

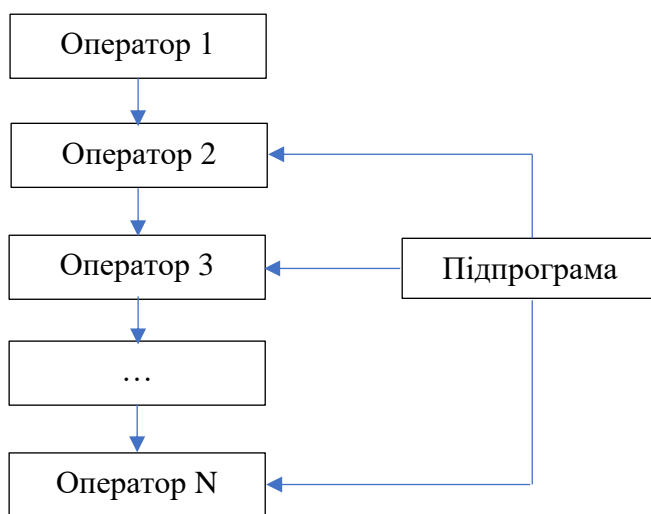
Тема 5. Лінійні, розгалужені та циклічні конструкції VBA

План

1. Лінійні конструкції VBA
2. Розгалужені конструкції VBA
3. Циклічні конструкції VBA

5.1. Лінійні мовні конструкції

Якщо певні дії повинні багаторазово виконуватись в різних фрагментах програми, то їх можна оформити у вигляді окремих підпрограм, які викликатимуться на виконання у разі потреби.



Переваги підпрограм:

- зменшення довжини програмного коду (набір інструкцій, що повторюються, потрібно ввести тільки 1 раз);
- прискорення процесу редагування (при необхідності внесення правок, текст коду потрібно змінити тільки в одному місці);
- наочність й легкість читання програмного коду (програмний код легше читати, оскільки в ньому вказується лише інструкція виклику підпрограми);
- прискорення процесу розробки програм (часто готові підпрограми можна копіювати в інші проекти, що прискорює процес розробки додатків).

У VBA існує два типи підпрограм:

1. Процедури – виконують набір дій, але не повертають ніякого результату;
2. Функції – виконують набір дій, але при цьому повертають результат.

Якщо, наприклад, потрібно обчислити відстань між двома географічними точками, то зручніше працювати з функцією, оскільки ця підпрограма повинна повертати результат.

Підпрограми можуть мати аргументи (параметри), які несуть вхідну інформацію.

Процедури

Процедура оголошується за допомогою наступного набору операторів:

```
Sub НазваПроцедури (Параметри)
    Оператор (и)
End Sub
```

Назва процедури підпорядковується тим самим правилам, що й назви змінних.

Приклад. Виведення на екран повідомлення, процедура ShowMessage:

```
Sub ShowMessage
    MsgBox("Hello World!")
End Sub
```

Якщо процедура не потребує вхідних даних, то вказувати параметри в її заголовку не обов'язково. Параметри, що передаються у процедуру ззовні, заключають в дужки після назви процедури.

Виклик процедури може здійснюватись різними способами:

```
ShowMessage          ' без аргументів
ShowMessage ()       ' порожній список аргументів
ShowMessage (Параметри) ' список аргументів
Call ShowMessage     ' оператор Call та відсутність
аргументів
Call ShowMessage ()   ' оператор Call та порожній список
аргументів
Call ShowMessage (Параметри) ' оператор Call та список
аргументів
```

Оператор Call використовується за бажанням.

Передача аргументів

Аргументи (параметри), що передаються в процедуру – це дані, які використовуються в обчисленнях на рівні процедури й повинні бути одержані зовні.

Приклад. Процедура з одним параметром:

```
Sub ShowMessage (strMessage)
    MsgBox (strMessage)
End Sub
```

Аргумент strMessage є змінною, яка містить значення, що передається процедурі. Ця змінна має локальну область видимості (тобто, значення цієї змінної можна побачити тільки в межах даної процедури).

Тоді, виклик процедури ShowMessage буде здійснюватись одним з наступних способів:

```
ShowMessage "Hello"
```

```
ShowMessage ("Hello")
Call ShowMessage ("Hello")
```

Є можливість визначити процедуру, якій передаються декілька аргументів (перераховуються через кому).

Функції

Функції багато в чому схожі на процедури, але на відміну від них можуть повертати значення.

Синтаксис визначення функції:

```
Function НазваФункції (Параметри)
    оператор (и)
    НазваФункції = Значення
End Function
```

Функції повертають результат за рахунок того, що її назві привласнюється деяке значення всередині самої функції.

Приклад. Функція розрахунку квадрату числа:

```
Number = 5

Function Sq(x)
    Sq = x * x
End Function

Result = Sq(Number)
```

Управління ходом виконання програм здійснюється за допомогою розгалужених та циклічних мовних конструкцій (умовних операторів та операторів циклу). В наступних питаннях розглянемо їх більш детально.

5.2. Розгалужені мовні конструкції

В ході свого виконання, програми можуть ухвалювати рішення, за яким шляхом йти далі. Ці рішення залежать від виконання певних умов: значення тієї чи іншої змінної, положення перемикачів на формі й інших чинників. Мова програмування VBA має повний набір умовних операторів, які дозволяють «навчити» програму ухвалювати рішення. Умовні оператори оцінюють деяку умову й вирішують, який блок команд слід виконувати в тому випадку, якщо ця умова виконується, або ні.

VBA підтримує наступні умовні оператори:

- If...Then
- If...Then...Else
- If...Then...Elseif
- Select Case

If...Then

Оператор If...Then, найпоширеніший умовний оператор, виконує один або групу операторів, якщо умова виконується:

```
If умова Then
    оператор (и)
End If
```

Якщо умова виконується, то виконується й група операторів, що розташована між Then та End If. Якщо ж умова не виконується, то програма переходить до виконання наступного за інструкцією End If рядка коду.

Якщо необхідно виконати тільки один оператор, то можна скористатись однорядковим синтаксисом:

```
If умова Then оператор
```

Приклад. Оператор If...Then:

```
Dim Sales, Commission
Commission = 0.05
Sales = 12500.00
If Sales >= 10000.00 Then
    Commission = 0.10
End If
```

If...Then...Else

Оператор If...Then...Else дозволяє включити альтернативну групу операторів. Якщо умова виконується, то виконується й перша група операторів. Якщо ж умова не виконується, то виконується друга група операторів:

```
If умова Then
    оператор (и) для випадку "умова виконується"
Else
    оператор (и) для випадку "умова не виконується"
End If
```

Приклад. Оператор If...Then...Else:

```
Sales = 12500.00
If Sales <= 10000.00 Then
    Commission = 0.10
Else
    Commission = 0.15
End If
```

If...Then...ElseIf

За допомогою оператора If...Then...ElseIf можна перевіряти виконання декількох умов одночасно:

```

If умова1 Then
    оператор(и) для випадку "умова1 виконується"
Elseif умова2 Then
    оператор(и) для випадку "умова2 виконується "
Elseif умова3 Then
    оператор(и) для випадку "умова3 виконується "
Else
    оператор(и) для випадку "жодна умова не виконується"
End If

```

У складі умовних операторів можуть використовуватися логічні операції.

Приклад. Оператор If...Then...ElseIf:

```

Sales = 3250.00
If Sales < 1000 Then
    Commission = 0.05
Elseif Sales >= 1000 And Sales < 5000 Then
    Commission = 0.10
Elseif Sales >= 5000 And Sales < 10000 Then
    Commission = 0.15
Else
    Commission = 0.20
End If

```

Select Case

Оператор Select Case використовується для вибору однієї з декількох груп операторів, залежно від значення логічного виразу.

```

Select Case вираз
Case значення1
    оператор(и)
Case значення 2
    оператор(и)
Case Else
    оператор(и) за замовчанням
End Select

```

Конструкція Case Else використовується для вибору операторів, які слід виконувати, якщо вираз не прийняв жодного з перерахованих значень.

Приклад. Оператор Select Case:

```

Commission = 0.15
Select Case Commission
Case .05
    Bonus = 0
Case .10
    Bonus = 500
Case .15
    Bonus = 1000
Case .20
    Bonus = 1500
Case Else

```

```
msgbox ("Invalid Commission Value")
End Select
```

5.3. Циклічні мовні конструкції

В багатьох програмах деякі рядки коду повинні виконуватись не один, а декілька (багато) разів. Мова VBA має оператори циклів, що дозволяють виконувати потрібний фрагмент коду необхідну кількість разів, або поки дотримується певна умова.

При цьому, кількість повторень групи операторів може бути:

- вказана, наперед відома кількість разів;
- не вказана, до тих пір, поки дотримується вказана умова;
- не вказана, до тих пір, поки не виконається вказана умова.

VBA має наступні конструкції операторів циклу:

- For...Next
- Do...Loop
- While...Wend

Цикл For...Next

Оператор For ... Next виконує групу операторів вказану кількість разів. Кількість повторень визначає лічильник циклу – змінна, яка автоматично збільшується (або зменшується) після кожного виконання циклу. Оператор For ... Next має наступний синтаксис:

```
For лічильник = початкове_значення To кінцеве_значення
    оператор (и)
Next
```

Лічильник циклу та його межі мають бути цілими числами. Початкове й кінцеве значення лічильнику задаються числами, або змінними.

Під час першого виконання циклу, лічильник приймає початкове значення. Після кожного виконання циклу, лічильник автоматично збільшується на одиницю. Цикл повторюється до тих пір, поки лічильник не перевищує кінцеве значення. Конструкція Next позначає кінець оператора циклу.

Приклад. Застосування оператора For ... Next:

```
Sub Дні_тижня()
    For i = 1 To 8
        Select Case i
            Case 1
                День = "Понеділок"
            Case 2
                День = "Вівторок"
            Case 3
                День = "Середа"
            Case 4
                День = "Четвер"
            Case 5
```

```

        День = "Пятниця"
    Case 6
        День = "Субота"
    Case 7
        День = "Неділя"
    Case Else
        День = "Немає варіантів"
    End Select
    MsgBox (День)
Next i
End Sub

```

Зверніть увагу, що змінна «i» автоматично збільшується при досягненні завершення циклу. Немає необхідності її змінювати окремою командою. За замовчанням, шаг зміни становить +1. Однак, його можна змінити, якщо вказати не обов'язкове ключове слово Step та величину шагу. Значення шагу може бути як додатнім, так й відємним, що дозволяє тож створювати цикли для обрахунку у зворотньому напрямку.

Do...Loop

Оператор циклу Do ... Loop повторює групу операторів доти, поки виконується деяка умова, або поки не буде виконана вказана умова. При цьому використовуються ключові слова While та Until. У першому випадку цикл виконується, поки умова істинна, в другому – поки умова не стане істинною.

При цьому умова може знаходитись як на початку, так й в кінці циклу (цикл з пред- та постумовою). Якщо умова знаходиться в кінці циклу, група операторів завжди виконається хоч би один раз.

Оператор Do ... Loop з ключовим словом while може приймати одну з двох форм:

```

Do While умова
    оператор (и)
Loop

```

або

```

Do
    оператор (и)
Loop While умова

```

У випадку з ключовим словом Until:

```

Do Until умова
    оператор (и)
Loop

```

або

```

Do

```

оператор (и)
Loop Until умова

Для дострокового виходу з циклу використовується оператор Exit Do.

Приклад. Застосування оператора Do...Loop:

```
Counter=100
Do While Counter > 0
  оператор (и)
  Counter = Counter - 1
Loop
```

While...Wend

Оператор While...Wend аналогічний циклу Do...Loop з передумовою:

```
While умова
  оператор (и)
Wend
```

При роботі з циклами Do ... Loop та While ... Wend є вірогідність входу в нескінченний цикл (заціклення), якщо значення заданої умови не може змінитись, наприклад:

```
Counter = 99
Do Until Counter = 0
  оператор (и)
  Counter = Counter - 2
Loop
```

Змінна Counter ніколи не прийме значення 0, а тіло циклу повторюватиметься нескінченну кількість разів. Виходом з цієї ситуації є умова, що має вигляд:

```
Counter = 99
Do Until Counter <= 0
  оператор (и)
  Counter = Counter - 2
Loop
```