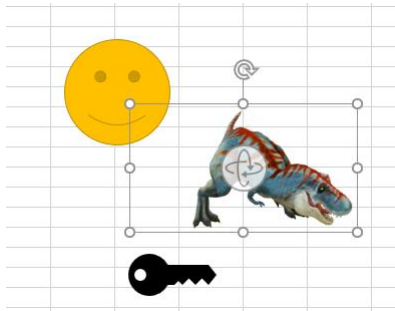


1. Klasa Shapes

Klasa reprezentuje kolekcję elementów rysunkowych (autokształty, obrazy, ikony, obiekty OLE, wykresy) należących do arkusza. Kolekcja ta jest dostępna z poziomu obiektu **Worksheet**.



Wybrane właściwości

Count – liczba kształtów w kolekcji

Range – pozwala na utworzenie podzbioru elementów klasy **Shapes**

Wybrane metody

AddShape – dodaje do kolekcji nowy autokształt

AddChart2 – dodaje do kolekcji nowy wykres

AddPicture – dodaje do kolekcji nowy obraz

Metoda AddShape

Parametry:

Type

rodzaj autokształtu, wartości typu **MsoAutoShapeType**, np.: **msoShapeRectangle**, **msoShapeOval**, **msoShapeSmileyFace**, ...)

Left, Top, Width, Height

określają położenie autokształtu na arkuszu, **Left** i **Top** to położenie lewego górnego narożnika a **Width** i **Height** to szerokość i wysokość prostokąta zawierającego autokształt

Zwracana wartość

utworzony obiekt klasy **Shape**

Przykład

W punkcie o współrzędnych (20, 20) dodawany jest prostokąt o wymiarach 100 x 100:

```
ActiveSheet.Shapes.AddShape msoShapeRectangle, 20, 20, 100, 100
```

Metoda AddChart2

Uwaga! Excel automatycznie przyjmuje dane dla wykresu w oparciu o zaznaczony zakres lub aktywną komórkę (w przypadku komórki wyznacza ciągły zakres obejmujący tę komórkę). Pusty wykres jest tworzony tylko jeśli aktywna komórka jest pusta.

Parametry (opcjonalne):

Style

określa wygląd wykresu, którego typ jest wskazany w parametrze **XlChartType**, domyślny styl ustawiany przy pomocy „-1”

XlChartType

typ wykresu np.: kolumnowy, kołowy, XY (punktowy), wartości typu **XlChartType**, np.: **xlColumnClustered**, **xlPie**, **xlXYScatterLines** ...)

Left, Top, Width, Height

znaczenie jak w przypadku metody **AddShape**

Zwracana wartość

utworzony obiekt klasy `Shape`,

Przykład

Na aktywnym arkuszu przygotowywane jest miejsce na wykres kolumnowy (wykres nie jest pokazywany ponieważ nie określony został zakres danych)

```
ActiveSheet.Shapes.AddChart2 -1, xlColumnClustered
```

Metoda `AddPicture`

Parametry:

`FileName`

wskazuje lokalizację pliku z obrazem (lokalną albo sieciową)

`LinkToFile`

określa czy do arkusza wstawiany jest tylko link do pliku (`msoTrue`) czy obraz jest fizycznie wstawiany do arkusza (`msoFalse`)

`SaveWithDocument`

określa czy obraz będzie zapisywany razem z arkuszem (`msoTrue`) czy nie (`msoFalse`)

`Left, Top, Width, Height`

znaczenie jak w przypadku metody `AddShape`

Zwracana wartość

utworzony obiekt klasy `Shape`

Przykład

Na aktywnym arkuszu wstawiane jest logo UZ

```
ActiveSheet.Shapes.AddPicture "https://www.uz.zgora.pl/images/uz-logo.png", msoFalse, _  
msoTrue, 20, 20, 100, 100
```

Przykład

Rysunek na kolejnej stronie pokazuje arkusz po uruchomieniu makra `test1`. W panelu widocznym po prawej stronie (panel włączany przy pomocy opcji *Okienko zaznaczenia* dostępnej np. z poziomu karty *Narzędzia główne* i opcji *Znajdź i zaznacz*) pokazywane są dodane kształty – Excel ponazywał kształty automatycznie nadając im w tym przypadku nazwy: *Obraz 4*, *Wykres 2* i *Prostokąt 1*.

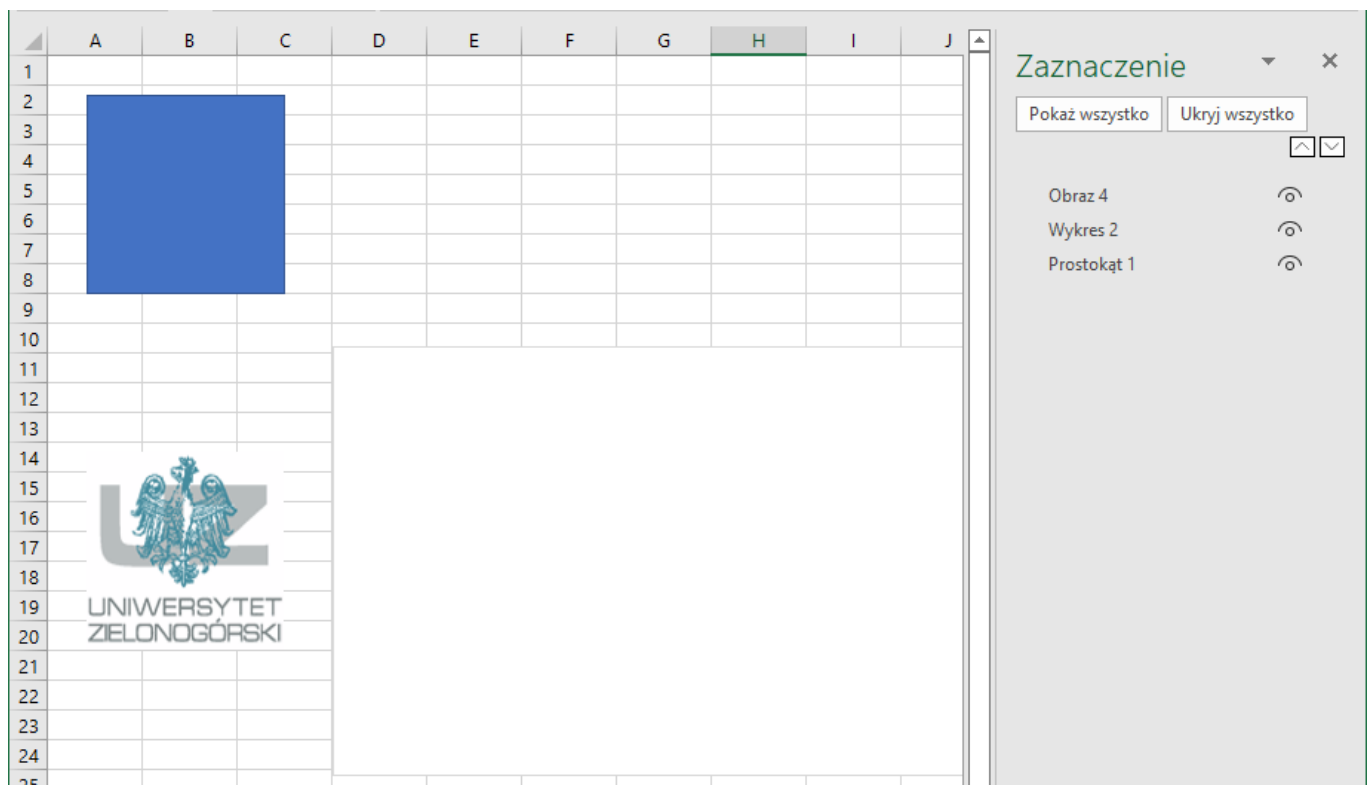
Sub `test1()`

```
ActiveSheet.Shapes.AddShape msoShapeRectangle, 20, 20, 100, 100
```

```
ActiveSheet.Shapes.AddChart2 -1, xlColumnClustered
```

```
ActiveSheet.Shapes.AddPicture "https://www.uz.zgora.pl/images/uz-logo.png", msoFalse, _  
msoTrue, 20, 200, 100, 100
```

```
End Sub
```



Klasa `Shapes` pozwala na dostęp do kształtów arkusza po podaniu numeru elementu w kolekcji lub jego nazwy.

Przykład

Makro wyświetla liczbę elementów graficznych na bieżącym arkuszu oraz ich nazwy (każdy obiekt Excela ma własność `Name`, która daje nazwę elementu lub pozwala na zmianę tej nazwy).

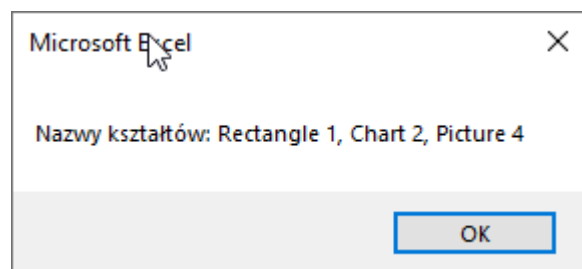
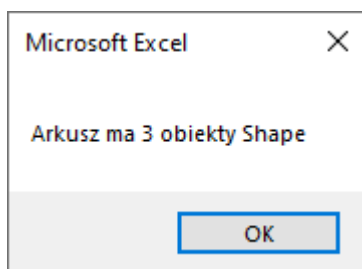
```
Sub test2()
```

```
    MsgBox "Arkusz ma " & ActiveSheet.Shapes.Count & " obiekty Shape"
```

```
    MsgBox "Nazwy kształtów: " & ActiveSheet.Shapes(1).Name & ", " & _
```

```
        ActiveSheet.Shapes(2).Name & ", " & ActiveSheet.Shapes(3).Name
```

```
End Sub
```



Uwaga! Wyświetlone nazwy elementów różnią się od tych pokazywanych w Okienku zaznaczenia. Okno zaznaczenia pokazuje nazwy, które można zobaczyć z pomocą własności `Range` klasy `Shapes` (nie wiem jaka własność przechowuje polskie nazwy elementów).

Przykład

Makro wyświetla nazwę elementu posługując się jego nazwą 😊 (dostęp do elementów kolekcji **Shapes** jest możliwy z pomocą nazwy elementu - zarówno polskiej jak i angielskiej).

Sub test3()

```
MsgBox "Pobranie nazwy na podstawie nazwy polskiej: " & _
ActiveSheet.Shapes("Obraz 4").Name & " oraz angielskiej: " & _
ActiveSheet.Shapes("Picture 4").Name
```

End Sub

2. Klasa Shape

Klasa reprezentuje element rysunkowy (autokształt, obraz, ikonę, obiekt OLE, wykres). Elementy dostępne są z poziomu kolekcji **Shapes** po podaniu numeru elementu w kolekcji lub jego nazwy.

Wybrane własności

Name

nazwa elementu

Left, Top, Width, Height

położenie i rozmiary prostokąta zawierającego element

AutoShapeType

typ (rodzaj) elementu, w przypadku autokształtu wartość odpowiada wartości podanej w metodzie **AddShape**, w przypadku wykresu wartość ustawiona na **msoShapeMixed** i nie może być zmieniana

Chart

Własność wykorzystywana tylko w przypadku elementów typu wykres, daje dostęp do obiektu **Chart** reprezentującego wykres

OnAction

nazwa makra uruchamianego po kliknięciu na elemencie

Przykład

Po utworzeniu zmieniany jest typ elementu z prostokąta na owal:

```
Dim s As Shape
```

```
Set s = ActiveSheet.Shapes.AddShape(msoShapeRectangle, 20, 20, 100, 100)
s.AutoShapeType = msoShapeOval
```

Przykład

Po utworzeniu zmieniana jest szerokość wykresu z domyślnej na 400:

```
Dim s As Shape
```

```
Set s = ActiveSheet.Shapes.AddChart2(-1, xlColumnClustered)
s.Width = 400
```

3. Klasa Chart

Klasa reprezentuje wykres. W przypadku wykresów umieszczonych na elementach rysunkowych dostęp jest możliwy z poziomu własności **Chart** elementu rysunkowego albo z kolekcji **ChartObjects** (wybranego arkusza) po podaniu numeru elementu w kolekcji lub jego nazwy.

Wybrane własności i metody

CharTitle

tytuł wykresu, własność zwraca obiekt typu **CharTitle**, który odpowiada za wygląd tytułu – ma szereg własności: **Text**, **Format**, **HorizontalAlignment**, **VerticalAlignment** ... pozwalających na ustalenie tekstu tytułu, jego formatu (koloru tła, czcionki itp.), wyrównania w poziomie i pionie, ...

SeriesCollection

metoda udostępniająca kolekcję zawierającą wszystkie serie danych tworzących wykres lub serię o określonym numerze

SetSourceData

metoda pozwalająca na podanie zakresu komórek wykorzystywanych przez wykres, metoda ma 2 parametry: **Source** – zakres (obiekt typu **Range**) i opcjonalny drugi parametr **PlotBy** określający czy serie danych są tworzone przez kolumny czy przez wiersze podanego zakresu (wartości typu **XIRowCol**, wartość **xlColumns** oznacza, że serie tworzą kolumny zakresu a **xlRows**, że serie tworzone są przez wiersze zakresu)

Przykład

Po utworzeniu wykresu ustawiany jest jego tytuł („Wykres”), określany jest zakres danych (komórki "a1:b5") i wyświetlana jest liczba serii danych

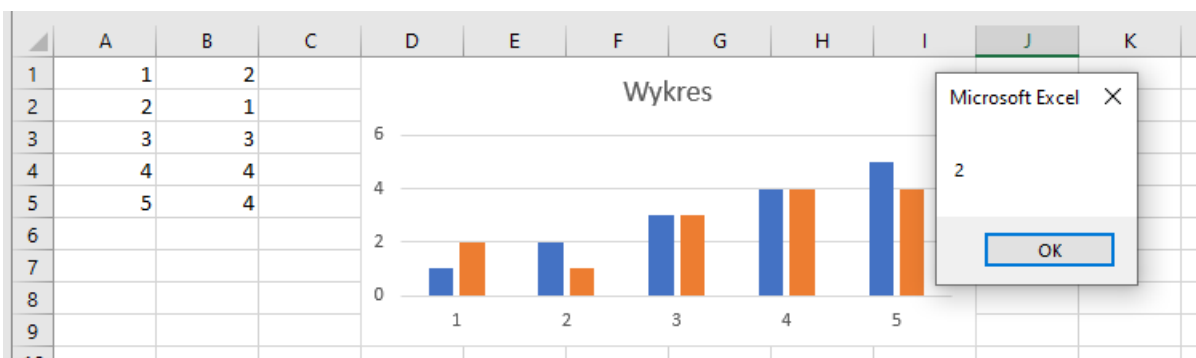
```
Dim s As Shape
```

```
Set s = ActiveSheet.Shapes.AddChart2(-1, xlColumnClustered)
```

```
s.Chart.ChartTitle.Text = "Wykres"
```

```
s.Chart.SetSourceData Range("a1:b5")
```

```
MsgBox s.Chart.SeriesCollection.Count
```



4. Klasa SeriesCollection

Klasa reprezentuje kolekcję zawierającą serie danych wykresu, elementy kolekcji dostępne są po podaniu numeru elementu w kolekcji, element kolekcji (tzn. pojedyncza seria danych) jest obiektem klasy [Series](#).

Wybrane własności i metody

Add

metoda dodaje kolejną serię lub serie danych, podstawowym parametrem metody jest parametr [Source](#) wskazujący zakres komórek z danymi kolejnej jednej lub kilku serii danych

NewSeries

metoda tworzy nową serię danych, parametry serii należy ustawić z pomocą własności utworzonego w ten sposób obiektu klasy [Series](#)

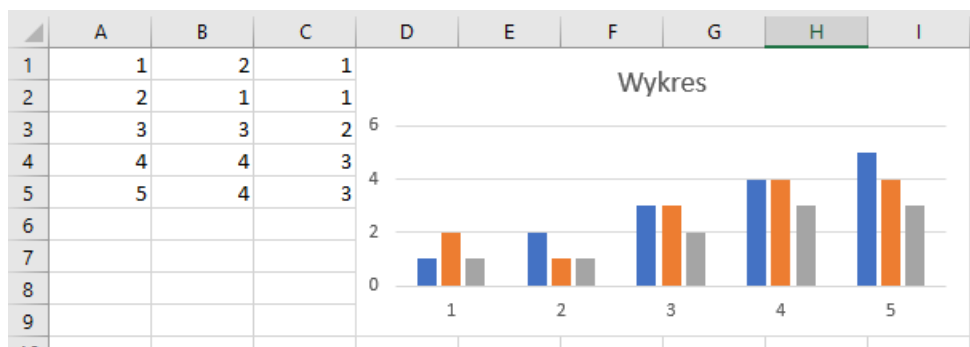
Przykład

Do pierwszego wykresu na aktywnym arkuszu dodawana jest kolejna seria danych zawierająca dane w komórkach "c1:c5")

Dim c As Chart

Set c = ActiveSheet.ChartObjects(1).Chart

c.SeriesCollection.Add Range("c1:c5")



5. Klasa Series

Klasa reprezentuje pojedynczą serię danych. Dostęp do obiektów tej klasy (poprzez numer serii danych) daje własność `SeriesCollection` obiektu `Chart`.

Wybrane własności

Values

wartości serii danych podane w postaci obiektu klasy `Range` lub tablicy liczb (wartości mogą być wymienione po kolei, oddzielone od siebie przecinkami i otoczone nawiasami klamrowymi)

XValues

jak własność `Values` ale opisuje dziedzinę wykresu

MarkerStyle

własność wykorzystywana przez wykresy: liniowy, XY (punktowy) i radarowy, decyduje o wyglądzie punktów, dopuszczalne wartości typu `XlMarkerStyle`, tzn. np. `xlMarkerStyleCircle`, `xlMarkerStyleDiamond`, `xlMarkerStyleNone` (markery w postaci kółek, rombów, bez markera)

Format

własność opisuje formatowanie wykresu z pomocą obiektu klasy `ChartFormat`

Przykład

Do pierwszego wykresu na aktywnym arkuszu dodawana jest kolejna seria danych (tym razem z wykorzystaniem metody `NewSeries`), własność `Values` jest wykorzystana do wskazania danych dla tej serii (komórki "c1:c5")

```
Dim c As Chart
Dim s As Series
```

```
Set c = ActiveSheet.ChartObjects(1).Chart
Set s = c.SeriesCollection.NewSeries
s.Values = Range("c1:c5")
```

Przykład

Do pierwszego wykresu na aktywnym arkuszu dodawana jest kolejna seria danych, własność `Values` jest wykorzystana do wskazania danych z pomocą zakresu zbudowanego z kilku zakresów (serię tworzą dane z komórek "a1:a3" i "b1:b2")

```
Dim c As Chart
Dim s As Series
```

```
Set c = ActiveSheet.ChartObjects(1).Chart
Set s = c.SeriesCollection.NewSeries
s.Values = Range("a1:a3,b1:b2")
```

Przykład

Do pierwszego wykresu na aktywnym arkuszu dodawana jest kolejna seria danych o wartościach podanych w postaci tablicy liczb (tablica zawiera wartości: 5, 4, 3, 2, 1)

```
Dim c As Chart
Dim s As Series
```

```
Set c = ActiveSheet.ChartObjects(1).Chart
Set s = c.SeriesCollection.NewSeries
s.Values = "{5,4,3,2,1}"
```



6. Klasa ChartFormat

Klasa reprezentuje format wykresu.

Wybrane własności

Line

własność opisuje formatowanie linii tworzących wykres z pomocą obiektu klasy [LineFormat](#)

Fill

własność opisuje wypełnienie elementów tworzących wykres z pomocą obiektu klasy [FillFormat](#)

7. Klasa LineFormat

Klasa reprezentuje format linii.

Wybrane własności

ForeColor

własność opisuje kolor linii z pomocą obiektu klasy [ColorFormat](#)

DashStyle

własność opisuje styl linii z pomocą wartości typu [MsoLineDashStyle](#) tzn. np.: [msoLineDash](#), [msoLineDashDot](#), [msoLineSolid](#) (linia przerywana z kresek, linia przerywana z kresek i kropek, linia ciągła)

Visible

widoczność linii

Weight

grubość linii podana jako wartość rzeczywista

Przykład

Formatowana jest pierwsza seria danych pierwszego wykresu aktywnego arkusza: ustawiana jest widoczność linii, styl linii ustawiany na linię przerywaną zbudowaną z kresek, grubości linii ustawiana jest na 2 punkty.

```
Dim c As Chart  
Dim s As Series
```

```
Set c = ActiveSheet.ChartObjects(1).Chart  
Set s = c.SeriesCollection(1)  
s.Format.Line.Visible = msoTrue  
s.Format.Line.DashStyle = msoLineDash  
s.Format.Line.Weight = 2
```

8. Klasa FillFormat

Klasa reprezentuje format wypełnienia.

Wybrane własności

ForeColor

własność opisuje kolor wypełnienia z pomocą obiektu klasy [ColorFormat](#)

Visible

widoczność wypełnienia

9. Klasa ColorFormat

Klasa reprezentuje kolor elementów, może być wykorzystywana do opisu pojedynczego koloru jak i kolorów gradientowych czy kolorów z deseniem.

Wybrane własności

RGB

własność typu `Long` opisuje kolor zdefiniowany przy pomocy składowych RGB

Przykład

Formatowana jest pierwsza seria danych pierwszego wykresu aktywnego arkusza: ustawiana jest widoczność wypełnienia i jego kolor na kolor zielony z Logo Microsoftu (czerwony: 124, zielony: 187, niebieski: 0).

```
Dim c As Chart
```

```
Dim s As Series
```

```
Set c = ActiveSheet.ChartObjects(1).Chart
```

```
Set s = c.SeriesCollection(1)
```

```
s.Format.Fill.Visible = msoTrue
```

```
s.Format.Fill.ForeColor.RGB = RGB(124, 187, 0)
```