Оператори за управление хода на програмата

I. Оператори за разклонение – това са вече познатите ви оператори IF Then Else и оператора за многовариантен избор Case

1. **Условен оператор –** оператор разклоняващ изчислителния процес в две посоки в зависимост от стойността на условие
2. **Видове условни оператори Във Visual Basic -** оператора If има три разновидности:

"Едноредов" оператор If – в този случай командата се изпълнява, ако условието се оцени като Истина (True).

"Многоредов" оператор If – в този случай имаме възможност да изпълним блок от команди, ако условието се оценя като True. За да отбележим края на блока команди използваме запазените думи End If.

"Многоредов" оператор If, който съдържа много блокове от команди – в този случай програмния поток се променя след оценяването на едно основно условие и на условията, посочени в ElseIf и Else.

Най-простия случай на употреба на оператора If би имал следния вид:

- **Оператор If-Then (кратка форма на условен оператор)**

If (условие) Then оператор

Условието е логическа променлива или логически израз. Първо се изчислява стойността на условието. Оператора след Then се изпълнява при стойност на условието True. При стойност False се преминава към следващия оператор. Когато след Then се налага използването на няколко оператора, тогава се преминава към друга форма на условния оператор

- **Оператор If-Then-End If (когато след Then има повече от един оператори)**

If Условие Then

Оператор1

Оператор2

.....

End If

- **Оператор If-Then-Else (Ако-Тогава-Иначе) пълна форма на условен оператор– тази форма позволява да се извършат действия и в двата случая.**

If Условие Then Оператор1	АКО условие ТО блок от команди 1
Else: Оператор2	ИНАЧЕ блок от команади 2
End If	КРАЙ НА АКО

Както при другите езици за програмиране, така и тук, първо се изчислява стойността на условието. При True – се изпълнява оператора след Then. При стойност False – се изпълнява оператора след Else, като се пропуска този след Then.

Между редовете на If-Then-End If If-Then-Else може да се записват повече от един оператори

- **Оператор If-Then-ElseIf – вложен условен оператор**

If Условие1 Then	АКО условие 1 ТО
Оператор1	блок от команди 1
Else If Условие2 Then	ИНАЧЕ АКО условие 2 ТО
Оператор2	блок от команади 2
End If	КРАЙ НА АКО

Съществува вероятност да не се изпълни нито един оператор, ако условията са False. Ако има повече условия бихме могли да влагаме условни оператори.

If Условие1 Then	Оператор1

Else If Условие2 Then	**блок от команди 1**
Оператор2	**ИНАЧЕ АКО условие 2 ТО**
Else If Условие3 Then	**блок от команади 2**
Оператор3	**ИНАЧЕ АКО условие 3 ТО**
End If	**блок от команади 3**
	КРАЙ НА АКО

АКО условие 1 ТО

И отново би могло да не се изпълнят оператори Ето защо

If Условие1 Then	**АКО условие 1 ТО**
Оператор1	**блок от команди 1**
Else If Условие2 Then	**ИНАЧЕ АКО условие 2 ТО**
Оператор2	**блок от команади 2**
Else	**ИНАЧЕ**
Подразбиращ се Оператор	**блок от команади 4**
End If	**КРАЙ НА АКО**

Не е желателно да се предприема прекомерно влагане на условни оператори. Осъществяването на такава стъпка говори за проблем със съставянето на алгоритъма и не толкова с неговата реализация.

3. Оператор за многовариантен избор

Оператор Select Case – когато се налага проследяване на много условия и влагането на оператор ElseIf изглежда тромаво. Действието на този оператор е аналогично на вложените условни оператори. Обратното също е вярно. Употребата му се налага при влагане на няколко условни оператора, ц цел по-лесно разчитане на програма и допускане на по-малко грешки

```
Select Case Променлива
        Case X
                Оператор1
        Case Y

                Оператор2
        Case Z

                Оператор3
End Select
```

Където X,Y и Z са стойности на променливата

- **Използване на оператор select Case съвместно с оператори за сравнение**

```
Select Case Day
        Case is>15
            Оператор1
        Case is>10
            Оператор2
End Select
If day>15 Then
        Оператор1
ElseIf Day>10 Then
        Оператор2
EndIf
```

II. Оператори за цикъл

1. **Определение – Цикълът представлява част от алгоритъм, която записана веднъж, се повтаря многократно. Всеки цикъл би могъл да се разгледа като съставен от четири основни части:**

2. **Части на цикъла**

   - Инициализация – в нея задаваме начални стойности на участващите в цикъла величини
   - Тяло на цикъла – това са операторите, които се изпълняват многократно.
   - Актуализация – тук задаваме нови стойности на някои от участващите в тялото на цикъла величини и ги подготвяме за следващото изпълнение
   - Условие за край на цикъла – това е логически израз, който прекратява действието на цикъла

За да се изпълнява цикъла задължително трябва да се дефинира променлива, която да променя стойността си при изпълнение на цикъла. Нарича се управляваща променлива.

3. **Видове цикли:**

   - Цикъл с предусловие – при него условието за край на цикъла се проверява най-напред и в зависимост от това, дали е вярно или не, тялото на цикъла се изпълнява. Не е трудно да преценим, че такъв цикъл може да не се изпълни и нито веднъж, ако още в началото условието не е изпълнено.
   - Цикъл с постусловие – при него първо се изпълнява тялото на цикъла, а чак след това се проверява верността на условието за край. Такъв цикъл се изпълнява минимум един път.
   - Цикъл с управляваща променлива (броячен цикъл) – това е цикъл, при който задаваме начална и крайна стойност на управляващата променлива и той се изпълнява определен брой пъти.

Общо взето нещата във VisualBasic по отношение на циклите не са по-различни от другите езици за програмиране. И тук има три основни вида цикли – с предусловие, с постусловие и броячен цикъл. По-голямо е разнообразието на разновидностите на съответните оператори.

4. **Цикли с предусловие -** Това са цикли, при които първо се проверява дали е изпълнено някакво условие. Ако това условие е изпълнено се изпълнява тялото на цикъла. В противен случай програмата продължава да се изпълнява от следващата след цикъла инструкция**.**

Във VisualBasic има два оператора за цикъл с предусловие: Do While и Do Until.

➢ Оператор Do While

Общият вид на оператора е:

Do While Условие

Инструкции

Loop

Условието трябва да е променлива или израз, чиято стойност е True или False. Тялото на цикъла може да се състои от една или повече инструкции.

Цикълът се изпълнява докато стойността на условието е True.

Пример;

K=0 - инициализация

Do While K<>0 - условие

K=K+1 - актуализация

Оператор - тяло

Loop

Такъв цикъл може и да не се изпълни нито веднъж, ако първоначалната стойност на условието е False.

➢ Оператор Do Until

Общият вид на оператора е:

Do Until Условие

Инструкции

Loop

Условието трябва да е променлива или израз, чиято стойност е True или False. Тялото на цикъла може да се състои от една или повече инструкции.

Цикълът се изпълнява докато стойността на условието е False. Когато тя стане True, цикълът спира.

Такъв цикъл може и да не се изпълни нито веднъж, ако първоначалната стойност на условието е True.

5. Цикли с постусловие

➢ Оператор Do-Loop While

Общият вид на оператора е:

Do

Инструкции

Loop While Условие

Условието трябва да е променлива или израз, чиято стойност е True или False. Тялото на цикъла може да се състои от една или повече инструкции.

Първо се изпълнява тялото на цикъла и след това се проверява условието. Ако стойността му е True, цикълът се изпълнява отново. Ако стойността е False, цикълът се прекратява.

Цикълът Do-Loop While се изпълнява поне веднъж.

➢ Оператор Do-Loop Until

Общият вид на оператора е:

Do

Инструкции

Loop Until Условие

Условието трябва да е променлива или израз, чиято стойност е True или False. Тялото на цикъла може да се състои от една или повече инструкции.

Първо се изпълнява тялото на цикъла и след това се проверява условието. Ако стойността му е False, цикълът се изпълнява отново. Ако стойността е True, цикълът се прекратява. Т.е. цикълът спира, когато условието стане вярно.

Цикълът Do-Loop Until се изпълнява поне веднъж.

	Докато условието е True	Докато условието е False
Изпълнява цикъл поне веднъж	Do Loop While Условие	Do Loop Until Условие
Изпълнява цикъл 0 или повече пъти	Do While Условие Loop	Do Until Условие Loop

6. Броячен цикъл FOR

Общият вид на оператора е:

For Counter = Start To End Step X

Инструкции

Next Counter

Тук Counter е управляващата променлива. Start и End задават съответно началната и крайната стойност на управляващата променлива. X е стъпката на изменение на управляващата променлива. Изпълнението на оператора е аналогично на оператора For в другите езици за програмиране.

По подразбиране броячът на цикъла се увеличава или намалява с 1. Така че ако изпуснем частта Step X стъпката е 1.

```
For X = 5 To 10
Print X
Next X
```

a. Действие на цикъла:

Първия ред инструктира: Създай променлива с име Хи й дай стойност 5.

Изпълнявай инструкциите докато стойността на X се променя 5, 6, 7, 8, 9, 10

Втория ред е мястото за операторите, които се повтарят (те могат да бъдат повече от един)

Третия ред изчислява следващата стойност на X

Броячът не бива да променя стойността си в тялото на цикъла.

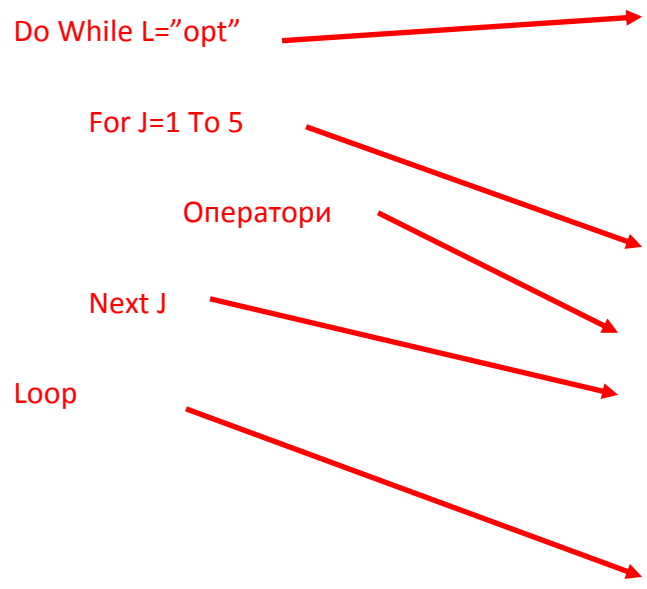
b. Броене напред и назад – когато се налага стъпката на броене би могла да бъде различна от 0. В тези случаи се използва служебната дума Step

```
For X = 15 To 10 Step -1.5
Print X
Next X
```

Когато се използва този вид цикъл НИКОГА не променяйте в тялото на цикъла стойността на управляващата променлива.

Този вид цикъл се използва винаги, когато се изисква даден цикъл да се изпълни определен брой пъти.

❖ Използване на вложени цикли – вложеният цикъл завършва пръв изпълнението си. Например:



I ред – създай променлива с име L и провери дали нейната стойност е равна на “opt”. Ако е така, премини към изпълнението на втория ред

II ред – създай променлива J и й задай стойност 1

III ред – команди

IV ред – Увеличи стойността с единица и се върни на първи ред от цикъла и така докато J не стане по-голямо от 5

V ред – край на външния цикъл

Цикълът For Next се изпълнява целият когато програмата преминава през тялото на Do While

❖ Проверка на действието на вложени цикли – за да е читаема програмата е здължително при писане на програмния код да се правят отстъпи. Пример:

```
Do While Name = “Sam”
Do
For K=20 To 50 Step 10
Do
Do Until Sex = “Male”
C Change some variables here
Loop
Loop While Age > 21
Next K
Loop Until LastName = “Doe”
Loop
```

```
Do While Name = “Sam”
    Do
        For K=20 To 50 Step 10
            Do
                Do Until Sex = “Male”
                    C Change some variables here
                Loop
            Loop While Age > 21
        Next K
    Loop Until LastName = “Doe”
Loop
```

❖ Незабавно напускане на цикъл – един цикълтип Do продължава да се изпълнява докато условието за край не стане True (False). Цикъл с брояч не прекратява действието си докато не изпълни определенят му брой итерации.

За да се прекрати изпълнението на цикъл тип Do се използва командата Exit Do. Например:

```
X=0
Do While X<6
  X = X + 1
  If X=4 Then Exit Do
Loop
```

За да се излезе принудително от цикъл For Next се изпълнява командата Exit For

```
For Y = 1 To 100
  If Y = 50 Then Exit For
Next Y
```

Ако използвате команда Exit Do/For в цикъл, вложен в друг цикъл, тези команди карат да се прекрати действието на текущия цикъл, като управлението се предава на външния цикъл