

## Лабораторна робота №5

### Умовний оператор. Розгалуження програми.

#### Операції над змінними типу Boolean.

Мета роботи: здобуття навичок використання умовних операторів для організації розгалуження програми.

### Теоретичні відомості

Умовні оператори забезпечують виконання одного з основних методів програмування - розгалуження програми. Умовний оператор, звичайно, включає деяку умову та варіанти продовження програми залежно від того виконується чи не виконується поставлена умова.

#### 5.1. Умовні оператори в Pascal

TurboPascal включає 2 варіанти операторів умовного розгалуження програми

##### 5.1.1. Оператор If.

Формат:

```
If <expression> then <statement>;
```

або

```
If <expression> then <statement1> else <statement2>;
```

**<expression>** - вираз, що своїм результатом має значення типу Boolean - **"True"** або **"False"**.

**<statement>** - оператор, в тому числі складний оператор.

Якщо значення виразу - **"True"** (істина), то виконується оператор, що стоїть після слова **then**, якщо ж значення **"False"**, то оператор не виконується. В такому випадку виконується або оператор, що стоїть після слова **else (<statement2>)**, якщо він існує, або наступний оператор програми. Якщо використано другу форму формату команди, то після

виконання оператора (<statement1>), управління передається наступному оператору програми.

Приклад:

```
If x < 1.5 Then z := x + y
Else If x < 2.5 Then z := 1.5
      Else Begin
        WriteLn('X >= 2.5');
        x := 0;
      end;
```

На місці <expression> може також знаходитись змінна типу Boolean, або складений логічний вираз. Складений логічний вираз може бути створений з логічних виразів та логічних змінних, з'єднаних між собою знаками логічних операцій: **AND** ( і ), **OR** ( або ) та **NOT** ( ні ). Операція **NOT** є унарною і має перевагу над рештою операцій (аналог - унарний знак "-" мінус). Операція **AND** називається операцією логічного множення і має перевагу над операцією **OR** - операцією логічного додавання. Дію операцій наведено в табл. 5.1

Таблиця 5.1 Варіанти дії логічних операторів TurboPascal

A	B	A AND B	A OR B	NOT B
True	True	True	True	False
True	False	False	True	True
False	False	False	False	True

Приклад:

```
If x>=a and x<=b then
WriteLn('X належить проміжку ['a,b']')
```

При використанні умовного оператора з декількома рівнями вкладеності може відбутися порушення однозначності:

Так у виразі

```
If Expr1 then
  if Expr2 then
    Stmt1 else Stmt2
```

може видатись незрозумілим до якого оператора If відноситься else. В таких випадках необхідно використовувати складений оператор - брати внутрішній оператор **If** зі всією структурою в логічні дужки **Begin** - **End**.

## 5.1.2. Оператор Case

Формат

```
Case <expression> of
  case1: statement;
  ...
  caseN: statement;
end;
```

або

```
Case <expression> of
  case1: statement;
  ...
  caseN: statement;
else statement;
end;
```

Оператор **Case** є своєрідним узагальненням оператора **If** на випадок кількості можливих варіантів більше 2. Аргумент **<expression>** повинен бути цілого або символічного типів, або обмеженого чи перелічувального типу, побудованого на їх основі.

Значення **<expression>** порівнюється по чергово з кожним зі значень (**caseI**) до тих пір, поки значення не співпадуть. При співпаданні значення виконується відповідний оператор (**statement**) і обробка оператора **Case** закінчується. У випадку, якщо **<expression>** не співпадає ні з одним значенням (**caseI**), то виконується оператор після слова (**else**), якщо він є і обробка оператора закінчується.

Приклад:

```
case Ch of
  'A'..'Z', 'a'..'z': WriteLn('Letter');
  '0'..'9':           WriteLn('Digit');
  '+', '-', '*', '/': WriteLn('Operator');
else WriteLn('Special character');
end;
```

На місці кожного оператора (**statement**), що виконується при тому чи іншому значенні (**caseI**) може стояти як одиничний оператор так і складний оператор обмежений логічними дужками **Begin - End**.

При використанні оператора **Case** необхідно керуватися такими правилами:

- значення виразу-селектора **<expression>**, повинні належати до одного з дискретних типів; для цілого типу вони повинні лежати в діапазоні - 32768..32768;
- усі константи (**caseI**) повинні бути сумісні за типом, сумісний з типом виразу **<expression>**;
- усі константи у виразах повинні бути унікальними (не повинні повторюватись) в межах одного оператора **Case**; якщо константи представлені діапазонами, то діапазони не повинні пересікатися і містити констант, вказаних в іншій альтернативі.

## 2. Умовні оператори в VBA

VBA має аналогічний набір умовних операторів.

### 2.1. Оператор If...Then...Else

За своїм призначенням та використанням оператор схожий до відповідного оператора TurboPascal, однак існують деякі відмінності.

Так в VBA оператор **If** може використовуватись у спрощеній однорядковій формі, яка має синтаксис:

```
If <expression> Then <statement1> [else <statement2>]
```

Квадратні дужки означають, що відповідна частина є необов'язковою.

Спрощена форма оператора може використовуватись тільки тоді коли уся фраза умовного вибору поміщається в одному рядку програми. Якщо результатом виразу **<expression>** є **TRUE**, то виконується оператор **<statement1>** в іншому випадку виконується оператор **<statement2>**.

Якщо необов'язкова частина не вказана, то програма відразу переходить до обробки наступного рядка програми.

Повна форма оператора **If** має синтаксис:

```
If <expression1> Then  
<statement1>  
[ElseIF <expression2> Then  
<statement2>[...]  
[Else <expression_n>
```

**<statement\_n>]**

**End If**

У випадку, якщо **<expression1>** має результат **TRUE**, то виконується **<statement1>**, в іншому випадку аналізується вираз **<expression2>**, якщо він вказаний. Оператор може містити декілька фраз **ElseIF** **<expression>** вкладених одна в одну. Оператор обов'язково закінчується фразою **End If**.

Як і в інших мовах програмування, щоб задати умову можна використовувати як просту умову, так і складний вираз, що об'єднує умови за допомогою символів логічних операцій, наведених в таблиці 5.2.

## 2.2. Оператор Select Case

За призначенням оператор є аналогом оператора **Case TP**, і має синтаксис:

**Select Case <expression>**

**Case <expression-1>**

**<statement-1>**

**[Case <expression-2>**

**<statement-2>]**

...

**[Case Else**

**<statement-n>]**

**End Select**

В якості **statement** може використовуватись як одиничний оператор так і набір операторів.

Таблиця 5.2 Варіанти дії логічних операторів VBA

A	B	A AND B	A OR B	NOT B	A EQV B	A IMP B	A XOR B
True	True	True	True	False	True	True	False
True	False	False	True	True	False	False	True

False	True	False	True	False	False	True	True
False	False	False	False	True	True	True	False

### **Завдання на лабораторну роботу.**

Скласти та відлагодити програму для вводу значень масиву  $A[n,3]$ , та сортування його таким чином, щоб виконувалась нерівність  $A[i-1,1] \leq A[i,1] \leq A[i+1,1]$  не порушуючи однозначного зв'язку всередині множин  $A[i,1]$ ,  $A[i,2]$ ,  $A[i,3]$ .

Рекомендована література: [1], [2], [4], [5], [6], [7], [8]