

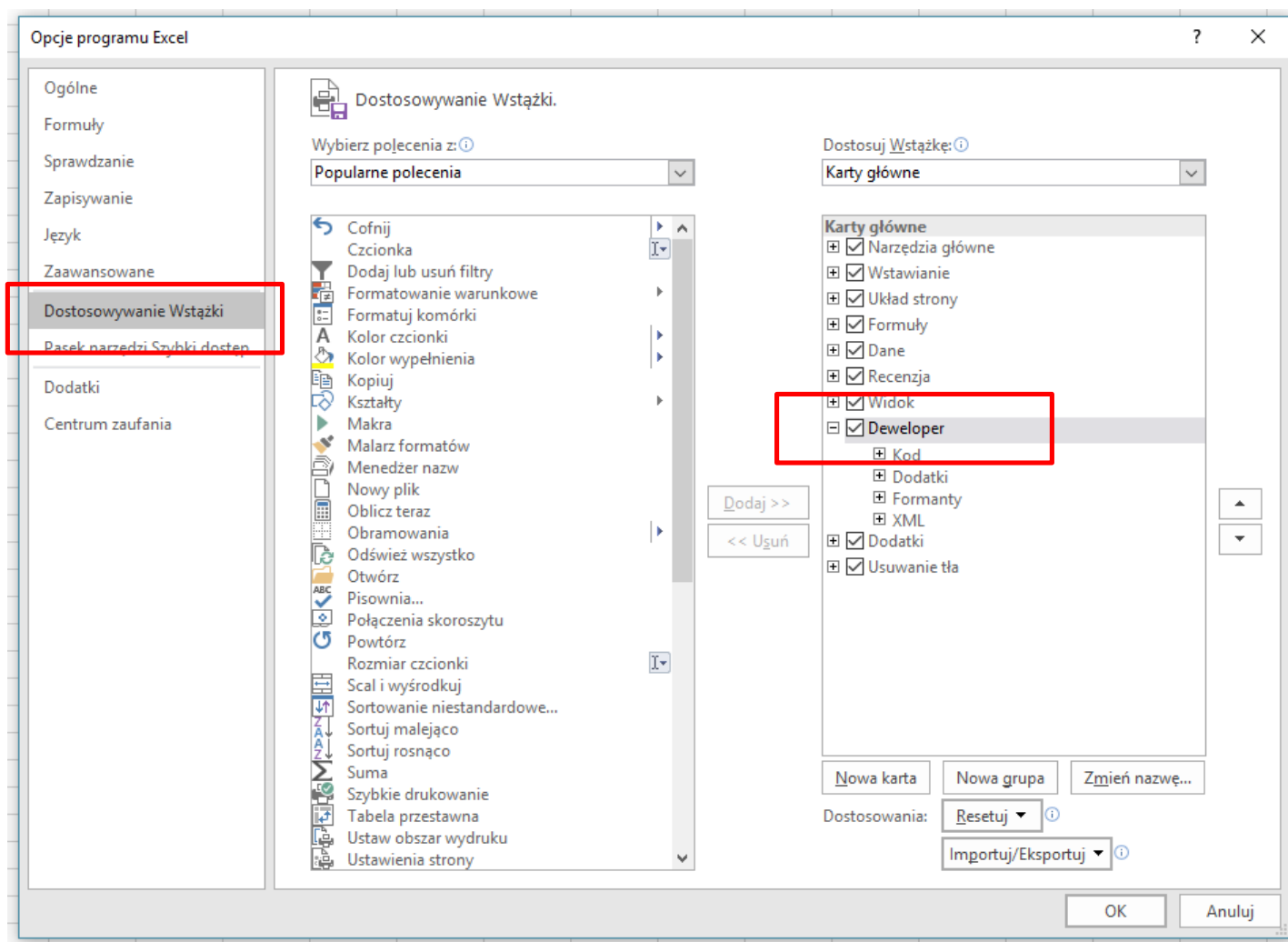
Technologie informacyjne – lab_VBA

01.10.2019

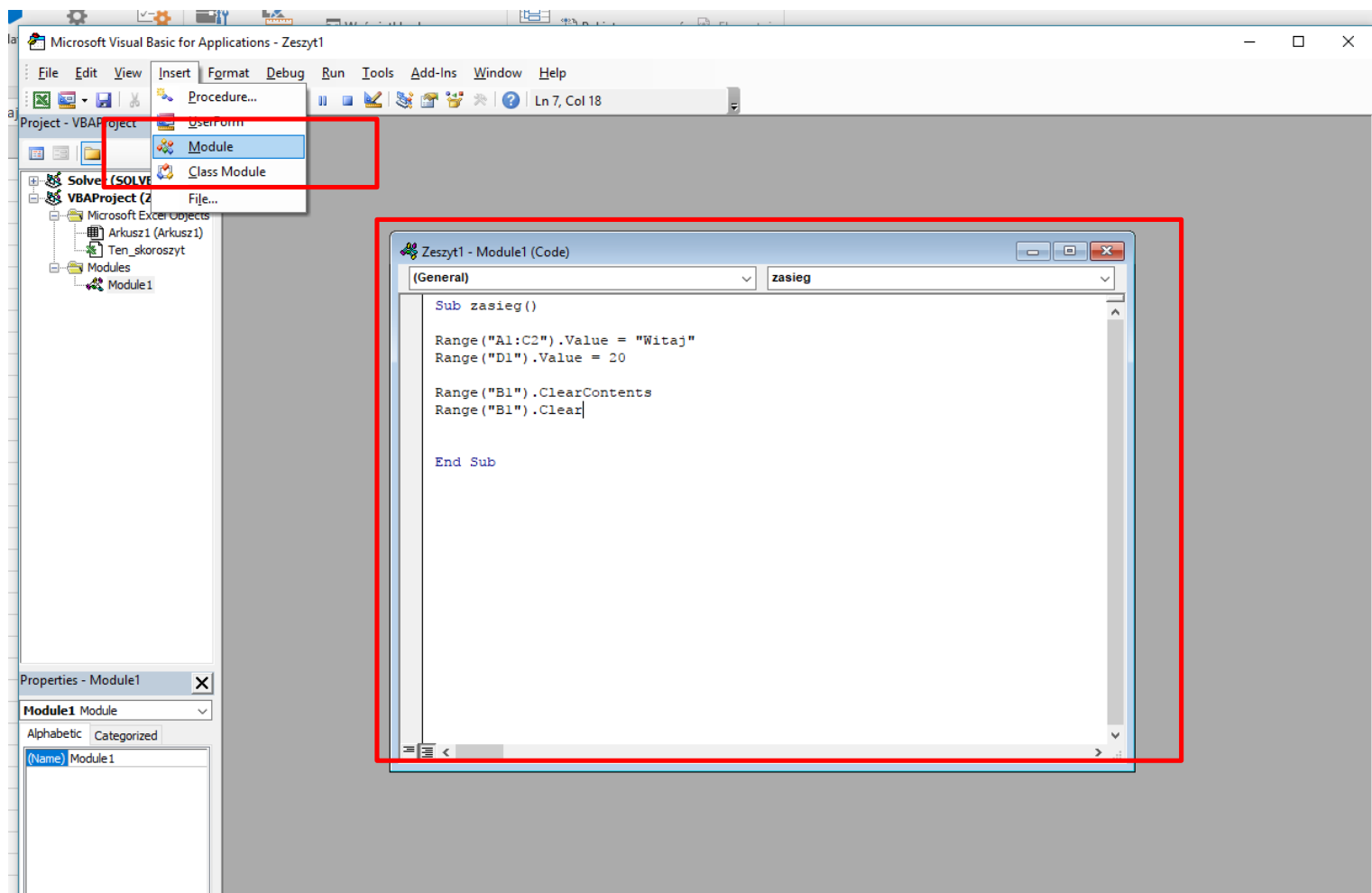
Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie
AGH University of Science and Technology



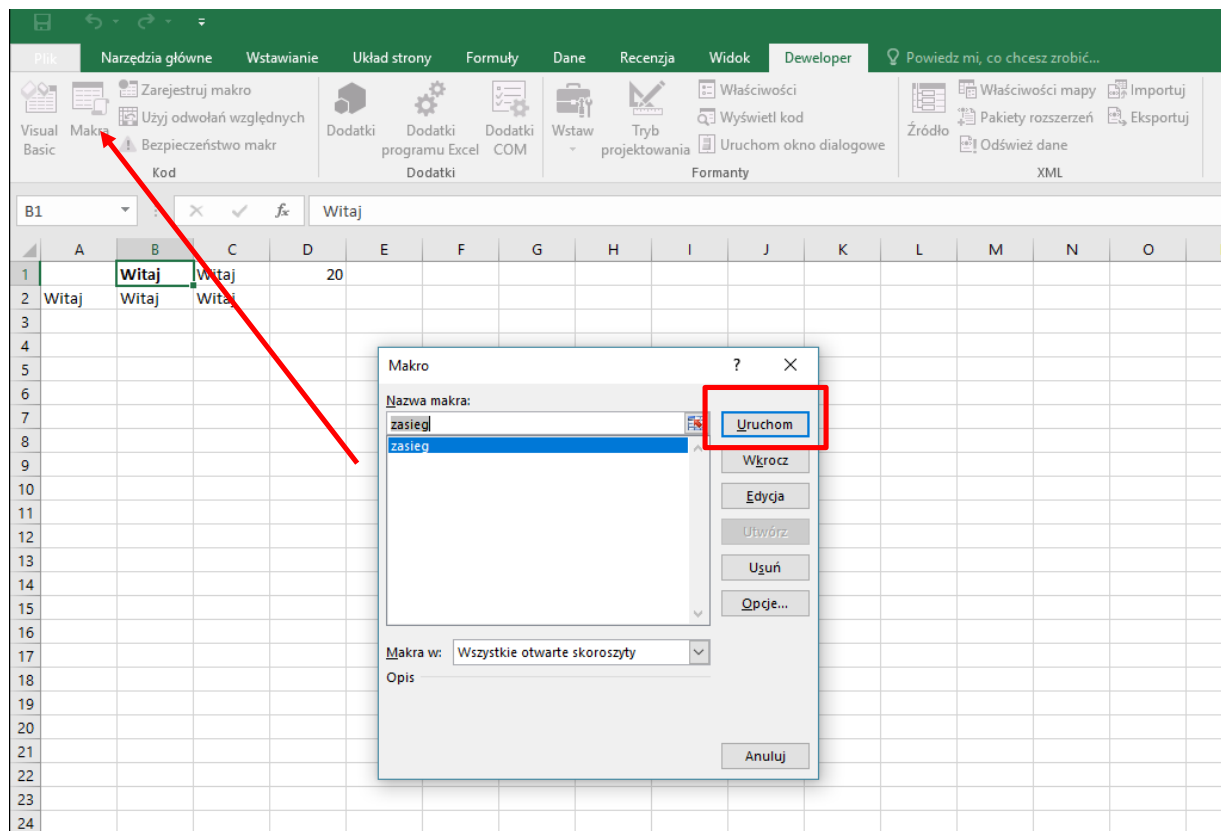
Deweloper



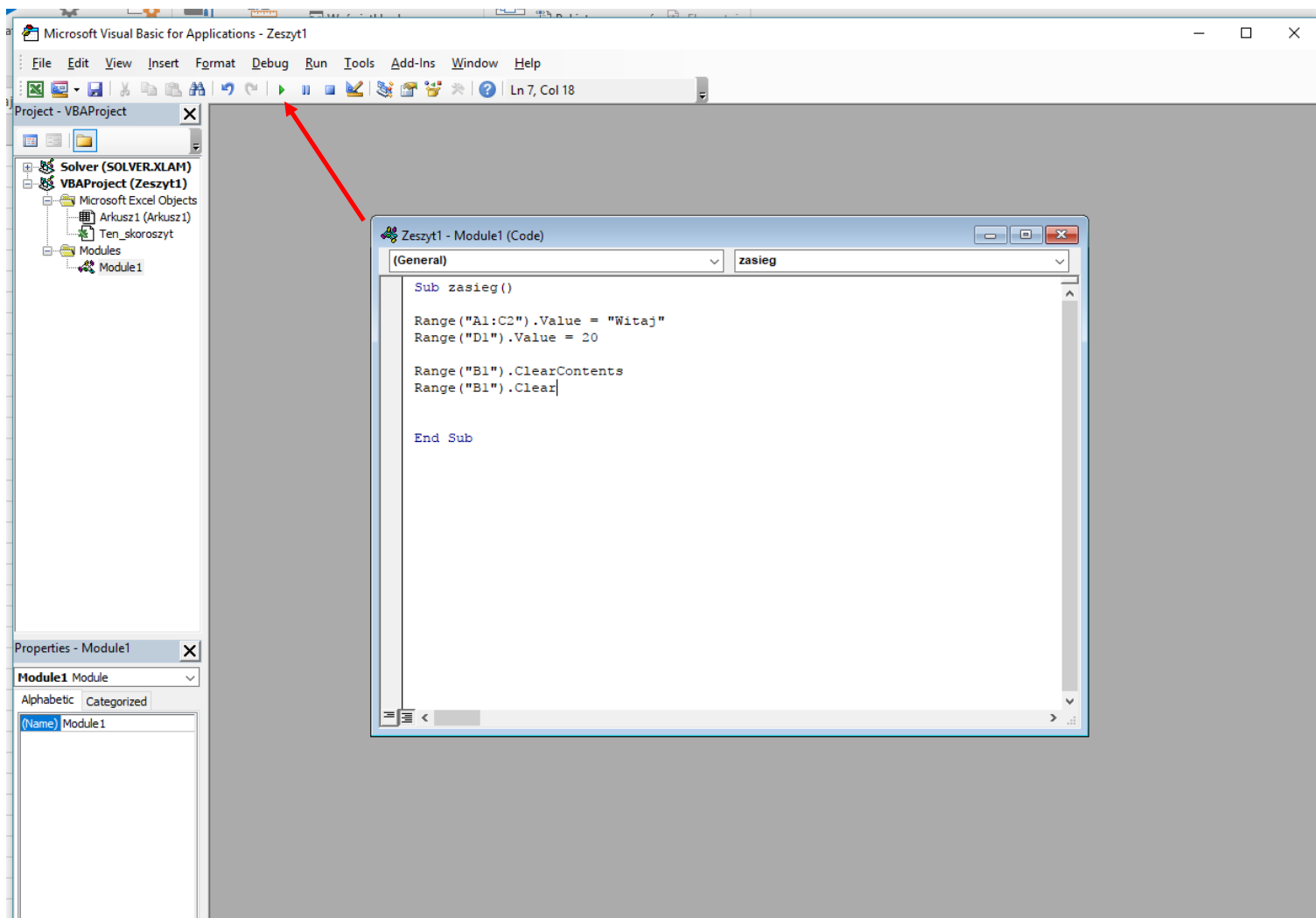
Deweloper



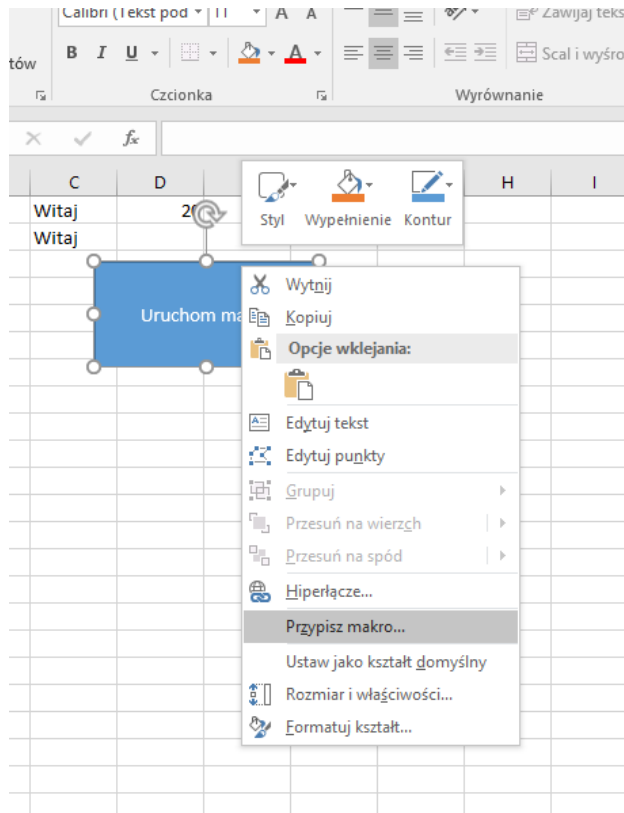
Makro - wywołanie



Makro - wywołanie



Makro - wywołanie



Istnieje możliwość tworzenia własnych przycisków uruchamiających nasze makra

VBA - zmienne

■ Zmienne lokalne i globalne

Typ danych	Liczba używanych bajtów	Zakres wartości
Boolean	2	wartość logiczna prawda lub fałsz wartości: True, False wartość domyślna False
Integer	2	liczba całkowita od -32 768 do 32 767 wartość domyślna 0
Long	4	liczba całkowita od -2 147 483 648 do 2 147 483 647 wartość domyślna 0
Single	4	liczba rzeczywista od -3,402823E38 do -1,401298E-45 i od 1,401298E-45 do 3,402823E38 wartość domyślna 0
Double	8	liczba rzeczywista dwukrotnie większa od Single od -1,79769313486232e+308 do -4,9406564581247e-324 i od 4,9406564581247e-324 do 1,79769313486232e+308 wartość domyślna 0
String	1 + 1 na każdy znak	tekst maksymalnie do 32 767 znaków wartość domyślna: "" - pusty
Date	8	data i czas w przedziale od 1 stycznia 100 r. do 31 grudnia 9999 roku wartość domyślna 0
Currency	8	typ walutowy, liczby rzeczywiste od -922 337 203 658 477,5808 do 922 337 203 685 477,5807 wartość domyślna 0
Variant	1 do 8	dowolny typ identyfikowany przez Visual Basic w zależności od pierwszego użycia tej zmiennej wartość domyślna: EMPTY (puste) domyślny typ dla nie zadeklarowanych zmiennych
Zdefiniowany przez użytkownika	zależy od definicji	zależny od definicji

```
Sub makro1()
Dim zmienna1 As Integer
[kod makra]
End Sub
```

```
Sub makro2()
Public zmienna2 As Integer
[kod makra]
End Sub
```

VBA - stałe

- *Option Explicit On | Off*

```
Sub makro3()  
Const stala1 as Integer = 5           'deklaracja stałej typu integer  
Const stala2 = 777                   'deklaracja stałej bez podawania typu danych  
Public Const stala3 as String = "Jan Kowalski" 'deklaracja stałej publicznej  
  
[kod makra]  
  
End Sub
```


Objects, Properties and Methods

- Programowanie obiektowe
- Properties – to co obiekt posiada
- Methods – to co można zrobić z obiektem

HIERARCHIA:

Object.Property.Method



Some Properties don't have details

Range("A1").Address

Range("A1").Value



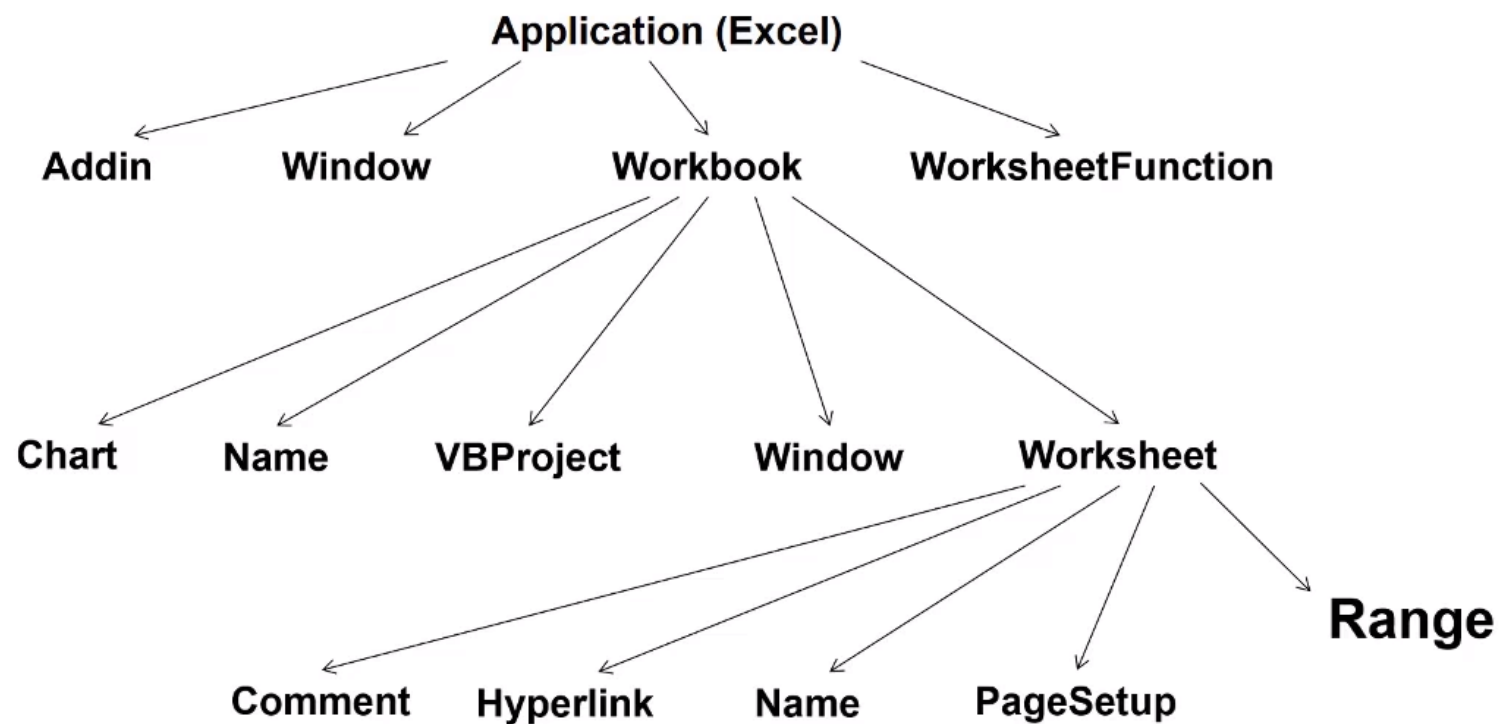
Some Properties return an object

Range("A2").Interior.Color

Range("A2").Font.Color

Objects

Object Hierarchy in VBA



Objects

Object Collections

Workbooks <u>s</u>	all currently-open workbook objects
Charts <u>s</u>	all charts contained in a particular workbook
Worksheets <u>s</u>	all sheets contained in a particular workbook

Referring to objects in a collection

Worksheets("Main")

Charts("XY")

Complete object references

Application.Workbooks("Project6").Worksheets("Main").Range("B16")

Objects

Object Collections

Workbooks <u>s</u>	all currently-open workbook objects
Charts <u>s</u>	all charts contained in a particular workbook
Worksheets <u>s</u>	all sheets contained in a particular workbook

Referring to objects in a collection

Worksheets("Main")

Charts("XY")

Complete object references

Application.Workbooks("Project6").Worksheets("Main").Range("B16")

Can be simplified when unambiguous

Workbooks("Project6").Worksheets("Main").Range("B16")

Worksheets("Main").Range("B16")

Range("B16")

Properties and Methods

Properties in VBA

Property is what an object has

An Analogy



Examples of Property

1. Color
2. Size
3. Type
4. Engine
5. Age
6.



Examples of Property

1. Color
2. Size
3. Gender
4. Material
5. Season
6.

Properties come after the object hierarchy

Car.Color – might be too broad

Properties and Methods

Range("A10").Address → Read-only

Range("A1").Value → Read & Write

Range("A1").Interior.Color Range("A2").Font.Color → Read & Write



Examples

```
Range("A1").Value = ActiveCell.Address
```

```
Range("A1").Interior.Color = vbRed
```

```
Range("A1").Font.Color = vbBlue
```

Properties and Methods

Methods in VBA

Method is what an object does

An Analogy



Examples of Method

1. Start
2. Stop
3. Crash



Examples of Method

1. PutOn
2. TakeOff

Methods can have additional information

How to start the car? Quickly or slowly?

Car.Stop Quickly

Car.Stop StopStyle:=Quickly

Methods can change properties

The "Crash" method would change the size property of the car

Properties and Methods

Methods Can Have Arguments

Range("A2").Clear	→	No further arguments
Range("A2").Delete	→	1 optional argument (Shift)
Sheet1.Copy	→	2 optional arguments (Before, After) which are optional but exclusive



Examples

Range("A2").Delete xlShiftToLeft

Delete(**[Shift]**)

Range("A3").Copy Range("B3")

Copy(**[Destination]**)

Range("C3").PasteSpecial xlPasteValues

PasteSpecial(**[Paste As XlPasteType = xlPasteAll]**, **[Operation As XlPasteSpecialOperation = xlPasteSpecialOperationNone]**, **[SkipBlanks]**, **[Transpose]**)

Sheet1.Copy After:=Sheet3

Copy(**[Before]**, **[After]**)

Sheet1.Copy , Sheet2

Leila Gharani - www.XelPlus.com

Properties and Methods

Properties, Methods and Events

properties are *attributes* of objects

methods are *actions* to be taken on objects

events are happenings that objects *respond* to

Examples:

Range("A1").Width	→	property
Range("B2").Clear	→	method
Open	→	event (workbook is opened)

There are 1,000s of Objects, Properties, Methods, and Events, but there are only a few that you'll use all the time. And, when you need a new one, you can find it.

VBA - składnia

- Sub *nazwa()*

Treść procedury

- **End Sub**
- **MsgBox ()** – wyświetlanie komunikatu;
- **Range(„Zakres1:Zakres2”).Value = ...** - przypisanie wartości dla danego zakresu
- **Range(„Zakres1:Zakres2”).Clear** – wyczyszczenie całej zawartości zakresu (również z formatowaniem)
- **Range(„Zakres1:Zakres2”).ClearContents** – wyczyszczenie tylko zawartości (formatowanie pozostaje)

VBA - składnia

- **Range(„Zakres1:Zakres2”).Copy Range(„Zakres3”)** – Kopiowanie zakresu (wystarczy podać pierwszą komórkę destynacji)
- **Funkcja Range może być używana z innymi funkcjami np.: MsgBox Range(„A1”).Value** – wyświetlenie zawartości komórki A1

Praca z arkuszami

- Aby istniała możliwość pracy z danymi konkretnego arkusza należy go aktywować: **Worksheets**(„Nazwa” lub nr arkusza).**Activate**
- Inne polecenie aktywujące arkusz może wyglądać następująco: **Sheets**(„Nazwa” lub nr arkusza).**Select**
- Dodawanie arkusza: **Sheets.Add After := Worksheets („Nazwa”)**; można również wstawiać „Before”
- Usuwanie Arkusza: **Worksheets**(„Nazwa”).**Delete**
- Zmiana nazwy arkusza: **Worksheets („Nazwa”).Name = „Nowa_Nazwa”**

Praca z plikami

- Otwieranie pliku: `Workbooks.Open(„ścieżka”)`
- Zamykanie pliku: `Workbooks(„Nazwa”).Close`
- Zapisywanie pliku: `Workbooks(„Nazwa”).SaveAs Filename := „Nowa Nazwa”` – nowa nazwa bez rozszerzenia

Funkcja if

If Range („A1”) = ... **Then**

... polecenie

Elseif ... **Then**

Else

... polecenie

End If

Operatory porównania i logiczne

operator <	And, operator
	Not, operator
operator <=	Or, operator
operator >	Xor, operator
operator >=	AndAlso, operator
operator =	OrElse, operator
	IsFalse, operator
operator <>	IsTrue, operator

Funkcja for

For i = 1 To 20

... polecenie

Next i

Operatory porównania i logiczne

operator <	And, operator
	Not, operator
operator <=	Or, operator
operator >	Xor, operator
operator >=	AndAlso, operator
operator =	OrElse, operator
	IsFalse, operator
operator <>	IsTrue, operator

Importowanie pliku

Dim Plik as string

Plik = „C:\...\nazwa_liku.txt”

Open Plik For Input as #1

numer_wiersza = 0

Do Until EOF (1)

Line Input #1, LineFromFile

Slowo = Split(LineFromFile,“,“)

ActiveCell.Offset(numer_wiersza, 0).Value = Slowo (0)

Loop

Close #1

Każdy otwarty plik musi mieć w Excellu przypisany numer



Cells, Rnd

Cells (row_ind, col_ind)

Rnd ()

Tablice

'tablica jednowymiarowa zawierająca 12 komórek

Dim tablica(1 To 12) As String

'tablica dwuwymiarowa o rozmiarach 10x10

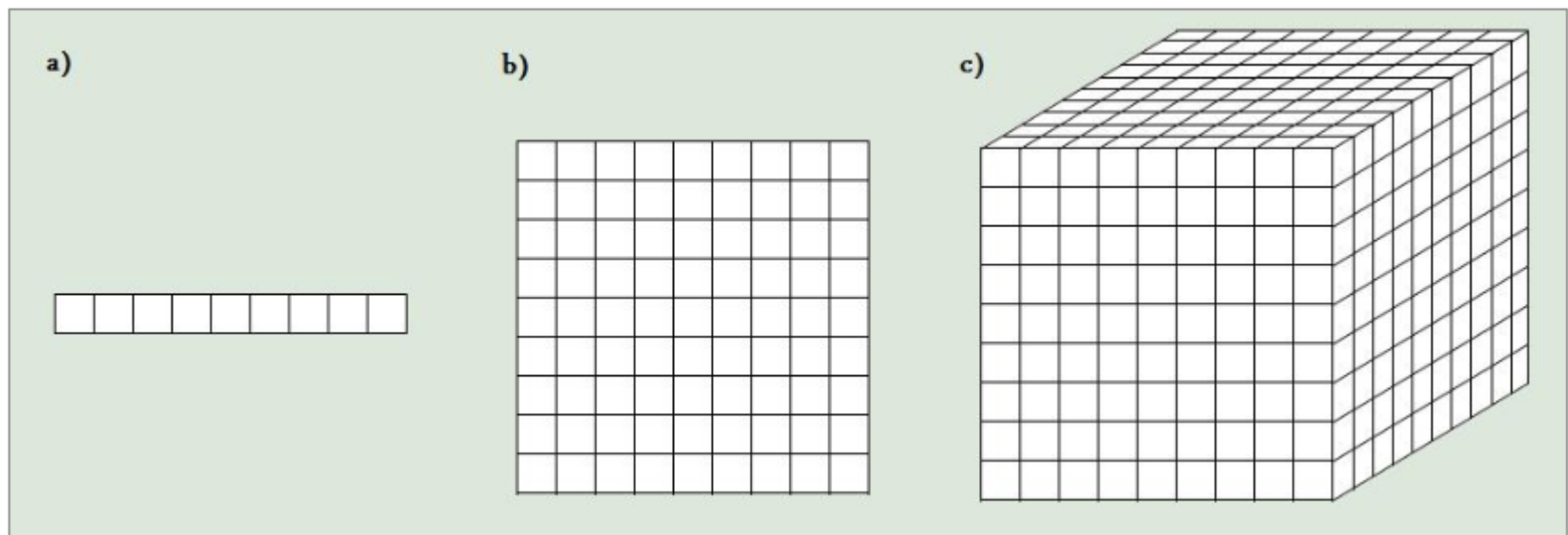
Dim tablica(1 To 10, 1 To 10) As Single

'tablica trójwymiarowa o rozmiarach 10x20x30

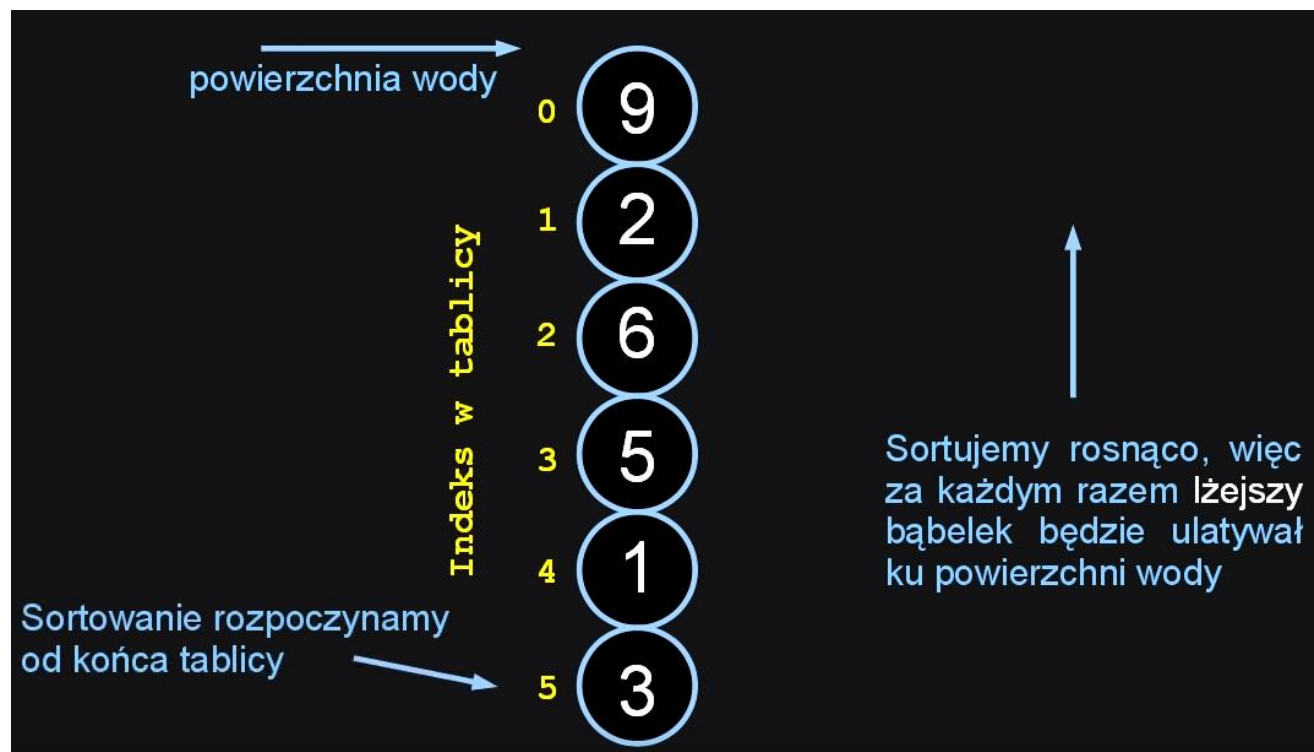
Dim tablica(1 To 10, 1 To 20, 1 To 30) As Double

'tablica sześciowymiarowa o rozmiarach 8x7x8x6x8x4

Dim tablica(1 To 8, 1 To 7, 1 To 8, 1 To 6, 1 To 8, 1 To 4) As Double



Sortowanie bąbelkowe



Sortowanie bąbelkowe

Diagram illustrating the Bubble Sort process. The table shows the state of an array of 6 numbers (9, 2, 6, 5, 1, 3) across 6 stages (0 to 5) of sorting. The numbers are represented in circles. The vertical axis is labeled "Indeks w tablicy" (Index in array) and the horizontal axis is labeled "Etapy sortowania" (Sorting stages).

	Etapy sortowania					
	0	1	2	3	4	5
0	9	1	1	1	1	1
1	2	9	2	2	2	2
2	6	2	9	3	3	3
3	5	6	3	9	5	5
4	1	5	6	5	9	6
5	3	3	5	6	6	9

Sortowanie QuickSort

Tablica do posortowania:

24	11	42	65	93	54	14	82
0	1	2	3	4	5	6	7

24 11 42 65 93 54 14 82

oś (ustanawiana w połowie tablicy, wyznaczana losowo, może to być także skrajny element tablicy)

W naszym przykładzie obrana losowo.