = A(i, j)

Next j

m= m+1

Next i

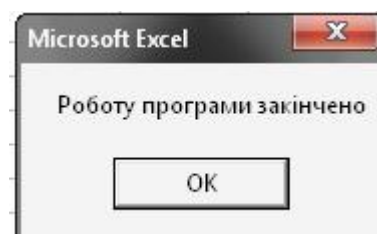
' Повідомлення про закінчення роботи

MsgBox ("Роботу програми закінчено")

End Sub

Результат роботи програми:

	A	B	C	D
1	3	7	11	6
2	1	5	3	9
3	6	23	8	2
4	7	1	4	5
5				
6	3	7	11	6
7	1	5	3	9
8	6	23	8	2
9	7	1	4	5



## 2. Використання одного з варіантів синтаксису об'єкта Range і властивості Cells:

Приклад 5:

Двовимірний масив А розміром 4 на 4 записано у листі Excel. Ввести масив в програму, а потім без змін записати в той же лист Excel.

Option Base 1

```
Public Sub prim1()
```

```
Dim A(8) As Integer, i As Integer
```

```
'Виділення діапазона комірок
```

```
Range("a1:a8").Select
```

```
' зчитування цілочисельного масива з листа Excel
```

```
For i = 1 To 8
```

```
    A(i) = Selection.Cells.Item(i)
```

```
Next i
```

```
'Виділення діапазона комірок
```

```
Range("b1:b8").Select
```

```
For i = 1 To 8
```

```
    Selection.Cells.Item(i) = A(i)
```

```
Next i
```

```
' Повідомлення про закінчення роботи
```

```
MsgBox ("Роботу програми закінчено")
```

```
End Sub
```

Результат роботи програми:

	А	В
1	3	3
2	1	1
3	6	6
4	7	7
5	6	6
6	9	9
7	2	2
8	5	5

