

PRACA Z FORMULARZAMI UserForm

ZFPBIG – LABORATORIUM

Formularz - okno lub okno dialogowe. Możemy powiedzieć że formularze są kontenerami dla formantów. Formularz jest obiektem, oznacza to że posiada on właściwości, metody i reaguje na zdarzenia. W edytorze VBA formularz nosi nazwę UserForm.

Formant - obiekt, który można umieścić na formularzu, posiadający własny zestaw rozpoznawalnych właściwości, metod i zdarzeń. Formantów można używać do pobierania i wyświetlania danych oraz wyzwalania procedur obsługi zdarzeń. Większością formantów można operować używając metod. Niektóre formanty są interaktywne (reagują na akcje użytkownika), podczas gdy inne są statyczne (dostępne tylko poprzez kod programu).

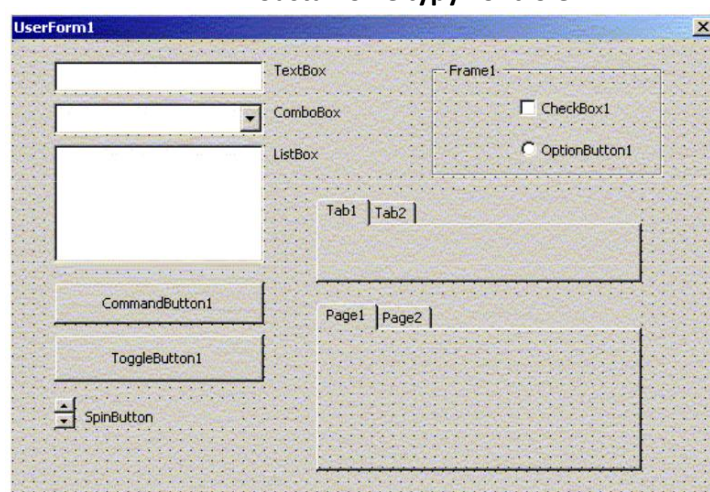
KONTROLKI FORMULARZA

Kontrolki (formanty) to elementy interfejsu, które programista wykorzystuje w swoim programie. Są to obiekty, których zdarzenie trzeba zaprogramować. Rozmieszcza się je na formularzu. Służą one np. do wprowadzania danych (pole tekstowe, suwaki), oraz ich wyprowadzania (etykiety). Są one ważną częścią programu, gdyż dzięki nim możemy w łatwy sposób sterować pracą programu. Rozmieszczenie kontrolki na formularzu jest bardzo ważne, gdyż świadczy to o estetyce programu i łatwości jego obsługi.

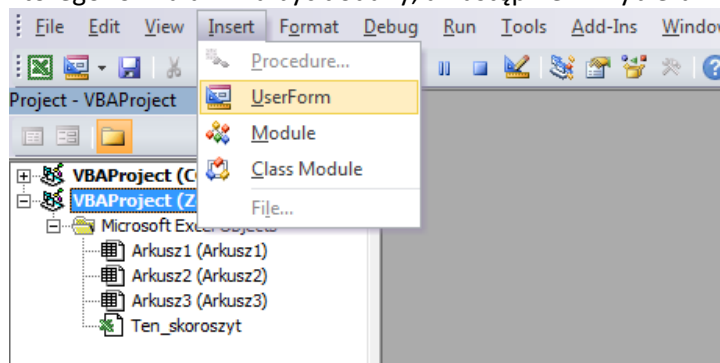
Program Microsoft Excel ma dwa typy formantów:

- **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.

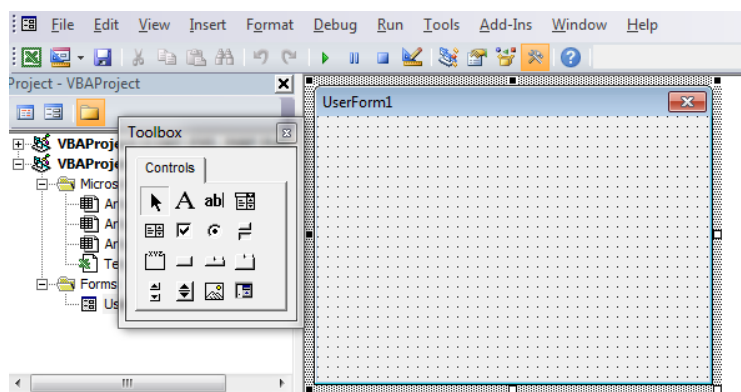
Podstawowe typy kontrolki



Aby dodać formularz, najpierw w oknie Project Explorer edytora VBA zaznaczamy plik Excela, do którego formularz ma być dodany, a następnie wybieramy z menu 'Insert' polecenie 'UserForm'.



Pusty formularz o nazwie 'UserForm1' został dodany do pliku VBA.xls, dodana została kategoria 'Forms' w której jest nasz formularz.



Uwaga!!

Tytuł formularza (caption) jest wyświetlany na górze formularza, na niebieskim pasku i nie zawsze pokrywa się z nazwą formularza (name).

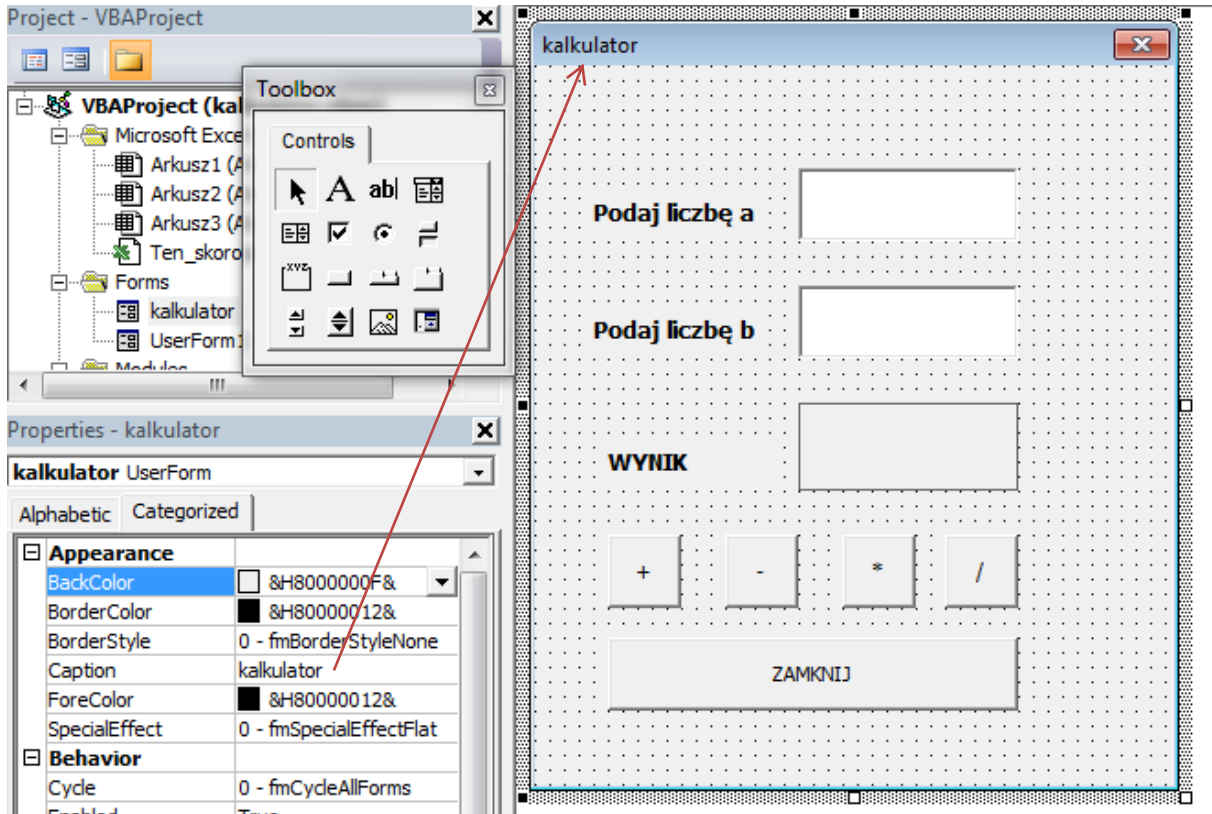
W naszym przykładzie dla tych dwóch właściwości wprowadziliśmy tą samą wartość, tj.

- Name → kalkulator
- Caption → kalkulator

Zadanie1.

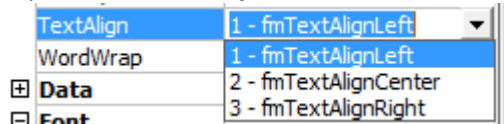
Celem zadania jest zaprojektowanie prostego kalkulatora realizującego podstawowe działania arytmetyczne.

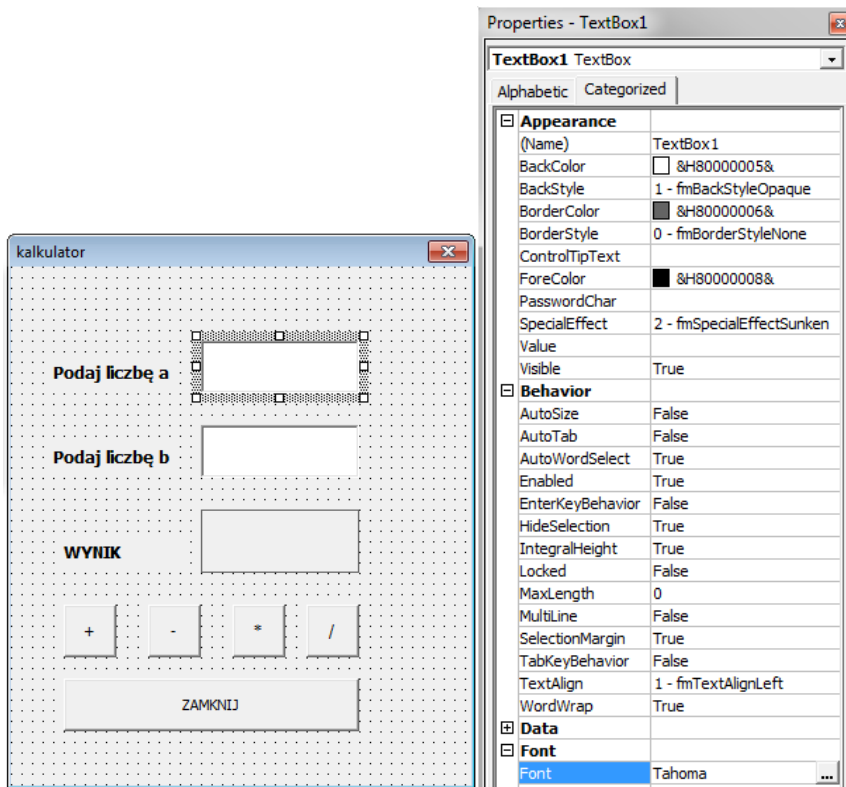
1. W oknie Properties ustaw właściwość Caption na Kalkulator natomiast nazwę formularza pozostawiamy domyślną UserForm1
2. Rozmieśćmy odpowiednie kontrolki formularza na przygotowanym obszarze, wg schematu:



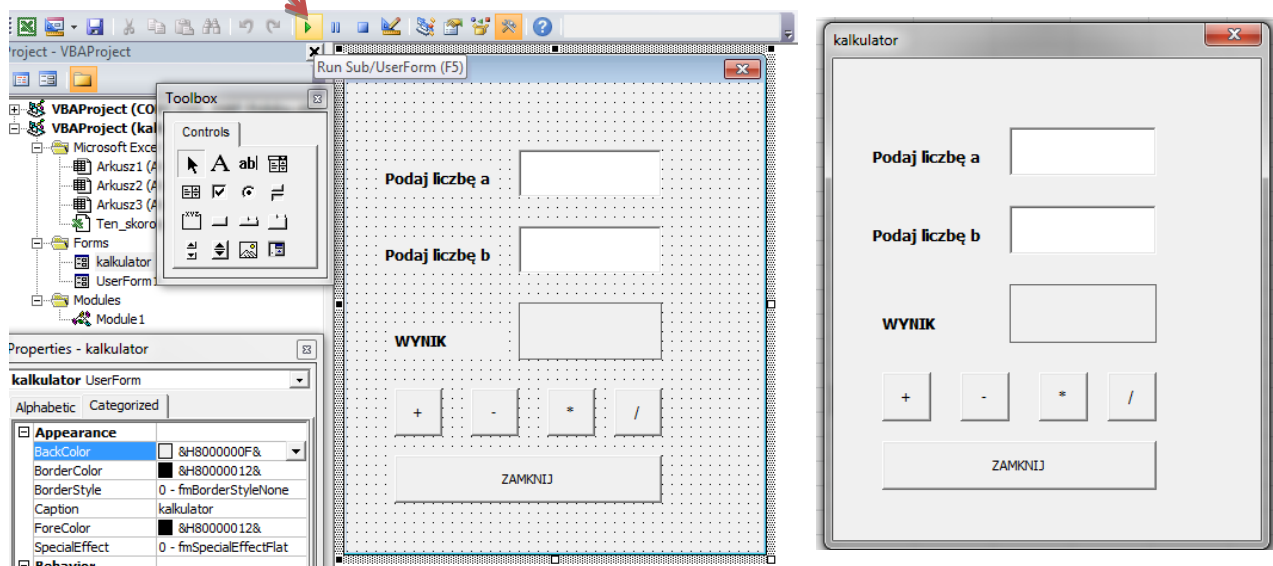
3. Po wprowadzeniu kontrolki możemy edytować jej ustawienia w oknie właściwości (properties), możliwe jest to wtedy gdy dana kontrolka jest aktywna. Dla wprowadzonych kontroltek zostały wprowadzone zmiany dotyczące:

- czcionki (wielkość, kolor, pogrubienie),
- zawartość, tj. pojawiający się wewnątrz napis (caption),
- nazwa formantu (name),
- wyrównania wewnątrz obszaru formantu:





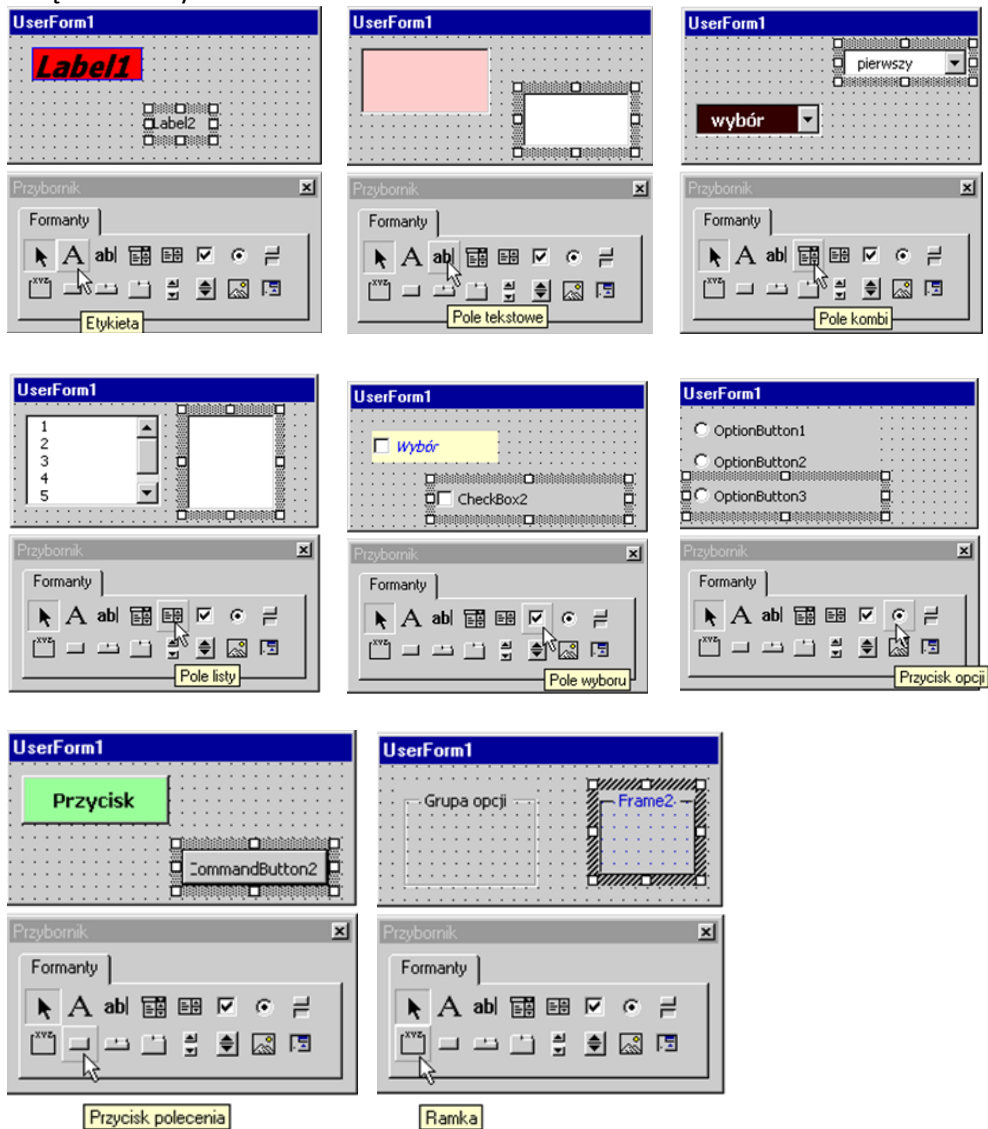
- Po zaprojektowaniu formularza możemy sprawdzić jego sposób wyświetlania (oraz działania, ale to dopiero jak napiszemy makra obsługujące formularz) w programie za pomocą klawisza funkcyjnego F5 lub kliknięciem na ikonę, znajdującą się na pasku narzędziowym.



Końcowym etapem projektu jest oprogramowanie kalkulatora. Z punktu widzenia systemu operacyjnego każde kliknięcie na wybrany przycisk kalkulatora jest pewnym zdarzeniem, na które oprogramowanie powinno w określony sposób zareagować np. przez wyświetlenie w polu wynik wartości. Sposób reakcji określa procedura obsługi zdarzenia, którą twórca programu powinien zaprojektować.

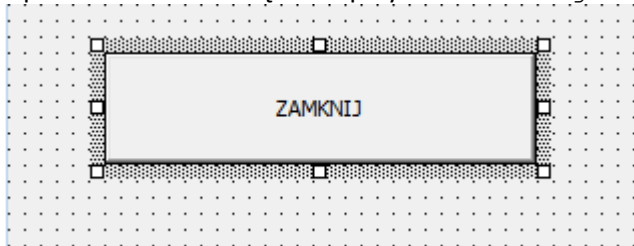
W zależności od użytego obiektu będzie on spełniał odpowiednią funkcje:

- wyświetlania danych,
- wprowadzania danych,
- wykonywania akcji,
- zwiększania czytelności formularza.



Dwukrotne kliknięcie na wybranym obiekcie formularza odsyła nas do edytora VBA gdzie w sposób automatyczny rozpoczynana jest procedura mająca za zadanie obsłużyć dane zdarzenie (przypisane do klikniętego obiektu).

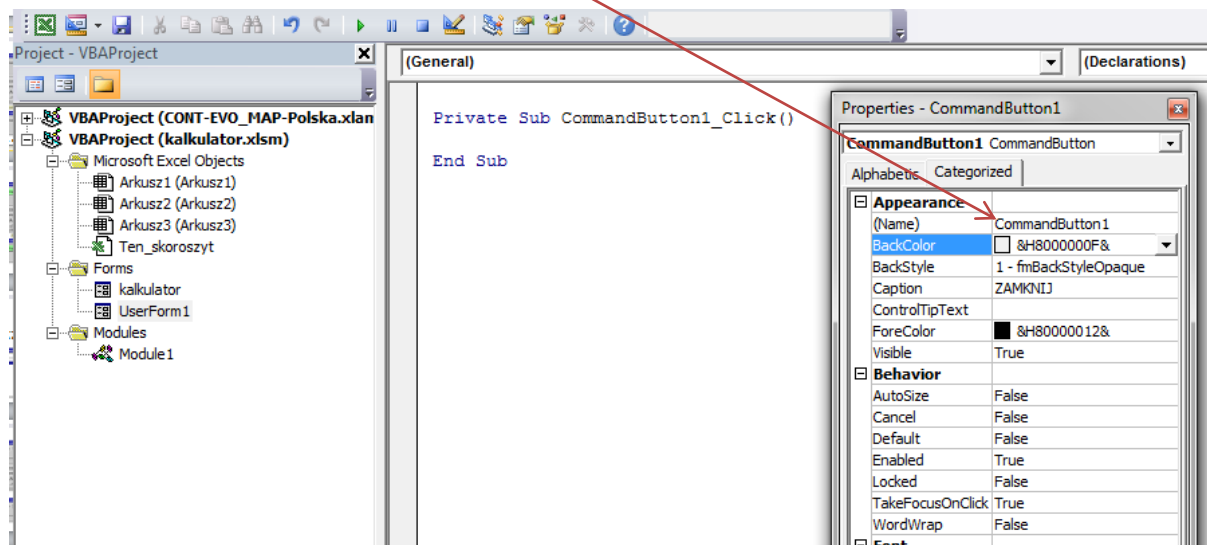
Np. dwukrotne kliknięcie na przycisku zamknij naszego formularza



Spowodowało otwarcie edytora vba oraz zapisanie początku i końca procedury przypisanej do tego zdarzenia. Nazwa takiej procedury w VBA tworzona jest automatycznie wg schematu:

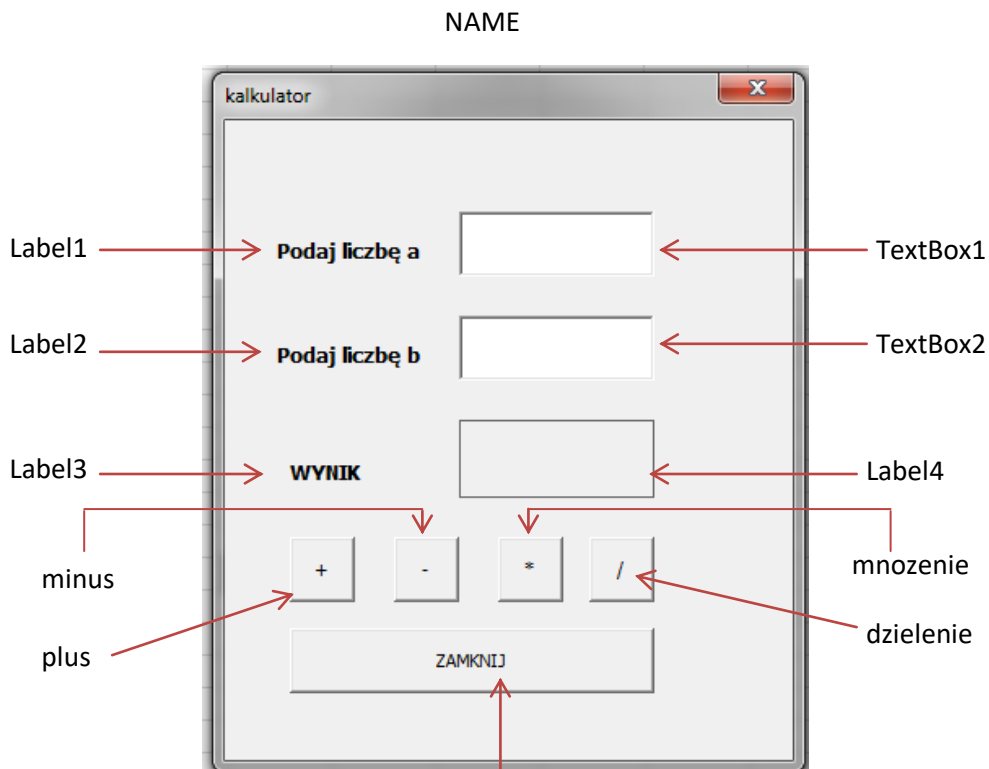
Nazwa-obiektu-kontrolki_ zdarzenie

w naszym przypadku: `CommandButton1_Click()`



Procedury obsługi zdarzeń mogą przetwarzać dane reprezentowane przez zmienne i stałe języku VBA, przez obiekty arkusza kalkulacyjnego jak i przez właściwości związane z obiektami (kontrolkami) zaprojektowanego formularza.

W naszym arkuszu poszczególne obiekty mają przypisane następujące nazwy (**name**), i te będziemy wykorzystywać tworząc ich obsługę:



```
Private Sub plus_Click CommandButton1
```

5. Pisanie makr, które będą obsługiwały formularz

Przedstawione poniżej rozwiązania niech posłużą do weryfikacji Twoich opracowań. Zanim skopiujesz (najlepiej nie kopiuj) spróbuj samodzielnie napisać wszystkie procedury.

```
Private Sub plus_Click()  
a = TextBox1  
b = TextBox2  
wynik = a + b  
Label4 = wynik  
  
End Sub
```

```
Private Sub minus_Click()  
Dim a As Double  
Dim b As Double  
a = TextBox1  
b = TextBox2  
wynik = a - b  
Label4 = wynik  
  
End Sub
```

```
Private Sub mnozenie_Click()  
Dim a As Double  
Dim b As Double  
a = TextBox1  
b = TextBox2  
wynik = a * b  
Label4 = wynik  
  
End Sub
```

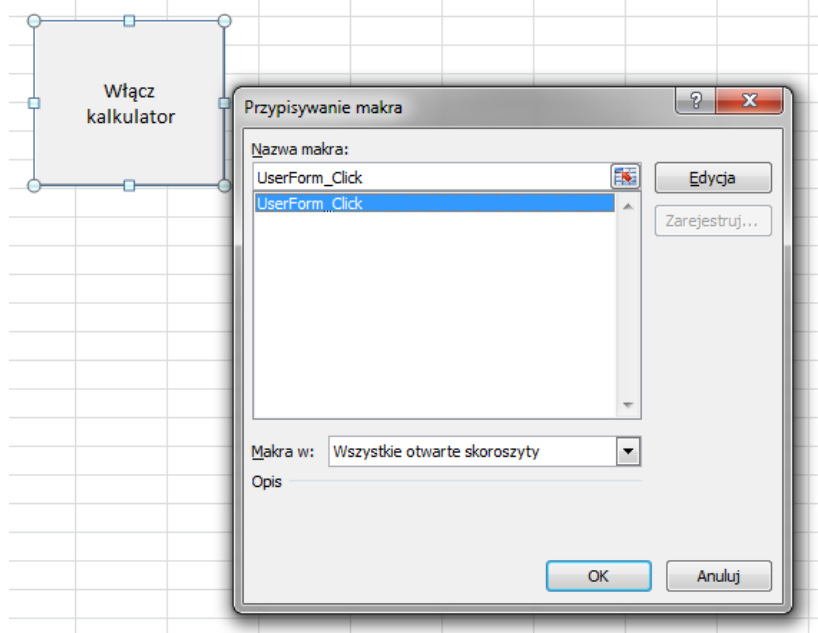
```
Private Sub dzielenie_Click()  
Dim a As Double  
Dim b As Double  
a = TextBox1  
b = TextBox2  
If b <> 0 Then  
wynik = a / b  
Label4 = wynik  
Else  
wynik = "nie dziel przez 0!!"  
Label4 = wynik  
End If  
  
End Sub
```

```
Private Sub CommandButton1_Click()  
Unload Me  
  
End Sub
```

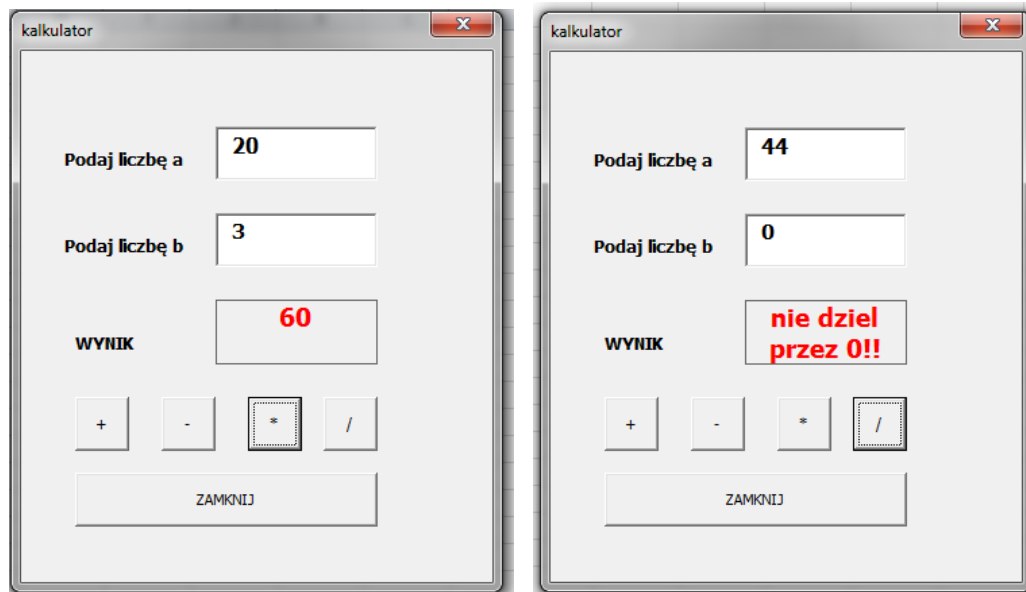
6. Na ostatnim etapie tworzenia formularza zaprojektujmy makro, które pozwoli nam go wywołać z poziomu excela. W tym celu należy wstawić nowy moduł (insert/moduł) a następnie wprowadzić do niego następujący zapis:

```
Sub UserForm_Click()  
kalkulator.Show  
End Sub
```

W następnym kroku, wstaw obiekt w arkuszu1, do którego przypiszesz niniejsze makro.



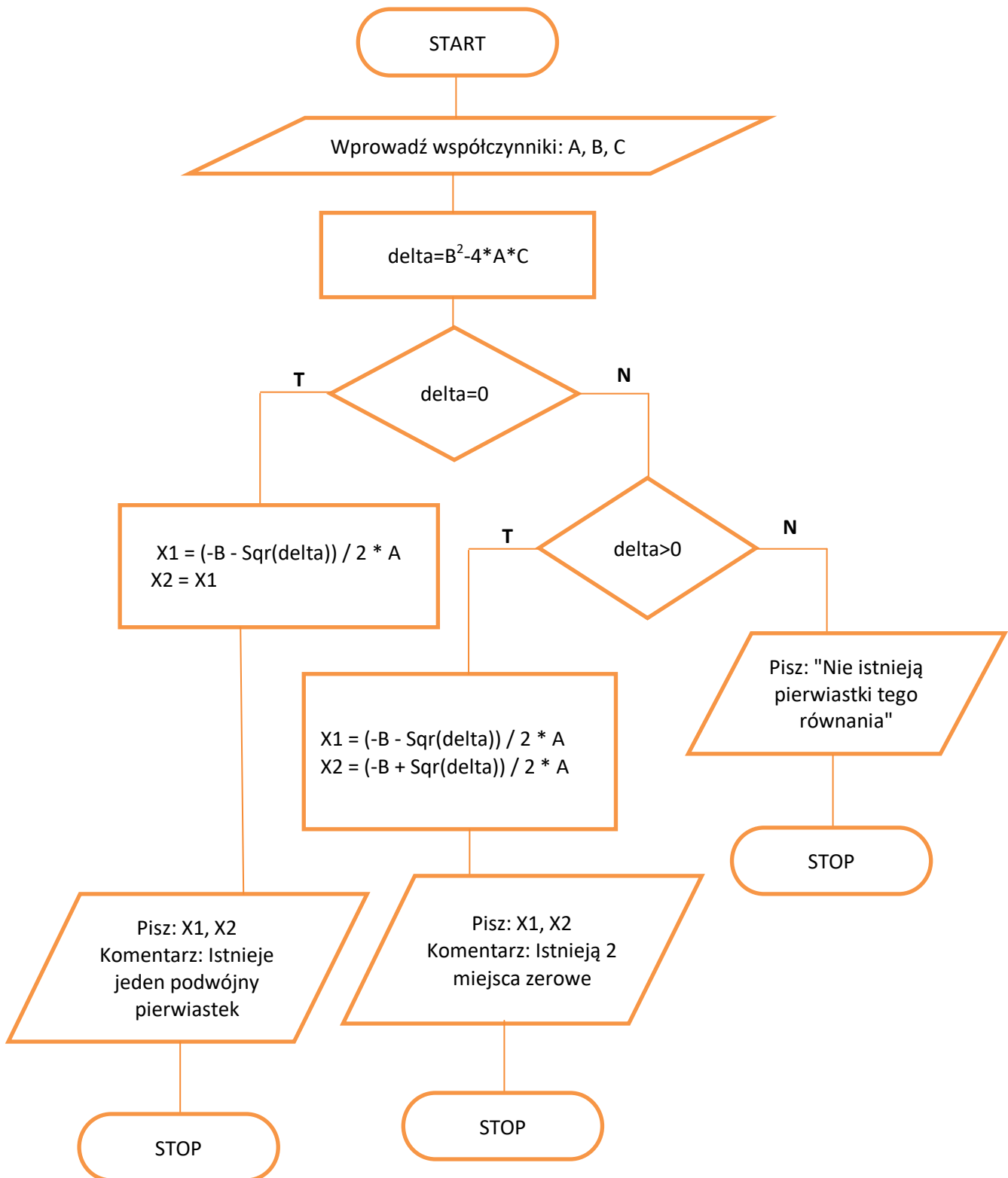
Po kliknięciu na przycisku zostaje uruchomiona procedura wywołująca formularz kalkulatora:



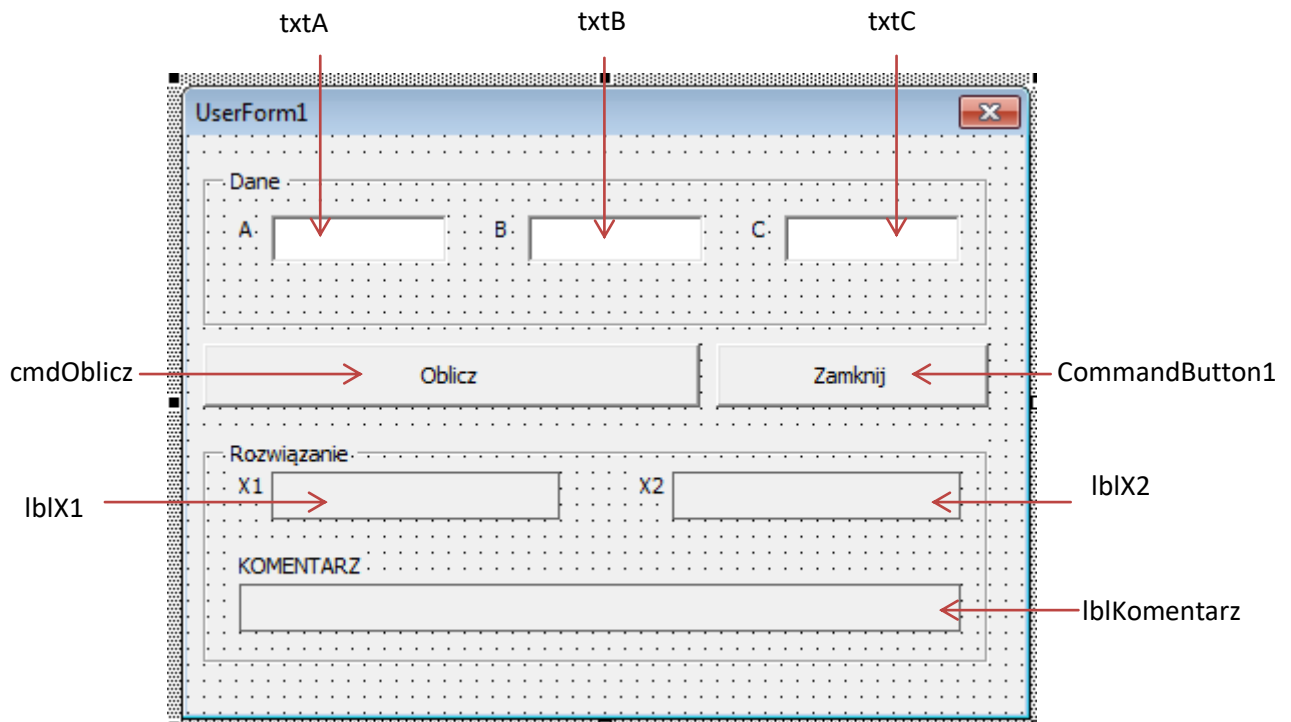
Zadanie 2

Zaprojektuj formularz za pomocą którego wprowadzimy dane do obliczeń dla wyznaczenia pierwiastków trójmianu kwadratowego:

$$Ax^2+Bx+C=0$$



Schemat formularza wraz z użytymi nazwami obiektów:



Pisanie makr, które będą obsługiwały formularz

```
Private Sub cmdOblicz_Click()  
  
Dim A, B, C As Single  
Dim delta As Single  
Dim X1, X2 As Single  
Dim komentarz As String  
  
A = txtA.Value  
    If IsNumeric(A) = False Then  
        MsgBox ("Podaj cyfrę")  
        Exit Sub  
    End If  
B = txtB.Value  
    If IsNumeric(B) = False Then  
        MsgBox ("Podaj cyfrę")  
        Exit Sub  
    End If  
C = txtC.Value  
    If IsNumeric(C) = False Then  
        MsgBox ("Podaj cyfrę")  
        Exit Sub  
    End If  
  
    delta = B ^ 2 - 4 * A * C
```

```

Select Case delta
Case Is > 0
X1 = (-B - Sqr(delta)) / 2 * A
X2 = (-B + Sqr(delta)) / 2 * A
komentarz = "Istnieją 2 miejsca zerowe: " & "X1 = " & X1 & ",
X2 = " & X2
lblKomentarz = komentarz
lblX1 = X1
lblX2 = X2
Case Is < 0
lblX1 = "-"
lblX2 = "-"
lblKomentarz = "Nie istnieją pierwiastki tego równania"
Case Is = 0
X1 = (-B - Sqr(delta)) / 2 * A
X2 = X1
lblKomentarz = "Istnieje jeden podwójny pierwiastek = " & X1
lblX1 = X1
lblX2 = X2
End Select
End Sub

Private Sub CommandButton1_Click()
Trojmian.Hide
End Sub

```

Zadanie 3

Zaprojektuj ankietę, z której wyniki będą wprowadzane do arkusza2 w kolejnych komórkach:

The image shows a Windows form titled "ankieta" with a dotted grid background. It contains three input sections:

- Podaj płeć:** A group box containing two radio buttons labeled "Kobieta" and "Mężczyzna".
- wykształcenie:** A group box containing a dropdown menu with the text "Podaj wykształcenie" and a label "wykształcenie" to its right.
- Zarobki brutto:** A group box containing a spinner control with a label "zarobki" to its right.

At the bottom of the form is a large button labeled "ok".

```

Private Sub ComboBox1_Change()
    Label2 = ComboBox1
End Sub

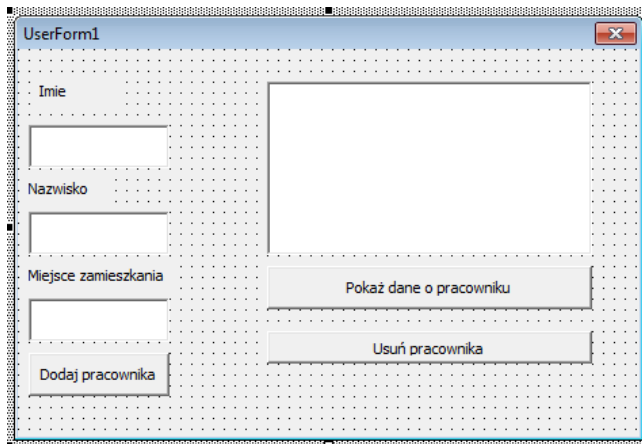
Private Sub ScrollBar1_Change()
    Label1 = ScrollBar1
End Sub

Private Sub CommandButton1_Click()
If CommandButton1 = True Then
    Worksheets("Arkusz1").Range("B3").Value = "K"
Else
    Worksheets("Arkusz1").Range("B3").Value = "M"
End If
    Worksheets("Arkusz1").Range("B1") = Label1
    Worksheets("Arkusz1").Range("B2") = Label2
    Unload Me
End Sub

```

Zadanie 4

Przeanalizuj kod i działanie udostępnionego makra w arkuszu o nazwie bazadanych.xlms pozwalającego na zapisywanie i usuwanie rekordów w bazie danych w odpowiednim miejscu.



JAK ZA POMOCĄ VBA FORMATOWAĆ KOMÓRKI W EXCEL?

Formatowanie arkusza Excel za pomocą kodu VBA jest niezbędne jeżeli chcemy:

ujednolicić wygląd generowanych dokumentów

odciążyć pracowników od pracochłonnej czynności związanych z wyglądem dokumentów

Formatowanie tekstu

Podstawowe właściwości tekstu wpisanego w komórkę arkusza, przechowywane są w obiekcie **Cells.Font**.

Aby zmienić rozmiar tekstu wpisujemy wielkość w punktach:

```
Cells(1, 1).Font.Size = 15
```

Pogrubienie tekstu:

```
Cells(1, 1).Font.Bold = True
```

Pochylenie tekstu:

```
Cells(1, 1).Font.Italic = True
```

Podkreślenie tekstu:

```
Cells(1, 1).Font.Underline = True
```

Przekreślenie tekstu:

```
Cells(1, 1).Font.Strikethrough = True
```

Indeks dolny:

```
Cells(1, 2).Font.Subscript = True
```

Indeks górny:

```
Cells(1, 3).Font.Superscript = True
```

Ponadto możliwe jest określenie **kierunku tekstu** w komórce:

```
Cells(1, 3).Orientation = 90
```

Powyższy kod spowoduje, że tekst w komórce C1 wpisywany będzie w kierunku od dołu do góry.

Wartość 0 oznacza standardowy tekst poziomy.

W celu **wyrównania tekstu w poziomie** użyjemy składni:

```
Cells(1, 3).HorizontalAlignment = xlCenter
```

Parametry w formatowaniu poziomym tekstu:

- xlCenter - tekst wyśrodkowany
- xlDistributed - tekst zawsze zaczyna się przy lewej a kończy przy prawej krawędzi komórki
- xlJustify - tekst wyjustowany
- xlLeft - tekst z lewej
- xlRight - tekst z prawej

Wyrównanie tekstu w pionie:

```
Cells(1, 3).VerticalAlignment = xlCenter
```

i parametry wyrównania pionowego:

- xlBottom - tekst na dole komórki
- xlCenter - tekst w środku komórki
- xlDistributed - tekst zaczyna się i kończy przy krawędziach komórki
- xlJustify - tekst wyjustowany w pionie
- xlTop - tekst na górze komórki

Kolorowanie tekstu oraz komórek

Aby zmienić **kolor** tekstu komórki używamy składni:

```
Cells(1, 1).Font.Color = rgbRed
```

gdzie rgbRed oznacza stałą koloru czerwonego.

Jeżeli nie możemy znaleźć specyficznego koloru możemy zapisać go za pomocą **składni RGB**:

```
Cells(1, 1).Font.Color = RGB(255, 0, 0)
```

Zmiana koloru wypełnienia komórki:

```
Cells(1, 1).Interior.Color = rgbYellow
```

Kod spowoduje pokolorowanie komórki na kolor żółty.

