

VB-2008, instrukcja warunkowa

Jeżeli zaistnieje konieczność sprawdzenia, czy pewne wielkości spełniają założone kryteria, można skorzystać z instrukcji warunkowej **If-Then**. W prostszym przypadku instrukcja ta ma postać:

```
If sprawdzany_warunek Then
'jeśli sprawdzany_warunek jest prawdą, to wykonaj
'ciąg instrukcji znajdujący się w tym miejscu
End If
```

Warunek może mieć postać np. **a>b, c<3, d=1, e=True**. W przykładach tych **a, b, c, d, e** są nazwami zmiennych, natomiast **True** jest wartością logiczną **Prawda**. Bardziej rozbudowana wersja instrukcji przyjmuje postać:

```
If warunek Then
'instrukcje wykonywane jeśli warunek jest prawdą
Else
'instrukcje wykonywane w przeciwnym wypadku
End If
```

Najprostsza postać:

```
If warunek Then 'instrukcje wykonywane jeśli warunek jest prawdą
```

Należy zwrócić uwagę na to, że jeśli jest do wykonania tylko jedna instrukcja, to może ona wystąpić bezpośrednio po słowie kluczowym **Then** i nie występuje wtedy **End IF**.

W przypadku zbadania bardziej złożonych warunków, można skorzystać z najbardziej rozbudowanej wersji, która wygląda tak jak w poniższym przykładzie (zaczepnięty z pomocy do VB2008 Express):

```
Dim liczba, cyfry As Integer
Dim tekst As String
liczba = 53
If liczba < 10 Then           'sprawdzenie czy wartość zmiennej<10
    cyfry = 1                 'jeśli tak, to zmienna cyfra=1
ElseIf liczba < 100 Then    'w przeciwnym razie, jeśli liczba<100
    cyfry = 2                 'podstaw wartość 2
Else                         'w przeciwnym razie
    cyfry = 3                 'podstaw wartość 3
End If
If cyfry = 1 Then tekst = "Jedna" Else tekst = "Więcej niż jedna"
```

W przykładzie zadeklarowano dwie zmienne całkowitoliczbowe (**Integer**), jedną zmienną typu ciąg znaków (**String**), do zmiennej **liczba** przypisano wartość liczbową 53. Z przykładu wynika, że słowo kluczowe **If** sprawdza pewien warunek, **Elseif** sprawdza inny warunek, jeśli obydwa warunki nie są prawdziwe, wykonane są instrukcje stojące po słowie kluczowym **Else**. Tę wersję można bardziej rozbudować przez wielokrotne stosowanie słowa kluczowego **Elseif**. Przed wykorzystaniem takiej wersji instrukcji warunkowej, należy ją odpowiednio zaprojektować, gdyż bardzo łatwo można uzyskać niewłaściwe działanie instrukcji.

VB-2008, pętla For

Czasami istnieje konieczność wielokrotnego wykonania tej samej instrukcji lub zestawu instrukcji. Na przykład należałoby napisać 1000 razy „Nie będę rozmawiać na lekcji”. Można sobie wyobrazić, że faktycznie napisze się tę frazę 1000 razy, ale bardziej rozsądnie byłoby wykorzystać pętlę **For** tak jak w poniższym przykładzie (pętlę stanowią instrukcje objęte klamrą):

```
FileOpen(1, "c:\kara.txt", OpenMode.Output) 'otwórz plik do zapisu.
{ For j As Integer = 1 To 1000
  Print(1, "Nie bede rozmawiac na lekcji.") 'zapisz zdanie do pliku.
Next j
FileClose(1) 'zamknij plik.
```

Zmienna **j** jest licznikiem pętli, czyli zlicza ile razy wykonane zostały polecenia wewnątrz pętli (w powyższym przykładzie polecenie **Print**). Początkową wartością przypisaną do **j** jest **1**, po dojściu do instrukcji **Next j**, licznik pętli automatycznie zwiększa się o jeden i pętla wykonywana jest ponownie. Dzieje się to tak długo, aż licznik pętli nie osiągnie wartości końcowej – w tym przykładzie **1000**. Utało się, że jako nazwy liczników pętli wykorzystuje się kolejne litery alfabetu: **i, j, k, l, m, n**. Często stosuje się tzw. zagnieżdżanie pętli. Polega to na tym, że jedna pętla znajduje się wewnątrz drugiej, może to wystąpić wielokrotnie. W powyższym przykładzie deklaracja typu zmiennej **j** (licznika pętli) następuje w samej pętli, może też wystąpić inny przypadek, kiedy licznik pętli jest deklarowany znacznie wcześniej, wówczas ta sama zmienna może być wykorzystywana wielokrotnie w różnych pętlach byle nie były zagnieżdżone w sobie. Stosując zagnieżdżanie bezwzględnie należy pamiętać, aby pętla zagnieżdżona zawierała się w całości w pętli nadrzędnej, czyli rozpoczyna się i kończy wewnątrz pętli nadrzędnej. Poniżej przedstawiono przykład prawidłowego i nieprawidłowego zagnieżdżenia pętli.

DOBRCZE

```
For i = 1 To 10
  { For j = 2 To 7
    'instrukcje do wykonania
  Next j
Next i
```

ŻŁE!

```
For i = 1 To 10
  { For j = 2 To 7
    'instrukcje do wykonania
  Next i
Next j
```

Jeśli licznik pętli nie ma się zwiększać o jeden, tylko np. o trzy, a w szczególności ma się zmniejszać, należy zastosować następującą konstrukcję:

```
For j = 1 To 1000 Step 3
```

lub

```
For j = 20 To 1 Step -1
```