Wydział Zarządzania AGH
Katedra Informatyki Stosowanej

Instrukcje sterujące

Programowanie komputerowe

Program wykładu

- Instrukcje **IF**
- Instrukcja **Case**

Instrukcje sterujące

Instrukcje sterujące umożliwiają kontrolę przebiegu programu.

Najbardziej elementarną instrukcją sterującą jest instrukcja warunkowa **If**. W języku VBA występuje ona w wielu wariantach składniowo-semantycznych.

Instrukcja If...

Instrukcję sterującą **If** wykorzystuje się do warunkowego wykonania fragmentu kodu programu.

Język VBA przewiduje dwojaki sposób zapisu tej instrukcji – blokowy oraz mniej popularny wierszowy.

1) postać wierszowa

If warunek **Then** polecenie [:polecenie_1]… [:polecenie_n]

2) postać blokowa

If warunek **Then**

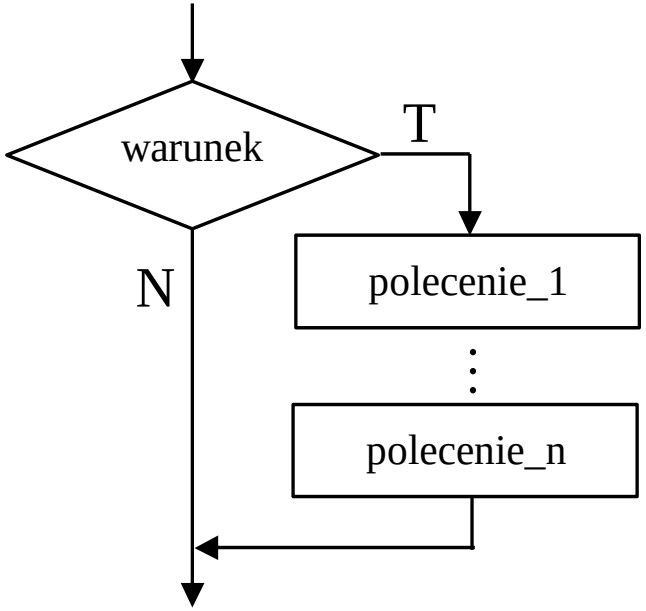
[polecenie_1]

…

[polecenie_n]

End If

Instrukcja If ...



Stosowanie zapisu wierszowego zaleca się wtedy, gdy wykonane ma być jedno polecenie lub grupa (2-4) krótkich poleceń. W pozostałych przypadkach powinno się stosować zapis blokowy.

Instrukcja If - przykłady

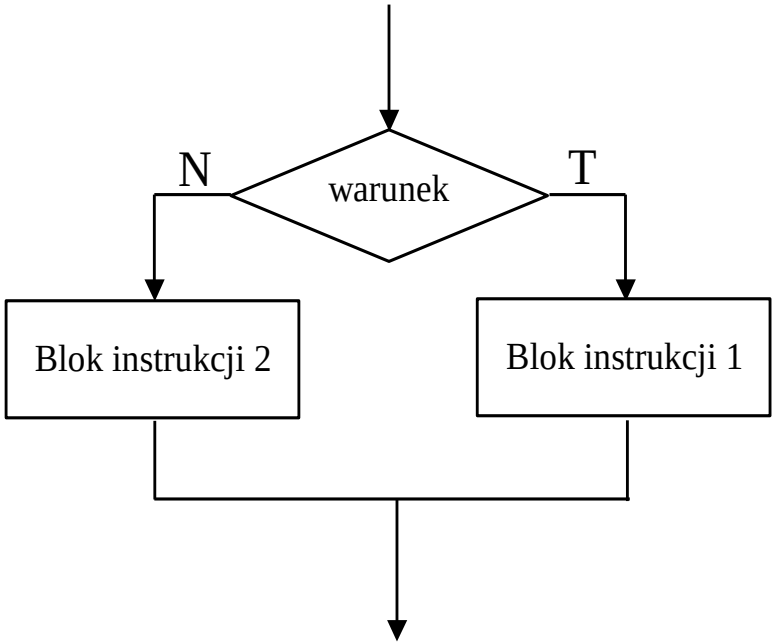

```
Sub Przyklad_6_2()
Dim Liczba As Integer
Liczba=InputBox("Podaj liczbę")

    If Liczba < 0 Then
        Liczba = -Liczba
    End If

    MsgBox "Moduł wynosi  " & _
                        Liczba
End Sub
```

Instrukcja If ...Then ... Else

Instrukcja **If**…**Then**…**Else** umożliwia wykonanie tylko jednego z dwóch bloków kodu, z których jeden występuje po słowie kluczowym **Then**, drugi po **Else**.

If…**Then**…**Else** może występować w postaci wierszowej i blokowej, aczkolwiek postać wierszowa jest rzadko stosowana ze względu na małą przejrzystość zapisu.

Instrukcja If ...Then ... Else



If warunek **Then**

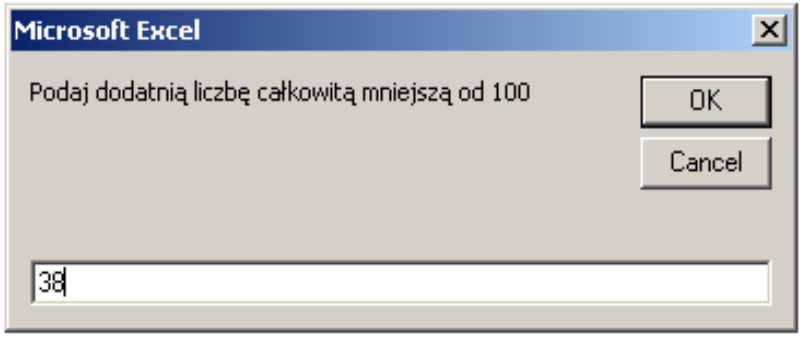
[blok_instrukcji_1]

Else

[blok_instrukcji_n]

End If

Instrukcja If ...Then ... Else




```
Sub Przyklad_6_4()
Dim Liczba As Byte
Liczba = InputBox("Podaj dodatnią liczbę
            całkowitą mniejszą od 100")
If Liczba < 10 Then
      MsgBox "Podano jednocyfrową liczbę"
Else
      MsgBox "Podano dwucyfrową liczbę"
End If
End Sub
```

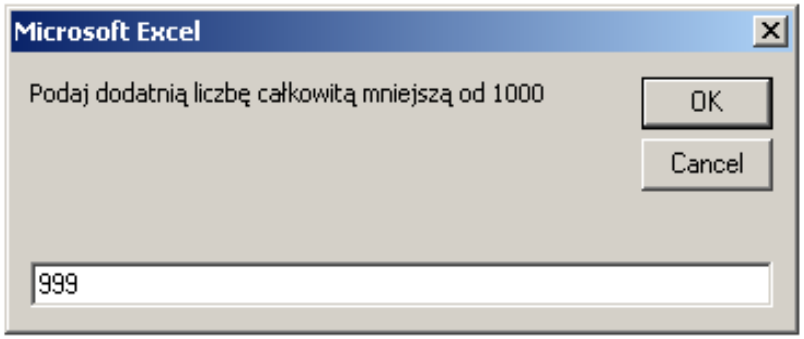
Instrukcja If... ElseIf... Else

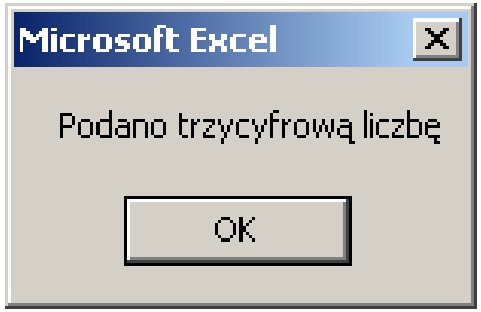
Instrukcja **If**… **ElseIf**… **Else** stanowi konsekwencję możliwości zagnieżdżania instrukcji **If**. Taka postać instrukcji sterującej daje programiście możliwość dodawania kolejnych warunków.

```
If warunek_1 Then
    [blok_instrukcji_1]
ElseIf warunek_2
Then
    [blok_instrukcji_2]
Else
    [blok_instrukcji_3]
End If
```

```
If warunek_1 Then
    [blok_instrukcji_1]
Else
    If warunek_2 Then
        [blok_instrukcji_2]
    Else
        [blok_instrukcji_3]
    End If
End If
```

Instrukcja If... ElseIf... Else






```
Sub Przyklad_6_5()
Dim Liczba As Integer
Liczba = InputBox("Podaj dodatnią liczbę
            całkowitą mniejszą od 1000")
If Liczba < 10 Then
        MsgBox "Podano jednocyfrową liczbę„
ElseIf Liczba < 100 Then
        MsgBox "Podano dwucyfrową liczbę„
Else
        MsgBox "Podano trzycyfrową liczbę„
End If
End Sub
```

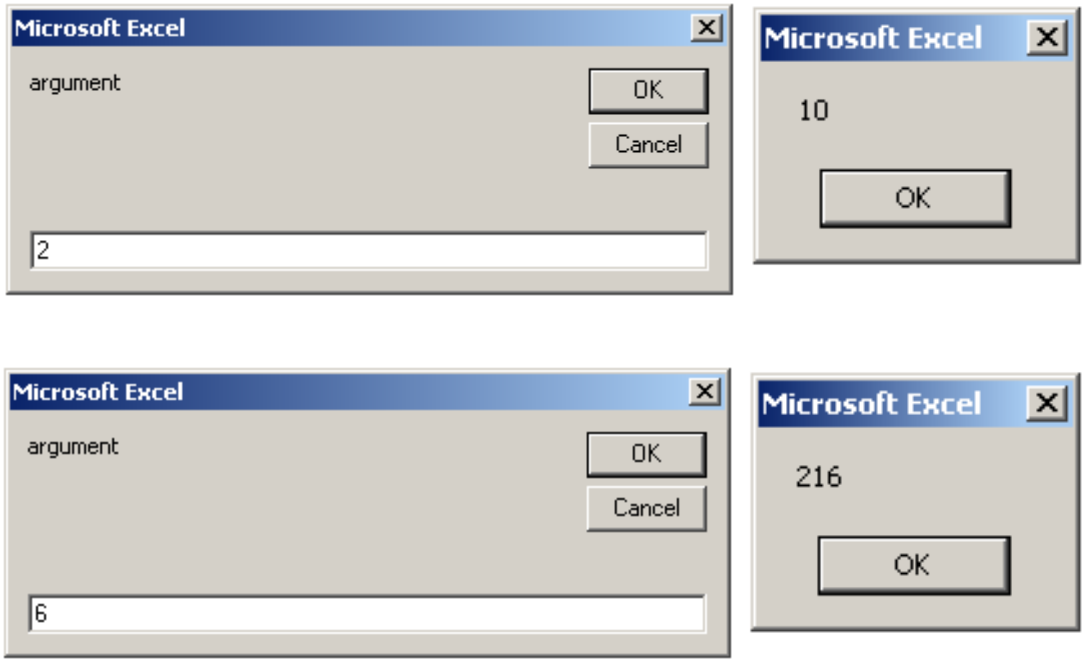
Zastosowanie operatorów logicznych

```
Sub Przyklad_1()
Dim Liczba As Byte
Liczba = InputBox("Podaj dodatnią
liczbę całkowitą")
If Liczba>= 3 And Liczba<=36 Then
        MsgBox „Podana liczba
        należy do  przedziału [3, 36]”
End If
End Sub
```

```
Sub Przyklad_2()
Dim Liczba As Byte
Liczba = InputBox("Podaj dodatnią
liczbę całkowitą")
If Liczba >= 3  Then
        If Liczba <= 36 Then
          MsgBox „Podana liczba
          należy do przedziału [3, 36]”
        End If
End If
End Sub
```

Zastosowanie operatorów logicznych

$$y = \begin{cases} x^3 + x & \text{dla } -5 \le x \le 5 \\ x^3 & \text{dla } -5 > x > 5 \end{cases}$$




```
Sub Przyklad_6_7()
Dim x As Integer
Dim y As Integer
x=InputBox("argument")
If x <= 5 And x >= -5 Then
        y = x ^ 3 + x
Else
        y = x ^ 3
End If
MsgBox y
End Sub
```

Instrukcje If...

$$sign(x) = \begin{cases} 1 & \text{dla } x > 0 \\ 0 & \text{dla } x = 0 \\ -1 & \text{dla } x < 0 \end{cases}$$

```
Sub Przyklad()
Dim X As Integer
Dim Sign As Byte
X = InputBox("Podaj liczbę")
If X > 0 Then
        Sign = 1
ElseIf X < 0 Then
        Sign = -1
Else
        Sign = 0
End If
MsgBox Sign
End Sub
```

```
Sub Przyklad()
Dim X As Integer
Dim Sign As Byte
X = InputBox("Podaj liczbę")
If X > 0 Then
        Sign = 1
Else
        If X < 0 Then
            Sign = -1
        Else
            Sign = 0
        End If
End If
MsgBox Sign
End Sub
```

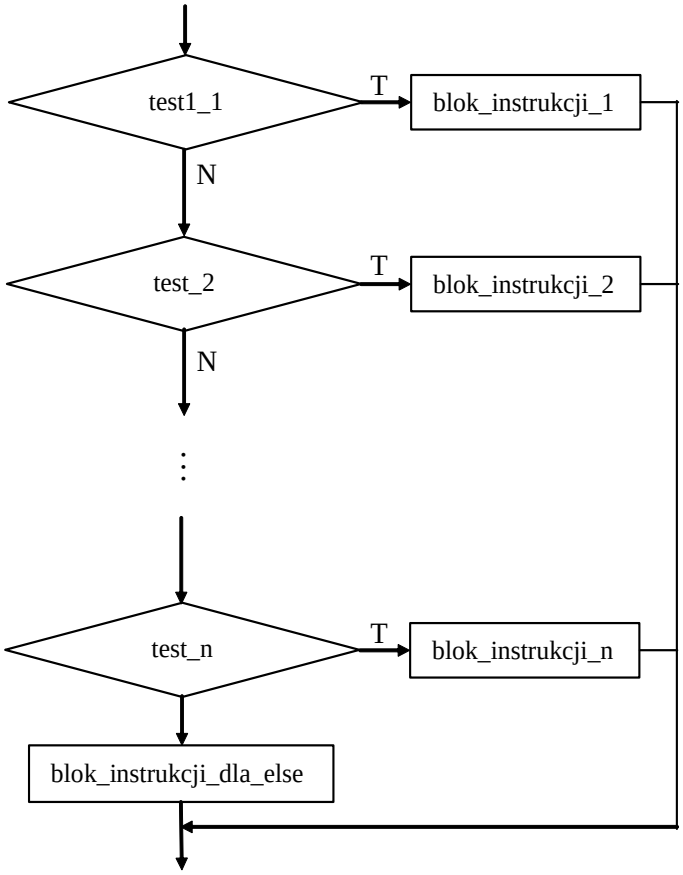
```
Sub Przyklad()
Dim Sign As Byte
Dim X As Integer
X = InputBox("Podaj liczbę")
If X < 0 Then
        Sign = -1
End If
If X = 0 Then
        Sing = 0
End If
If X > 0 Then
        Sign = 1
End If
MsgBox Sign
End Sub
```

Instrukcja Select Case (1)

Select Case jest instrukcją, która może zastapić wielokrotnie zagnieżdżone instrukcje **If**… **ElseIf** … **Else**, czyniąc kod bardziej czytelnym i efektywnym.

Istota działania instrukcji **Select Case** polega na możliwości dokonania wyboru pomiędzy różnymi wariantami przebiegu programu.

Instrukcja Select Case (2)



Select Case wyrazenie_testowe
[**Case** test_1
[blok_instrukcji_1]]
[**Case** test_2
[blok_instrukcji_2]]

.....

[**Case** test_n
[blok_instrukcji_n]]
[**Case Else**
[blok_instrukcji_dla_else]]
End Select

Instrukcja Select Case (3)

Wybór fragmentu programu do wykonania dokonywany jest na podstawie wyniku porównania pomiędzy *wyrażeniem*, a zbiorem wartości, jakie może ono przyjmować.

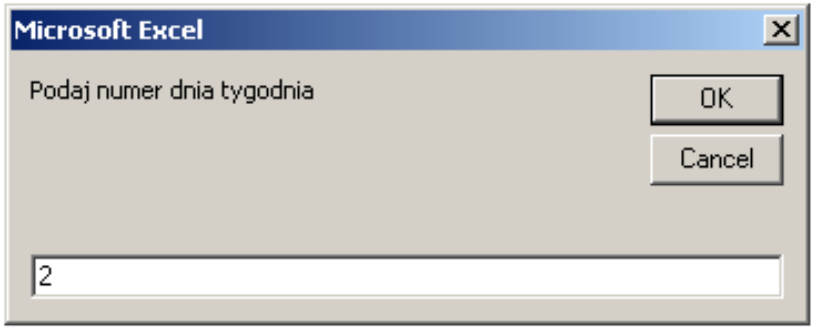
W przypadku kiedy porównanie zwróci wartość logicznej prawdy (**True**) wykonywany jest wskazany blok instrukcji, a następnie struktura **Select Case** kończy działanie (tzn. wykonywana jest instrukcja w następnej linii po słowach kluczowych **End Select**).

Instrukcja Select Case (4)

Porównania wykonywane są w kolejności ich wystąpienia. Należy zaznaczyć, iż wykonywany jest tylko jeden blok instrukcji, występujący po pierwszym porównaniu, które zwróciło wartość logiczną **True** – pozostałe porównania nie są wówczas analizowane. W przypadku kiedy żadne z porównań nie zwróci wartości **True**, podejmowane przez strukturę **Select Case** działanie uzależnione jest od wystąpienia opcjonalnego rozkazu **Case Else**.

Select Case - przykład

```
Sub Dni_Tygodnia()
    Dim NrDnia As Byte
    NrDnia = InputBox("Podaj numer dnia tygodnia")
    Select Case NrDnia
      Case 1
         MsgBox "Poniedziałek„
      Case 2
         MsgBox "Wtorek„
      Case 3
         MsgBox "Środa„
      Case 4
         MsgBox "Czwartek„
      Case 5
         MsgBox "Piątek„
      Case 6
         MsgBox "Sobota„
      Case 7
         MsgBox "Niedziela„
      Case Else
         MsgBox „Nie ma dnia oznaczonego podanym numerem”
    End Select
End Sub
```



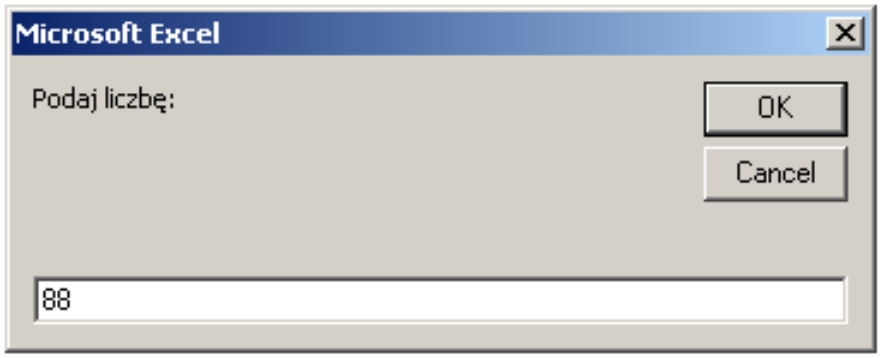
Instrukcja Select To

Instrukcja **Case To** sprawdza, czy wartość wyrażenia znajduje się w zadanym przedziale (zakresie).

Np.

Case 1 **To** 90	przedział <1, 90>
Case ”łyżka" **To** "talerz”	przedział ⟨”łyżka", "talerz”⟩ w porządku alfabetycznym

Case To - przykład




```
Sub Liczba_Cyfr()
    Dim s As String
    x = InputBox("Podaj liczbę:")
    Select Case x
        Case 0 To 9
            s = "Jednocyfrowa„
        Case 10 To 99
            s = "Dwucyfrowa„
        Case 100 To 999
            s = "Trzycyfrowa„
    End Select
    MsgBox s
End Sub
```

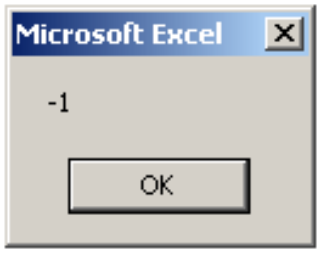
Instrukcja Case Is

Instrukcji **Case Is** można używać do porównywania wartości *wyrażenia* z innymi wartościami tego samego typu, za pomocą relacji porządkowych lub znaków równości i różności. Słowo kluczowe **Is** używane w instrukcji **Select Case** nie jest tożsame z operatorem porównania **Is**.

Np. **Case Is** <> 10

sprawdzenie, czy wartość *wyrażenia* jest różna od 10

Case Is - przykład




```
Sub Znak()
    Dim X As Integer
    Dim Sign As Byte
    X = InputBox("Podaj liczbę")
    Select Case X
        Case Is > 0
            Sign = 1
        Case Is < 0
            Sign = -1
        Case Is = 0          ' lub po prostu Case 0
            Sign = 0
    End Select
    MsgBox Sign
End Sub
```

Select Case, Case To, Case Is

Visual Basic dopuszcza łączenie wszystkich konstrukcji Select Case.

Zadanie:

Oblicz wysokość należnego podatku, zgodnie z podaną skalą podatkową.

Podstawa obliczenia podatku w PLN		**Podatek wynosi**
ponad	**do**	
	37.024	19% podstawy obliczenia minus kwota **530 zł 08 gr**
37.024	74.048	6.504 zł 48 gr + 30% nadwyżki ponad 37.024 zł
74.048		17.611 zł 68 gr + 40% nadwyżki ponad 74.048 zł

Select Case, Case To, Case Is

```
Sub Podatek()
Dim Podstawa As Single
Dim Podatek As Single
    Podstawa = InputBox("Podaj podstawę opodatkowania")
    Select Case Podstawa
        Case 0 To 2790
            Podatek = 0
        Case 2790 To 37024
            Podatek = Podstawa * 0.19 - 530.08
        Case 37024 To 74048
            Podatek = 6504.48 + 0.3 * (Podstawa - 37024)
        Case Is > 74048
            Podatek = 17611.68 + 0.4 * (Podstawa - 74048)
    End Select
    MsgBox "Wysokość obliczonego podatku: " & Round(Podatek, 2)

End Sub
```

Select Case, Case To, Case Is