



MS Excel

VBA: poziom I

Materiały szkoleniowe

Wersja demonstracyjna



KONTAKT

Adres

Expose sp. z o. o.
ul. Skierniewicka 10a
01-230 Warszawa



Telefon

+ 48 22 465 88 88
+ 48 22 240 19 99

Online

biuro@expose.pl
www.expose.pl
www.chcesieuczyc.pl

Konspekt kursu Excel Programowanie w VBA: poziom I

1. Rejestrowanie makr
 - Zapisywanie i uruchamianie makr
 - Przypisywanie makr do przycisków i klawiszy skrótów
 - Bezpieczeństwo makr
 - Budowa makr
2. Edytor VBA
 - Budowa edytora Visual Basic
 - Znaczenie kolorów w edytorze
 - Przeglądarka obiektów
 - Moduły
 - Komentowanie kodu
3. Zmienne
 - Deklarowanie i nazwy zmiennych
 - Typy zmiennych (tekstowe, liczbowe, data i godzina, logiczne i inne)
 - Zakresy zmiennych
 - Jawne i niejawne deklarowanie zmiennych
 - Deklaracja i używanie stałych
 - Nazwy zmiennych
4. Operatory
 - Przypisania, matematyczne, tekstowe, logiczne
 - Operacje na tekstach
 - Praktyczne przykłady stosowania operatorów
5. Komunikacja z programem
 - Funkcja MsgBox
 - Funkcja InputBox
 - Metoda InputBox
 - Aktualnie zaznaczona komórka
 - Jak wybrać/zaznaczyć komórkę?
6. Instrukcje VBA
 - Instrukcja warunkowa IF i jej warianty
 - Instrukcja SELECT CASE
 - Pętla For ... Next
 - Pętla For Each ... Next
 - Pętla Do While ... Loop
 - Obsługa błędów

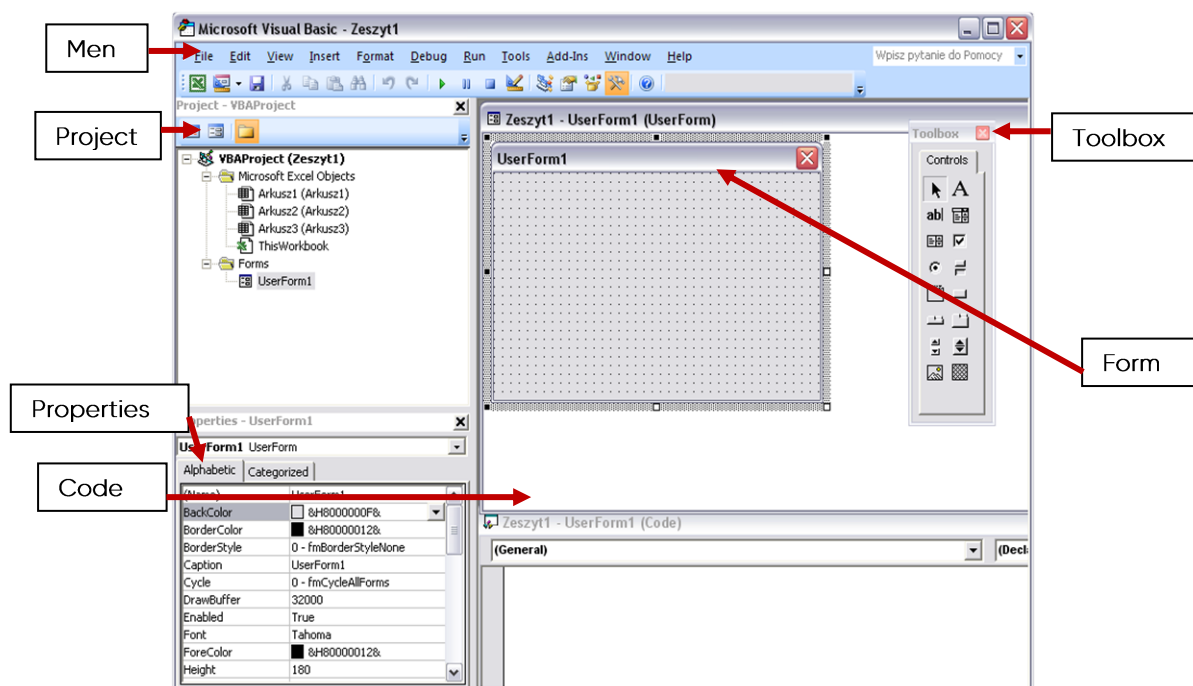
- 7. Funkcje VBA
 - Wstęp do funkcji w VBA
 - Korzystanie z gotowych funkcji VBA
 - Tworzenie własnych funkcji
 - Ograniczenia funkcji
- 8. Model obiektowy MS Excel
 - Czym są obiekty?
 - Obiekt Application
 - Obiekt Workbook i kolekcja Workbooks
 - Obiekt Worksheet
 - Obiekt Range
 - Obiekt Cells

1 Edytor VBA

Podstawy edytora Visual Basic

Dzisiejsze języki programowania znacznie się różnią od języków programowania sprzed kilku lat. Dzięki systemowi operacyjnemu Microsoft Windows, który używa graficznego interfejsu, powstało wiele narzędzi, bardziej zaawansowanych, które wykorzystują interfejs graficzny. Programy te, choć często spełniają taką samą rolę jak ich poprzednicy, są ze względu na swój wygląd, bardziej przyjazne dla użytkownika. Tworzenie programów z graficznym interfejsem i wykorzystujących zdarzenia generowane przez użytkownika stało się czasochłonne w językach strukturalnych. Wymusiło to rozwój języków programowania. Wprowadzenie wizualizacji do języków programowania znacznie uprościło proces tworzenia graficznego interfejsu aplikacji.

Visual Basic jest aplikacją Windows i sam wykorzystuje interfejs graficzny. Platforma projektowa i środowisko projektowe Visual Basic nosi nazwę Developer Studio. Składa się ono z wielu okien i narzędzi, które pozwalają (i ułatwiają) tworzyć rozbudowane aplikacje działające w środowisku Microsoft Windows. W fazie projektowania programu, główne okno Visual Basic zawiera kilka różnych okien. Każde z nich może być ukryte w miarę potrzeby. Każde z nich ma inne zadanie i jest wykorzystywane do innych czynności niezbędnych przy tworzeniu aplikacji.



Budowa edytora Visual Basic

Pasek menu jest paskiem rozwijanych, wielopoziomowych menu, z których można wybrać polecenia konieczne do ustawienia odpowiednich opcji zarówno tworzonej aplikacji, jak i samego Visual Basic.

Pasek narzędziowy Toolbar, jest szeregiem ikon, które tworzą skrót do odpowiednich pozycji menu. Zamiast rozwijać wielopoziomowe menu i wybierać odpowiednie elementy, można kliknąć ikonę, która spowoduje wykonanie tych samych czynności.

Okno Form (Object) jest głównym oknem aplikacji. W tym oknie tworzony jest cały wygląd aplikacji, rozmiar, kolor, rozmieszczenie elementów.

Okno Project jest oknem zawierającym informację o wszystkich częściach i plikach aktualnie tworzonej aplikacji.

Okno Toolbox jest zbiorem kontrolki, które można umieszczać na formie tworzonej aplikacji, aby wykonywały polecane im zadania.

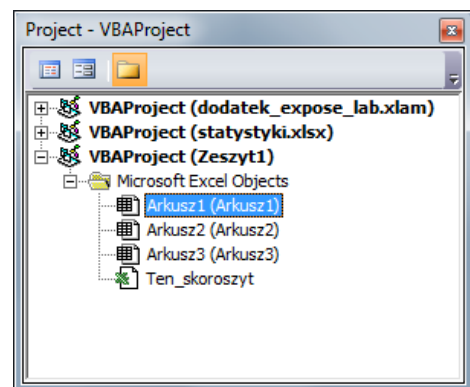
Okno Properties to okno, w którym można ustawić właściwości wykorzystywanych elementów, form, kontrolki, obiektów. Można tam ustawić cechy odpowiedzialne za wygląd, położenie, zachowanie poszczególnych elementów. Aby wybrać odpowiednią wartość właściwości, należy wpisać ją w odpowiednie okno lub wybrać jedną z pozycji rozwijanej listy.

Okno Code jest oknem edytora tekstu, w którym można wstawić wykonywalny kod programu. Kod ten to zbiór dostępnych rozkazów, które zostaną wykonane, gdy nastąpi określone zdarzenie, które spowoduje wykonanie odpowiedniej procedury i zawartej w niej kodu.

Okno Project – okno eksploratora projektu.

Zawartość projektu:

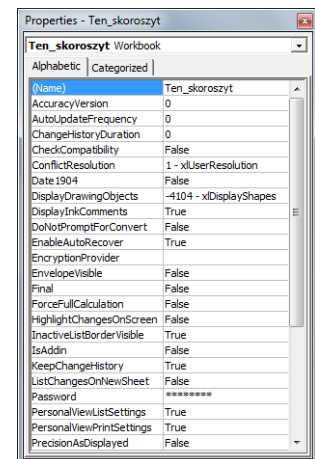
- Obiekty Excela
 - This Workbook – skoroszyt projektu
 - Arkusze
 - Wykresy
- Formularze – własne okna dialogowe
- Moduły – moduły kodu (standardowe)
- Moduły klas – moduły klas wykorzystywane przy tworzeniu własnych obiektów
- Odwołania do innych projektów



Okno Properties

Obiekty – poszczególne arkusze, wykresy, moduły, moduły klas oraz *This Workbook*

- Zbiory właściwości różnych obiektów są różne
- Nie wszystkie właściwości zawsze można zmieniać
- Nie wszystkie właściwości są zawsze widoczne



Okno Code

Możliwości wyświetlania:

- Wszystkie procedury z modułu oddzielone poziomymi liniami
- Tylko jedna aktualnie wybrana procedura
- Okno podzielone na dwie niezależnie przewijane części
- Zawartości listy obiektów i listy procedur zależą od obiektu aktualnie wybranego do edycji

```

dodatek_expose_lab1.xlam - dodatek_exposelab (Code)
[General] | [Declarations]
Set stary_aktyny_arkusz = ActiveWorkbook.ActiveSheet
Set ActRan = Selection
Set najmn = Sheets(1)
For i = 1 To Sheets.Count
    For j = 1 To Sheets.Count
        If Sheets(j).Name < najmn.Name Then
            Set najmn = Sheets(j)
        End If
        najmn.Move before:=Sheets(1)
    Next
    On Error GoTo blad
    Set najmn = Sheets(i + 1)
Next
blad: stary_aktyny_arkusz.Activate
stary_aktyny_arkusz.Activate
ActRan.Select
End Sub

Sub Macro3(control As IRibbonControl)

```

Edycja kodu

- Po przejściu do nowej linii automatycznie jest sprawdzana poprawność i dokonywane formatowanie (zamiana wielkości liter, stosowanie wcięć i kolorowanie składni)
- Znak podkreślenia '_' umożliwia kontynuowanie instrukcji w nowej linii, ale nie można dzielić w ten sposób ciągu znaków w cudzysłowach

Okno Immediate

Okno instrukcji bezpośrednich służy do wykonywania pojedynczych poleceń VBA. Wypisywanie tekstu następuje instrukcją Debug.Print (lub krócej '?'), Umożliwia sprawdzanie wyniku wykonania instrukcji

Okno Immediate jest bardzo przydatne do testowania poleceń i wyrażeń języka VBA. Na przykład po wprowadzeniu wyrażenia:

```

Immediate
Activecell.Interior.Color = RGB(250,0,0)
Debug.print Activecell.Text
aaaa
? 7/12
0,583333333333333
? now()
2013-02-14 15:26:01
? date()
2013-02-14

```

? Range("A1").Value

i naciśnięciu Enter w następnym wierszu okna Immediate zostanie wyświetlony wynik.

• Znaczenie kolorów w edytorze

```

vba.xlsm - Rejestrator_makr (Code)
[General]
Option Explicit

Sub WklejWartosci()
'
' WklejWartosci Makro
' Klawisz skrót: Ctrl+Shift+Q
'|
    Selection.Copy
    Selection.PasteSpecial Paste:=xlPasteValues
    Application.CutCopyMode = False
End Sub

```

Na zdjęciu powyżej jest dokładnie skopiowany kod programu, łącznie z kolorami, które też mają swoje znaczenie.

kolor niebieski – słowa kluczowe VBA

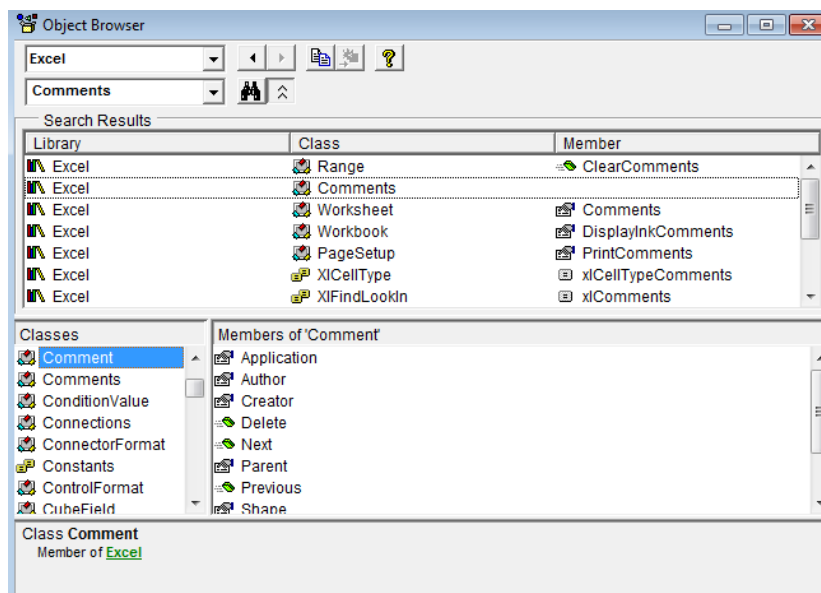
kolor zielony – wszystkie dodatkowe informacje, opisy – jednym słowem tekst własny programisty, w którym możemy wpisywać wszystkie uwagi. Zaczynają się one od górnego apostrofu ', kończą naciśnięciem Enter, czyli przejściem do następnej linii.

• Przeglądarka obiektów

Przeglądarka Obiektów, czyli Object Browser jest bardzo przydatnym narzędziem wyświetlającym wszystkie właściwości i metody dostępne dla danego obiektu. Po uaktywnieniu edytora Visual Basic przeglądarkę można uruchomić na jeden z trzech sposobów:

- Przyciskiem F2
- Z menu View wybrać polecenie Object Browser
- Klikając przycisk Object Browser znajdujący się na pasku narzędzi Standard

Okno przeglądarki Object Browser zostało przedstawione poniżej.



Lista rozwijalna w lewym górnym narożniku Object Browser zawiera wszystkie dostępne biblioteki obiektów:

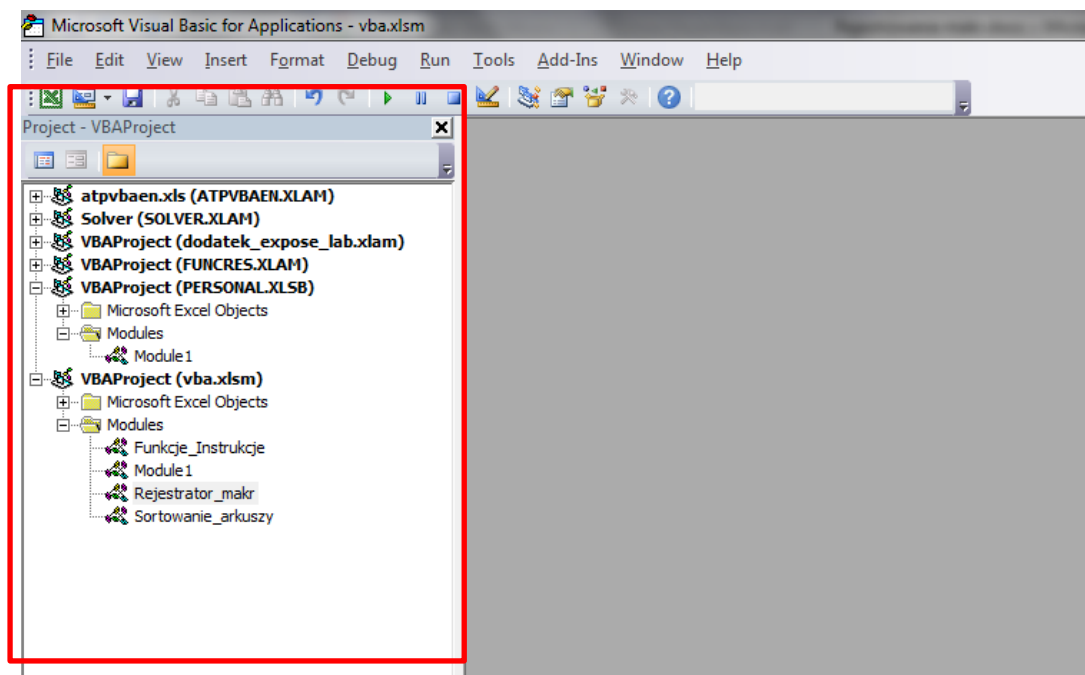
- Program Excel
- MSForms (używana do tworzenia niestandardowych okien dialogowych)
- Office (obiekty wspólne dla wszystkich aplikacji pakietu Microsoft Office)
- Stdole (obiekty automatyzacji OLE)
- VBA
- Bieżący projekt i wszystkie skoroszyty, do których się odwołuje.

Zawartość okna Classes zależy od elementu wybranego z listy. Z kolei zawartość okna Members Of zależy od elementu, który zostanie zaznaczony w oknie Clases.

Jeśli po wybraniu biblioteki chce się uzyskać listę właściwości i metod zawierających określony tekst, można go wprowadzić w polu szukania Search Text.

• Moduły

Ważnym elementem edytora VBA jest część Project, oznaczona na rysunku poniżej



Jeżeli nie widzisz w swoim edytorze ramki Project, kliknij kombinację klawiszy Ctrl+R.

W tej sekcji edytora przedstawione są wszystkie komponenty otwartego projektu VBA.

Każdy projekt VBA może składać się z następujących części:

- Microsoft Excel Object (obiekty Excela). Są to wszystkie elementy związane z normalną obsługą Excela, czyli arkusze i wykresy.
 - Czasem może się zdarzyć, że w edytorze VBA widocznych będzie więcej arkuszy niż w normalnym widoku Excela – wynika to z faktu, że niektóre arkusze mogą być ukryte i nie są widoczne w normalnym widoku, natomiast w edytorze VBA pokazane są wszystkie arkusze bez względu na to, czy są ukryte czy nie.
- Forms (formularze). Graficzne interfejsy użytkownika.
- Modules (moduły). Moduły to podstawowe i najbardziej uniwersalne obiekty do przechowywania kodu VBA.
- Class Modules (moduły klas). Najbardziej skomplikowany obiekt edytora VBA.

1.1 Pętle

Konstrukcja	Składnia, przykład	Opis
For ... Next	Składnia: For licznik = start To koniec [Step krok] [instrukcje] [Exit For] [instrukcje] Next [licznik]	Powtarza instrukcję lub blok instrukcji określoną liczbę razy, gdzie: licznik – zmienna używana jako licznik pętli, start i koniec – wartość początkowa i końcowa licznika, opcjonalnie argument słowa kluczowego Step określa wartość zmian licznika dla każdego wykonania bloku instrukcji (wartość domyślna wynosi 1). Pętla może zawierać instrukcje Exit For umieszczone w dowolnym miejscu pomiędzy For ... Next jako alternatywne wyjście z bloku pętli. Exit For przekazuje sterowanie do instrukcji bezpośrednio następującej po Next .
Do ... Loop	Wariant I Do [{While Until} warunek] [instrukcje] [Exit Do] [instrukcje] Loop	Powtarza instrukcję lub blok instrukcji, dopóki warunek While ma wartość True (warunek jest spełniony) lub warunek Until nie jest spełniony (ma wartość False).
	Wariant II Do [instrukcje] [Exit Do] [instrukcje] Loop [{While Until} warunek]	Warunek sprawdzany jest na końcu pętli (pętla zostanie wykonana przynajmniej jeden raz).
	Pętla może zawierać instrukcję Exit Do umieszczone w dowolnym miejscu pomiędzy Do ... Loop jako alternatywne wyjście z bloku pętli. Exit Do przekazuje sterowanie do instrukcji bezpośrednio następującej po Loop . W przypadku zagnieżdżonych pętli Do ... Loop instrukcja Exit Do przekazuje sterowanie do pętli o jeden poziom powyżej pętli, w której nastąpiło przerwanie.	
While ... Wend	While warunek [instrukcje] Wend	Wykonuje instrukcję lub blok instrukcji, dopóki warunek będzie miał wartość True . Pętla może być zagnieżdżona.

Można umieszczać pętlę w obrębie drugiej pętli. Wewnętrzna pętla zostanie wykonana w całości (wszystkie instrukcje zawarte w pętli) w każdym cyklu pętli zewnętrznej. Można dowolnie zagnieżdżać pętle bez względu na ich rodzaj.

Nieskończone pętle można przerwać kombinacją klawiszy – **ctrl+break**.

Przykłady:

- **Instrukcja warunkowa IF i jej warianty**

```
Sub GreetMe1()
    If Time > 0.5 Then MsgBox "Witam przed południem!"
End Sub
```

```
Sub GreetMe2()
    If Time < 0.5 Then MsgBox "Witam przed południem!"
    If Time >= 0.5 Then MsgBox "Witam przed południem!"
End Sub
```

```
Sub GreetMe3()
    If Time < 0.5 Then
        MsgBox "Witam przed południem!"
        ' pozostałe polecenia
    Else
        MsgBox "Witam! Już po południu!"
        ' pozostałe polecenia
    End If
End Sub
```

```
Sub GreetMe4()
    If Time < 0.5 Then MsgBox "Witam przed południem"
    If Time >= 0.5 And Time < 0.75 Then MsgBox "Witaj! Już po
    południu!"
    If Time >= 0.75 Then MsgBox "Dobry wieczór!"
End Sub
```

```
Sub GreetMe5()
    If Time < 0.5 Then
        MsgBox "Witam przed południem!"
    ElseIf Time >= 0.5 And Time < 0.75 Then
        MsgBox "Witaj! Już po południu!"
    Else
        MsgBox "Dobry wieczór!"
    End If
End Sub
```

```
Sub Discount1()  
  
Dim Quantity As Variant  
Dim Discount As Double  
  
Quantity = InputBox("Wprowadź liczbę kupowanych książek: ")  
If Quantity = "" Then Exit Sub  
  
    If Quantity >= 0 Then Discount = 0.1  
    If Quantity >= 25 Then Discount = 0.15  
    If Quantity >= 50 Then Discount = 0.2  
    If Quantity >= 75 Then Discount = 0.25  
  
MsgBox "Rabat: " & Discount  
End Sub
```

- **Pętle FOR**

Suma pierwiastków kwadratowych pierwszych 100 liczb całkowitych

```
Sub SumPierw()

Dim Sum As Double
Dim Count As Integer

Sum = 0
  For Count = 1 To 100
    Sum = Sum + Sqr(Count)
  Next Count
  MsgBox "Suma kwadratów 100 liczb całkowitych wynosi: " & Sum
End Sub
```

Usuwa z aktywnego skoroszytu 2, 4, 6, 8, oraz 10 wiersz

```
Sub DelRows()

Dim RowNum As Long
  For RowNum = 10 To 2 Step -2
    Rows(RowNum).Delete
  Next RowNum
End Sub
```

Wypełnianie od 1 do 100 od aktywnej komórki

```
Sub GoodLoop()

Dim Start As Integer
Dim NumToFill As Integer
Dim cnt As Integer

Start = 1
NumToFill = 100
  For cnt = 0 To NumToFill - 1
    ActiveCell.Offset(cnt, 0).Value = Start + cnt
  Next cnt
End Sub
```

• Pętla WHILE (Do While)

Wypełnia komórki datami z obecnego miesiąca

```
Sub EnterDate1()  
Dim TheDate As Date  
  
TheDate = DateSerial(Year(Date), Month(Date), 1)  
Do While Month(TheDate) = Month(Date)  
    ActiveCell = TheDate  
    TheDate = TheDate + 1  
    ActiveCell.Offset(1, 0).Activate  
Loop  
End Sub
```

Otwiera plik tekstowy, odczytuje kolejne wiersze, zamienia małe litery na wielkie i zapisuje je w aktywnym arkuszy, poczynwszy od komórki A1 w dół kolumny

```
Sub DoWhileDemo1()  
Dim LineCt As Long  
Dim LineOfText As String  
  
Open "c:\data\plik_tekstowy.txt" For Input As #1  
LineCt = 0  
Do While Not EOF(1)  
    Line Input #1, LineOfText  
    Range("A1").Offset(LineCt, 0) = UCase(LineOfText)  
    LineCt = LineCt + 1  
Loop  
Close #1  
End Sub
```

• Instrukcja SELECT CASE

```
Sub GreetMe6()  
Dim Msg As String  
  
Select Case Time  
    Case Is < 0.5  
        Msg = "Witam przed południem!"  
    Case 0.5 To 0.75  
        Msg = "Witaj! Już po południu!"  
    Case Else  
        Msg = "Dobry wieczór!"  
End Select  
MsgBox Msg  
End Sub
```

```
Sub Discount2()  
  
Dim Quantity As Variant  
Dim Discount As Double  
  
Quantity = InputBox("Wprowadź liczbę kupowanych książek: ")  
  
Select Case Quantity  
    Case ""  
        Exit Sub  
    Case 0 To 24  
        Discount = 0.1  
    Case 24 To 49  
        Discount = 0.15  
    Case 50 To 74  
        Discount = 0.2  
    Case Is >= 75  
        Discount = 0.25  
End Select  
  
MsgBox "Rabat: " & Discount  
End Sub
```

Czy dzisiejszy dzień to sobota lub niedziela?

```
Sub Weekend()  
  
Select Case Weekday(Now)  
    Case 1, 7  
        MsgBox "Mamy już weekend!"  
    Case Else  
        MsgBox "To jeszcze nie weekend!"  
End Select  
  
End Sub
```



ul. Skierniewicka 10a
01-230 Warszawa
Tel.: 22 465 88 88
biuro@expose.pl
www.expose.pl