

**Aplikacje w środowisku VBA**

**Visual Basic for Applications**

# Podstawowe informacje o VBA

- **Visual Basic for Applications**, w skrócie VBA, to język programowania rozwijany przez Microsoft, którego zastosowanie pozwala manipulować obiektami w takich produktach jak; **Excel, Word, PowerPoint**, którymi w przypadku Excela są; **wykresy, tabele przestawne, scenariusze, oraz spora liczba funkcji matematycznych, finansowych, inżynierskich i biznesowych.**

# Podstawowe informacje o VBA

- Sterowanie obiektami staje się możliwe poprzez zastosowanie i tworzenie makropoleczeń. Aby ta czynność stała się możliwa, obowiązkowo należy;
- **poznać ogólne zasad wykorzystania makropoleczeń,**
- **opanować umiejętność rejestrowania makropoleczeń i ich testowania oraz rejestrowanie makropoleczeń, które wywołują inne makropoleczenia.**

# Zakres zastosowań VBA 1

- wprowadzania bloków tekstów.
- automatyzacja często wykonywanego zadania,
- automatyzacja powtarzalnych operacji,
- obsługi dużej ilości danych,
- tworzenie własnego polecenia,
- tworzenie własnego przycisku,

# Zakres zastosowań VBA 2

- zabezpieczenie danych przed błędami,
- tworzenie własnych funkcji arkusza kalkulacyjnego,
- tworzenie własnych dodatków do Excela,
- tworzenie kompletnych aplikacji opartych na makrach.

# Zakres zastosowań VBA 3

- Narzędziem wymagającym głębszej wiedzy, jest korzystanie z języka programowania Visual Basic for Applications. Do posługiwania się nim konieczne jest:
- poznanie kilku pojęć,
- zdobycie umiejętności obsługi edytora VBA,
- poznanie zasad programowania w VBA.

# Stosowane pojęcia 1

- **makro** - akcja lub zestaw akcji, którego można użyć do automatyzacji zadań. Makra są rejestrowane w języku programowania Visual Basic for Applications,
- **kod programu** - wykonanie kodu napisanego w języku VBA pozwala zrealizować żądane operacje.
- **moduł** - moduły VBA są przechowywane w skoroszybie Excela, ale do ich przeglądania lub modyfikacji musisz użyć edytora VBE (Visual Basic Editor),

# Stosowane pojęcia 2

- **procedury** - to uproszczona jednostka kodu programu realizująca określone zadanie.
- **obiekty** - język VBA operuje na obiektach powiązanej z nim aplikacji (np. Excel). Excel oferuje ponad 100 klas obiektów, na których można wykonywać operacje.



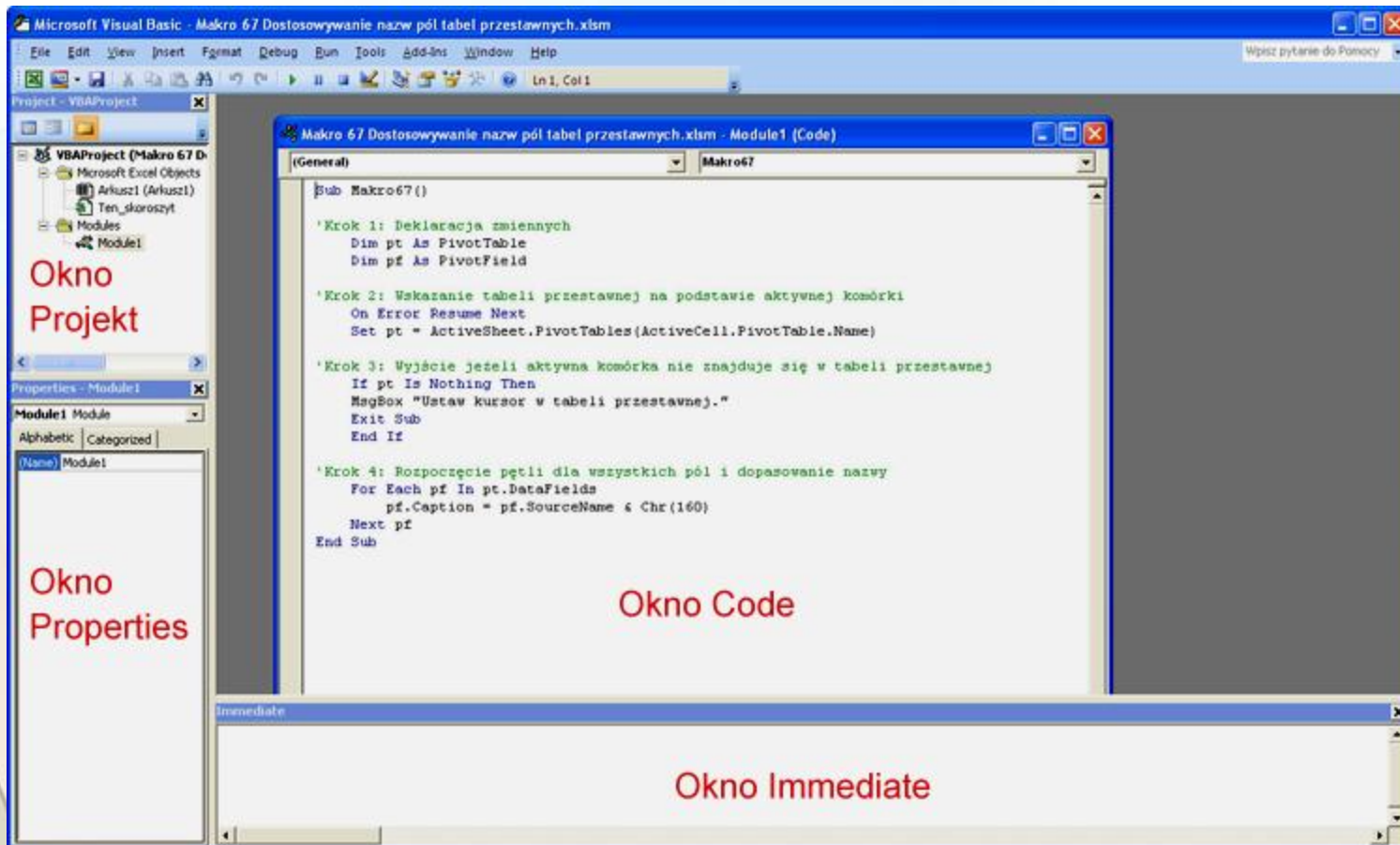
# Edytor VBA 1

- Edytor VBE składa się z kilku elementów, a najważniejszymi są;
- **pasek menu edytora VBE** - zawiera polecenia obsługujące różnorodne składniki VBE.
- **pasek narzędzi** - umieszczony jest domyślnie pod paskiem menu.

# Edytor VBA 2

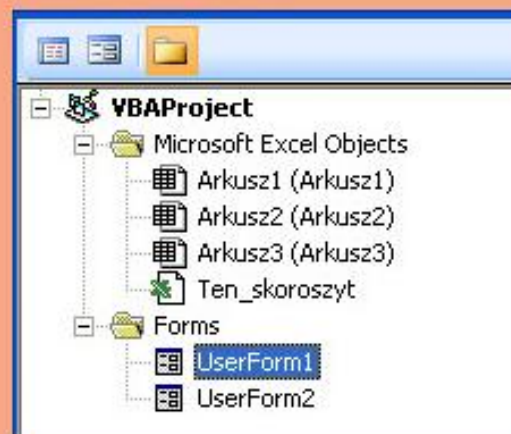
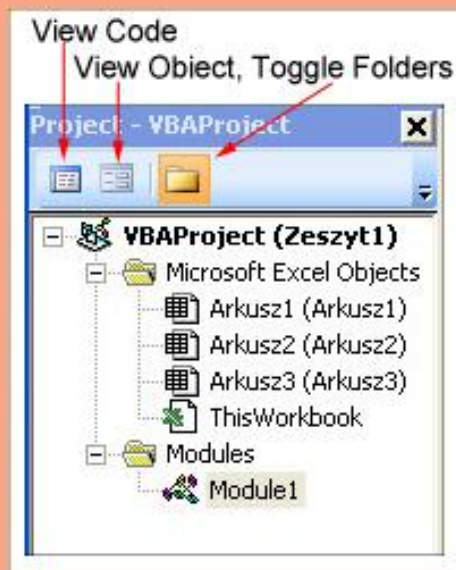
- **Okno Project** - w oknie wyświetlana jest struktura zawierająca wszystkie aktualne otwarte skoroszyty Excela (również te ukryte oraz dodatki) i pokazuje hierarchiczny spis wszystkich elementów projektu.
- **Okno Code** - w oknie jest wyświetlany kod VBA. Każdy obiekt w projekcie ma skojarzone ze sobą okno Code.

# Edytor VBA 3



# Edytor VBA - okno Projekt

**Okno Project** (Eksplorator projektów) wyświetla hierarchiczny spis wszystkich elementów projektu, tj. skoroszyty i służy do poruszania się pomiędzy elementami projektu. Na rysunku widoczne są trzy arkusze (Arkusz 1, Arkusz 2, Arkusz 3), a pod nimi skoroszyt (ThisWorkbook, Ten\_skoroszyt)

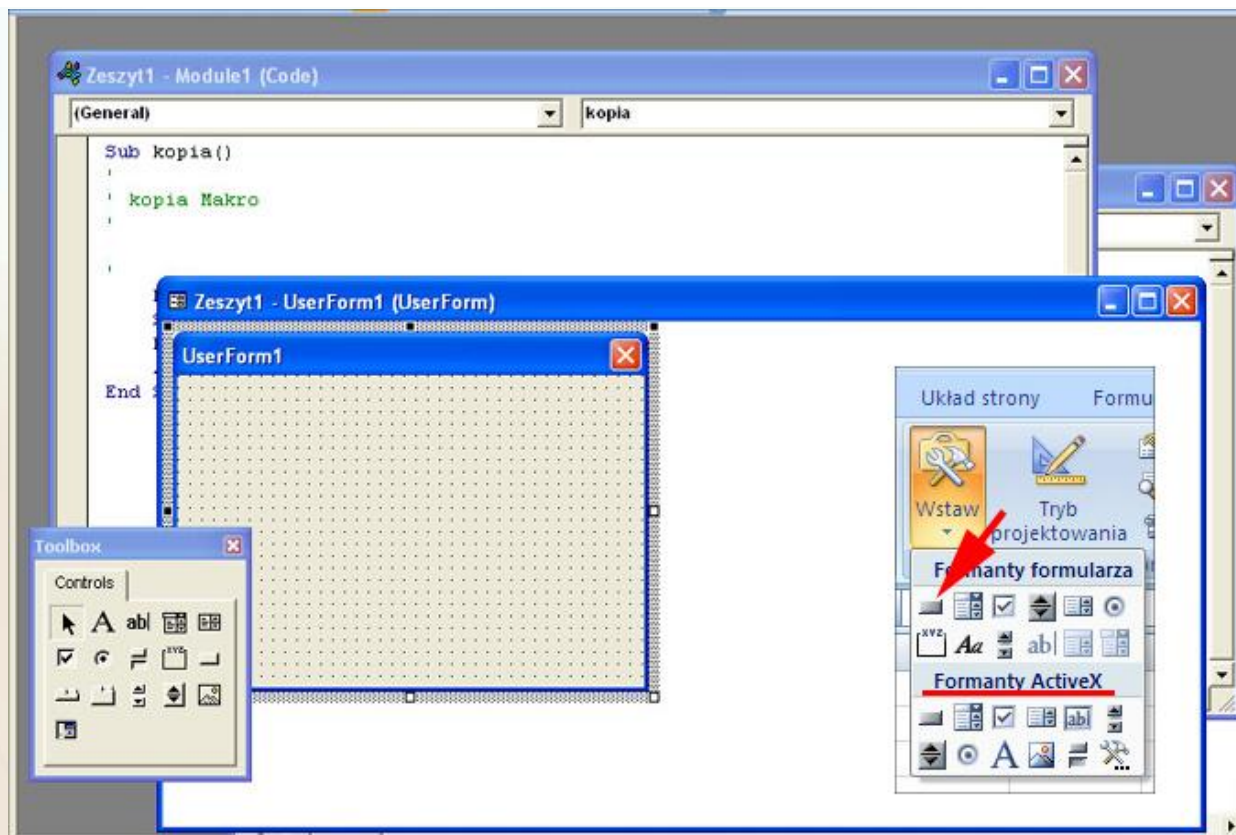


# Edytor VBA - okno Projekt

- W górnej części okna znajdują się przyciski opisane jako:
- **View Code** - otwiera moduł kodu zaznaczonego w oknie obiektu,
- **View Object** - możemy użyć tego przycisku aby otworzyć zaznaczony w oknie obiekt na przykład: formularz lub powrócić do arkusza Excela jeżeli jest on zaznaczony,
- **Toggle Folders** - ukrywa lub pokazuje foldery obiektów.

# Edytor VBA - obiekty UserForm

Projekt może również zawierać węzeł o nazwie **Forms (Formularze)**, przechowujący obiekty **UserForm**, tj. przyciski, obrazy i obszary tekstu. Formularze są kontenerami dla formantów.



# Edytor VBA - formanty

- Kontrolki (formanty) to elementy interfejsu, które programista wykorzystuje w swoim programie. Są to obiekty, których zdarzenie trzeba zaprogramować.
- **Program Microsoft Excel ma dwa typy formantów:**
- **Formanty formularza** są prostsze w użyciu, gdyż nie wymagają pisania kodu w języku Visual Basic for Applications (VBA)
- **Formanty ActiveX** wymagają umiejętności tworzenia makr w języku VBA, ale dzięki temu oferują znacznie więcej możliwości.

# Formanty ActiveX

The screenshot shows a Microsoft Excel spreadsheet with a UserForm1 dialog box overlaid. The spreadsheet has a column of months from January to December. The UserForm1 dialog box contains the following controls:

- CheckBox1
- CommandButton1
- Frame1 containing an image of green cacti.
- Label1
- Styczeń, Luty, Marzec, Kwiecień, Maj, Czerwiec, Lipiec (List Box)
- Page1 | Page2 (Page Navigation)
- RadioButton1
- RadioButton2
- Navigation buttons (left, right, double left, double right)
- Tab1 | Tab2 (Tab Navigation)
- ToggleButton1



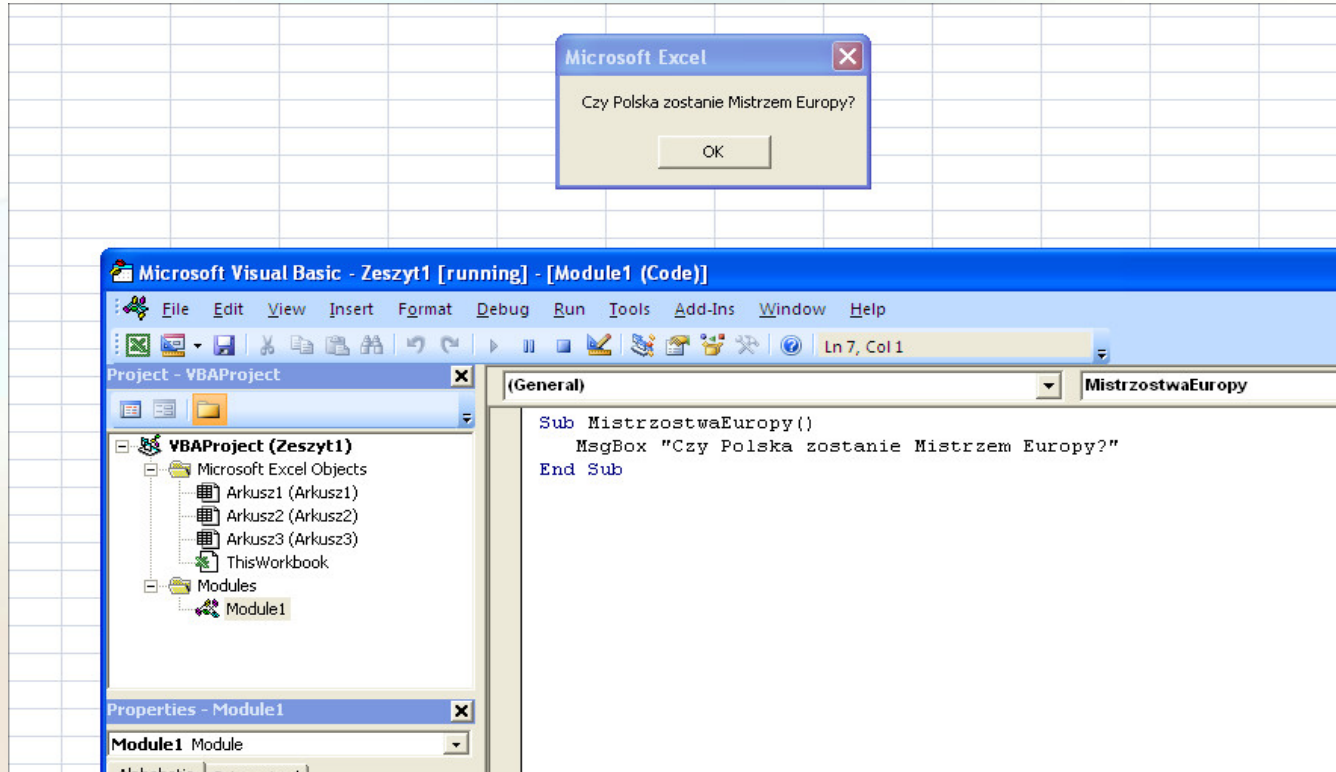
# Procedury, deklaracje

- W oknie kodu można przechowywać trzy rodzaje kodu języka VBA;
- **Procedury Sub** - procedura to zbiór poleceń, które wykonują określone operacje,
- **Procedury Function** - funkcja to zbiór instrukcji zwracających wartość lub tablicę.
- **Deklaracje** - deklaracje to informacja na temat zmiennej zastosowanej w kodzie źródłowym języka VBA.

# Instrukcje programu

- **Instrukcje programu zorganizowane są w procedury, moduły i projekty.**
- **Procedura** - składa się z przynajmniej jednej instrukcji umieszczonej między dwiema specjalnymi instrukcjami: z których pierwsza z nich deklaruje procedurę a ostatnia ją zamyka.
- **Moduł** - zawiera z jedną lub więcej procedur oraz sekcje deklaracji w której umieszczamy instrukcje wspólne dla w wszystkich procedur tego modułu.
- **Projekt** - obejmuje wszystkie moduły, formularze, obiekty aplikacji macierzystej dokumentu oraz sam dokument.

# Przykłady - procedura **Sub**



**Funkcja MsgBox** umożliwia wyświetlanie komunikatów na ekranie, w zależności od wybranych parametrów może wyświetlać różne przyciski, to co użytkownik wybierze ma wpływ na to jakie czynności wykona makro

# Przykłady - procedura **Function**

The image shows a screenshot of Microsoft Excel and the Visual Basic Editor (VBE) illustrating a VBA function call. In the Excel spreadsheet, cell A1 contains the number 5. A red arrow points from this cell to the VBA code editor. The VBE window shows the following code in the (General) tab of the potegowanie module:

```
Function potegowanie(ByVal liczba1 As Double) As Double
    potegowanie = liczba1 * liczba1
End Function
Sub komunikat ()
    l1 = Range ("A1")
    wynik_potegowania = potegowanie(l1)
    MsgBox "Potęga podanej liczby wynosi " & wynik_potegowania
End Sub
```

Below the code editor, the Properties window for Module1 is visible. A red arrow also points from the VBA code editor to a message box that appears on the Excel spreadsheet. The message box contains the text: "Potęga podanej liczby wynosi 25".

# Deklarowanie zmiennych, DIM

The image shows a screenshot of Microsoft Excel with the Visual Basic editor open. The Excel window displays a grid with columns A through M and rows 1 through 26. A message box titled "Microsoft Excel" is open, showing the number "6" and an "OK" button. A red arrow points from the "6" in the message box to the line "Liczba = 36" in the VBA code window. The VBA editor window is titled "Microsoft Visual Basic - Zeszyt1 [running] - [Module1 (Code)]" and shows the following code:

```
Sub ShowRoot()  
    Dim Liczba As Double  
    Dim SquareRoot As Double  
    Liczba = 36  
    SquareRoot = Sqr(Liczba)  
    MsgBox SquareRoot  
End Sub
```

The VBA editor also shows a Project Explorer window with the following structure:

- VBAProject (Zeszyt1)
  - Microsoft Excel Objects
    - Arkusz1 (Arkusz1)
    - Arkusz2 (Arkusz2)
    - Arkusz3 (Arkusz3)
    - ThisWorkbook
  - Modules
    - Module1

The Properties window for Module1 is also visible, showing the name "Module1".

# Deklaracje zmiennych

- Deklaracje zmiennych są bardzo proste i rozpoczynają się od słowa kluczowego Dim
- Dim MojaNazwa As String
- Dim NumerWiersza As Long
- Dim Licznik As Integer
- Dim PracaZakonczona As Boolean
- **Dim**, pochodzi od angielskiego słowa dimension, czyli wymiar. Używając tego słowa nadajemy konkretny rozmiar zmiennej