

Programowanie obiektowe



Okna dialogowe i formularze użytkownika

**standardowe okna dialogowe
tworzenie formularzy użytkownika**

Okna dialogowe dostępne jako procedury, funkcje lub metody

- `MsgBox` – okno komunikatu (procedura lub funkcja),
- `InputBox` – odczyt wartości (funkcja lub metoda obiektu **Application**),
- `GetOpenFileName` – wybór pliku do odczytu (metoda obiektu **Application**),
- `GetSaveFileName` – wybór pliku do zapisu (metoda obiektu **Application**).

Wbudowane okna dialogowe

`Dialogs` jest własnością obiektu **Application**, która daje dostęp do kolekcji wbudowanych okien dialogowych (1426 elementów w Excel 2019). Odwołanie do elementów kolekcji odbywa się poprzez indeksy typu `XLBuiltinDialog`. Elementami kolekcji są obiekty typu **Dialog**, które udostępniają metodę funkcyjną `Show`:

```
Function Show([Arg1], [Arg2], ... [Arg30]) As Boolean
```

Metoda wyświetla okno dialogowe i zwraca `True` jeżeli zostało zamknięte przyciskiem OK lub `False` jeżeli zostało zamknięte przyciskiem Anuluj. Liczba i typ argumentów są zależne od typu okna dialogowego.

Uwaga: uruchomienie okna z kolekcji `Dialogs` powoduje wykonanie funkcji programu Excel związanej z danym oknem dialogowym (np. modyfikację zawartości lub formatu aktywnej komórki arkusza).

Standardowe okna dialogowe – MsgBox

```
MsgBox(prompt As String, _  
        Optional buttons As VbMsgBoxStyle = vbOKOnly, _  
        Optional title As String = "Microsoft Excel") As Integer
```

Argumenty

- `prompt` – tekst wyświetlany w oknie komunikatu
- `buttons` – zestaw przycisków dostępnych w oknie:
 - `vbOKOnly`, `vbOKCancel`, `vbAbortRetryIgnore`, `vbYesNoCancel`, `vbYesNo`, `vbRetryCancel`
 - `vbCritical`, `vbQuestion`, `vbExclamation`, `vbInformation`
 - `vbDefaultButton1`, ..., `vbDefaultButton4`
 - wartości określające typ przycisku oraz okna mogą być sumowane
- `title` – tytuł okna (domyślnie „Microsoft Excel”)

Rezultat: `vbOK`, `vbCancel`, `vbAbort`, `vbRetry`, `vbIgnore`, `vbYes`, `vbNo`

Uwaga: `MsgBox` może być wywołana jako procedura (argumenty podawane bez nawiasów) lub jako funkcja (argumenty w nawiasach). Rezultat wykonania (kod naciśniętego przycisku) można pobrać tylko w drugim przypadku.

MsgBox – przykład

```
Public Sub Komunikat1()  
MsgBox "To jest okienko komunikatu, które nie zwraca wartości", _  
vbInformation, "Komunikat1"
```

```
End Sub
```

```
Public Sub Komunikat2()
```

```
Dim odp As Integer
```

```
odp=MsgBox("Czy programowanie w VBA jest fajne?", vbQuestion +  
vbYesNoCancel + vbDefaultButton1 )
```

```
If odp = vbYes Then
```

```
Range("E6").Value = "Też tak myślę"
```

```
ElseIf odp = vbNo Then
```

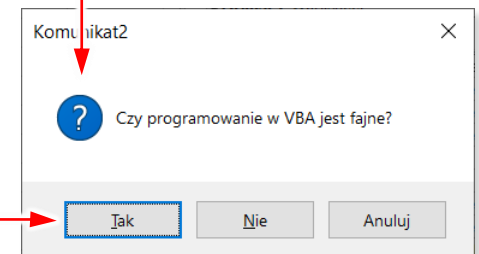
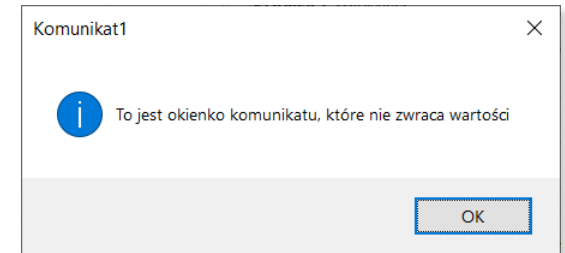
```
Range("E6").Value = "Zastanów się..."
```

```
ElseIf odp = vbCancel Then
```

```
Range("E6").Value = "Nie potrafisz podjąć decyzji?"
```

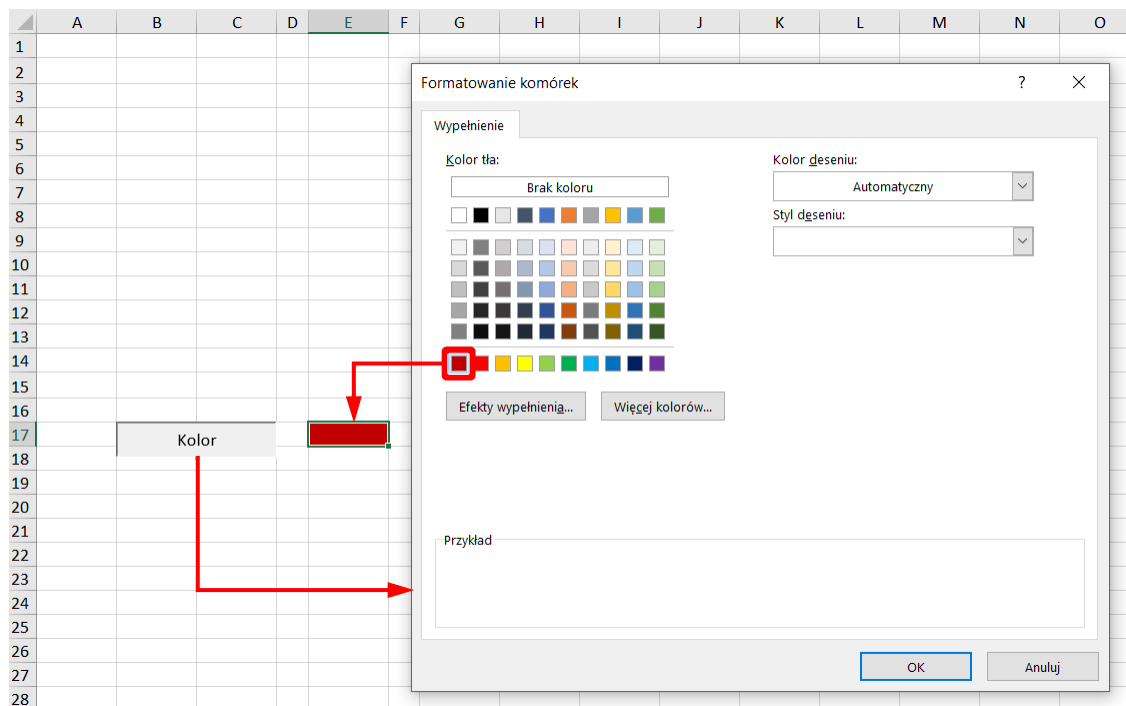
```
End If
```

```
End Sub
```



Wbudowane okna dialogowe – xlDialogPatterns

```
Public Sub Kolor()  
    Dim w As Boolean  
    Range("E17").Activate  
    w = Application.Dialogs(xlDialogPatterns).Show()  
    If w Then Range("G17").Value = "Wybrany kolor: " & _  
        Range("F17").DisplayFormat.Interior.Color  
End Sub
```



Formularz użytkownika (*User Form*) – formularz zaprojektowany przez użytkownika jako interfejs dla własnej funkcji zaprogramowanej w VBA. Powinien być używany, gdy związana z nim operacja wymaga ustawienia wielu opcji i nie jest możliwe wykorzystanie standardowych okien dialogowych dostępnych w Excel-u.

Formant (inaczej kontrolka) – element formularza, który pozwala na wprowadzenie danych lub wybór opcji określających działanie projektowanej operacji.

Uwaga: każdy formularz i format jest obiektem, który posiada zestaw własności, metod i zdarzeń, na które reaguje podczas wykonania programu.

Przykład

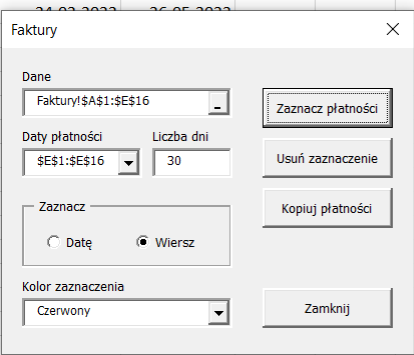
Automatyczny wybór faktur, których płatności przypadają w określonym przedziale czasowym.

Funkcje:

- Zaznacza płatności
- Usuwa zaznaczenie
- Kopiuje płatności (sporządza raport)

	A	B	C	D	E	F	G	H
1	Lp.	Nazwa	Kwota	Data wystawienia	Data płatności			
2	1	A-Z Elementy dekoracyjne	2 300,00 zł	16.02.2022	22.03.2022			
3	2	Hydrostal Sp. z o.o. ZP	980,20 zł	04.02.2022	16.06.2022			
4	3	Kaiser Schody Sp. z o.o.	3 575,00 zł	15.02.2022	12.04.2022			
5	4	RECTOR Polska Sp. z o.o.	227,50 zł	28.02.2022	19.06.2022			
6	5	ComfortHouse sp. z o.o.	1 232,80 zł	05.02.2022	28.05.2022			
7	6	AB KLIMA Chigo.pl	2 500,00 zł	06.02.2022	09.03.2022			
8	7	AS Instrument Polska	128,75 zł	19.02.2022	06.04.2022			
9	8	Bikrol-Stal s.c.	3 750,00 zł	18.02.2022	23.05.2022			
10	9	Hydrostal Sp. z o.o. ZP	235,50 zł	15.02.2022	22.06.2022			
11	10	Karo. Bramy garażowe, ogrodzenia	3 200,00 zł	06.02.2022	23.05.2022			
12	11	Paroc Panel System	783,25 zł	27.02.2022	13.03.2022			
13	12	AS Instrument Polska	287,50 zł	28.02.2022	21.04.2022			
14	13	ACO sp. z o.o. sp. komandytowa	1 872,90 zł	15.02.2022	15.04.2022			
15	14	Bikrol-Stal s.c.	4 320,00 zł	24.02.2022	25.05.2022			
16	15	AQUAFORM Inc. Sp.z o.o.	588,30 zł					
17								
18								
19								
20								
21								
22								
23								
24								
25								
26								
27								
28								
29								
30								

Wynik działania programu uruchomionego 1.04.2022

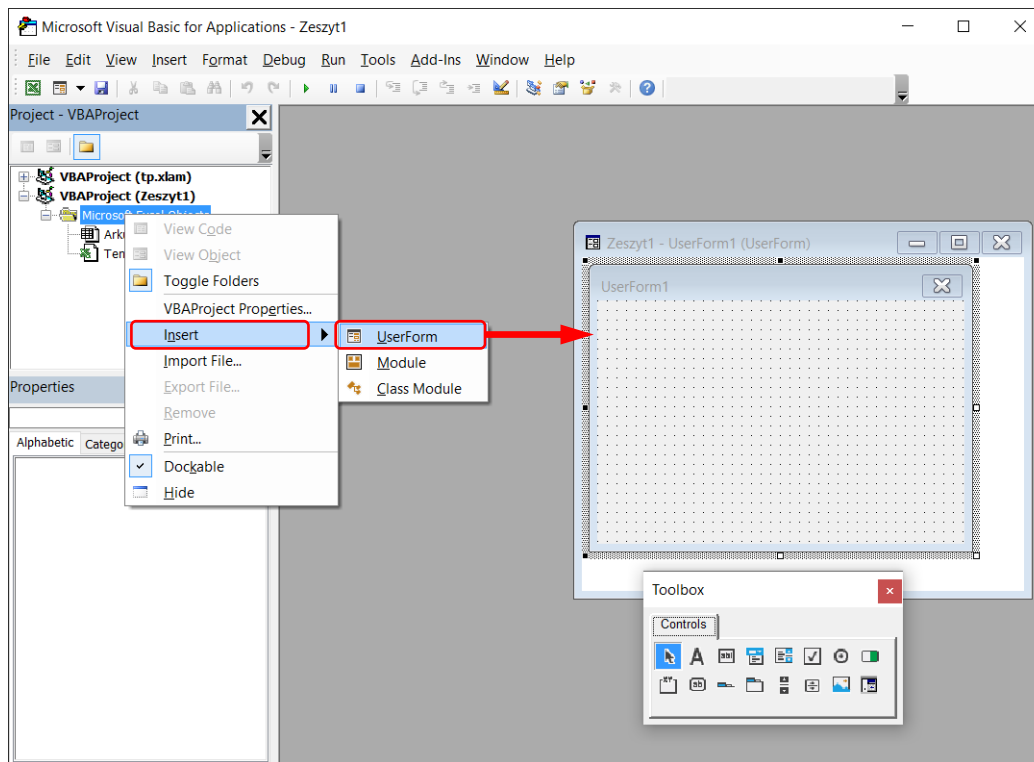


Tworzenie formularza

Menu: Insert->UserForm

Prawy przycisk w oknie projektu: Insert->UserForm

Okno toolbox (formanty): View->Toolbox



Podstawowa wersja programu jest dostępna w materiałach na stronie przedmiotu jako UserForm1

Formanty, właściwości, metody

ComboBox – lista rozwijalna, własności i metody:

- **ListIndex** – numer pozycji wybranej w liście (numeruje od zera),
- **ListStyle** – właściwość, sposób wyświetlania pozycji w liście (opcje lub lista),
- **Style** – właściwość, sposób działania listy (edycja dostępna lub zablokowana),
- **AddItem** – metoda, dodaje nową pozycję do listy.

CommandButton – przycisk polecenia, uruchamia pewną funkcję.

Frame – grupa formantów.

Label – etykieta, statyczny tekst zazwyczaj opisujący inne elementy.

OptionButton – przycisk opcji, własność **Value** określa czy opcja jest wybrana.

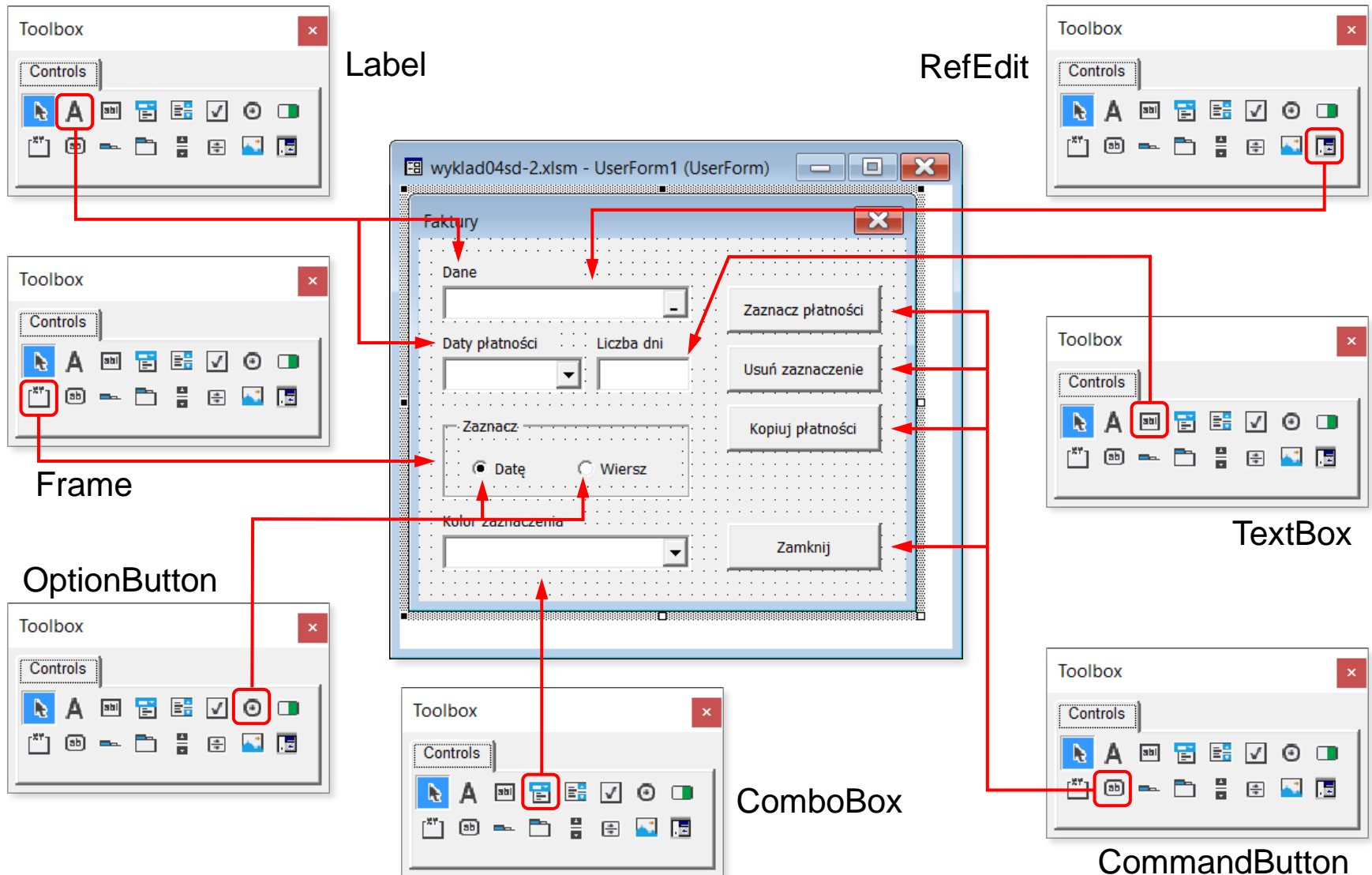
RefEdit – zaznaczenie zakresu, własność **Value** określa wybrany zakres (string).

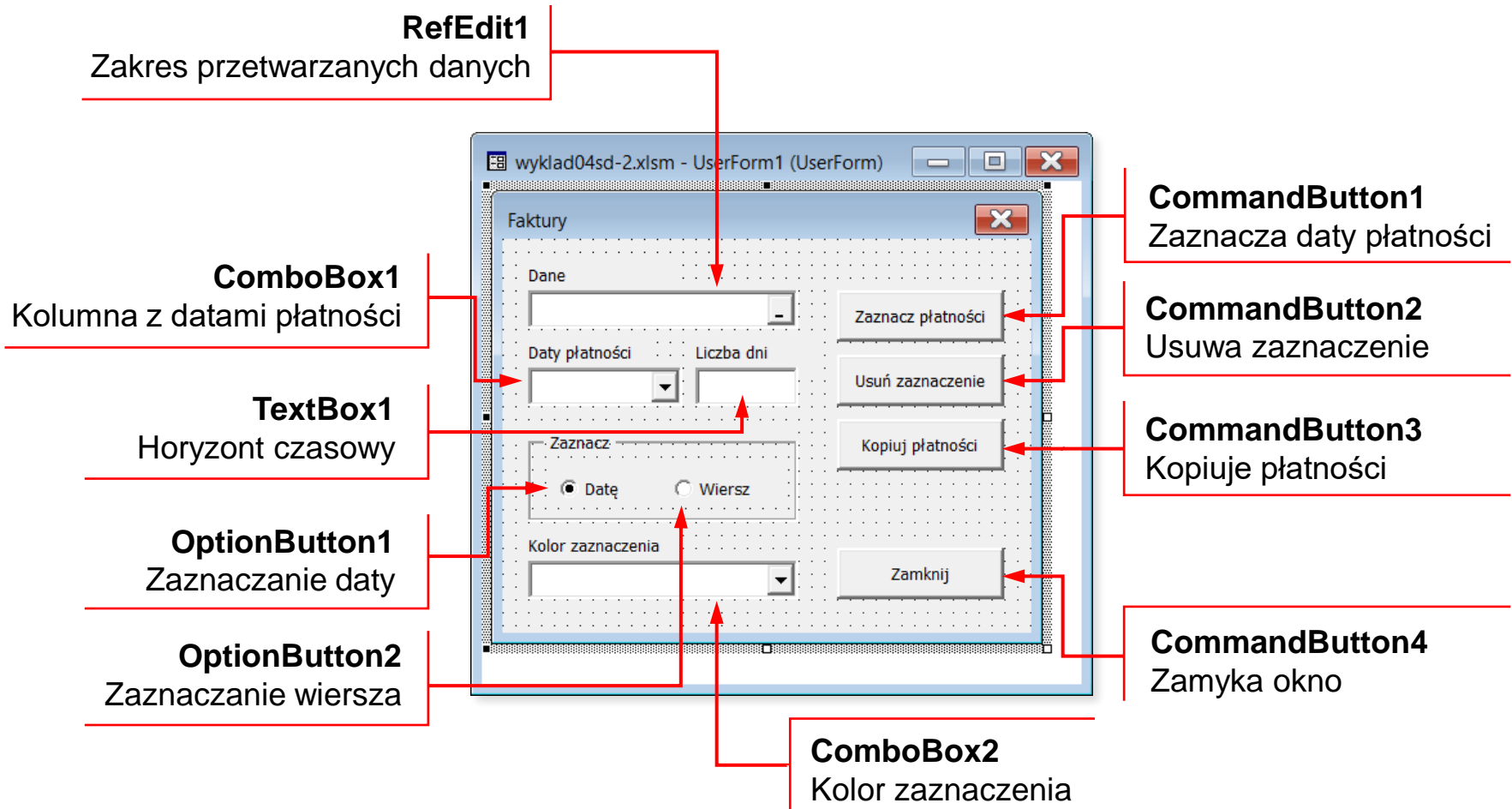
TextBox – pole edycyjne, dane wprowadzane z klawiatury, wartość wprowadzoną w polu określają własności **Text** i **Value**.

Uwaga 1: Formanty są identyfikowane przez unikalne nazwy określone własnością **Name**.

Uwaga 2: Wiele formantów (np. **CommandButton**, **Frame**, **Label**, **OptionButton**) posiada własność **Caption**, która określa nagłówek (opis) elementu.

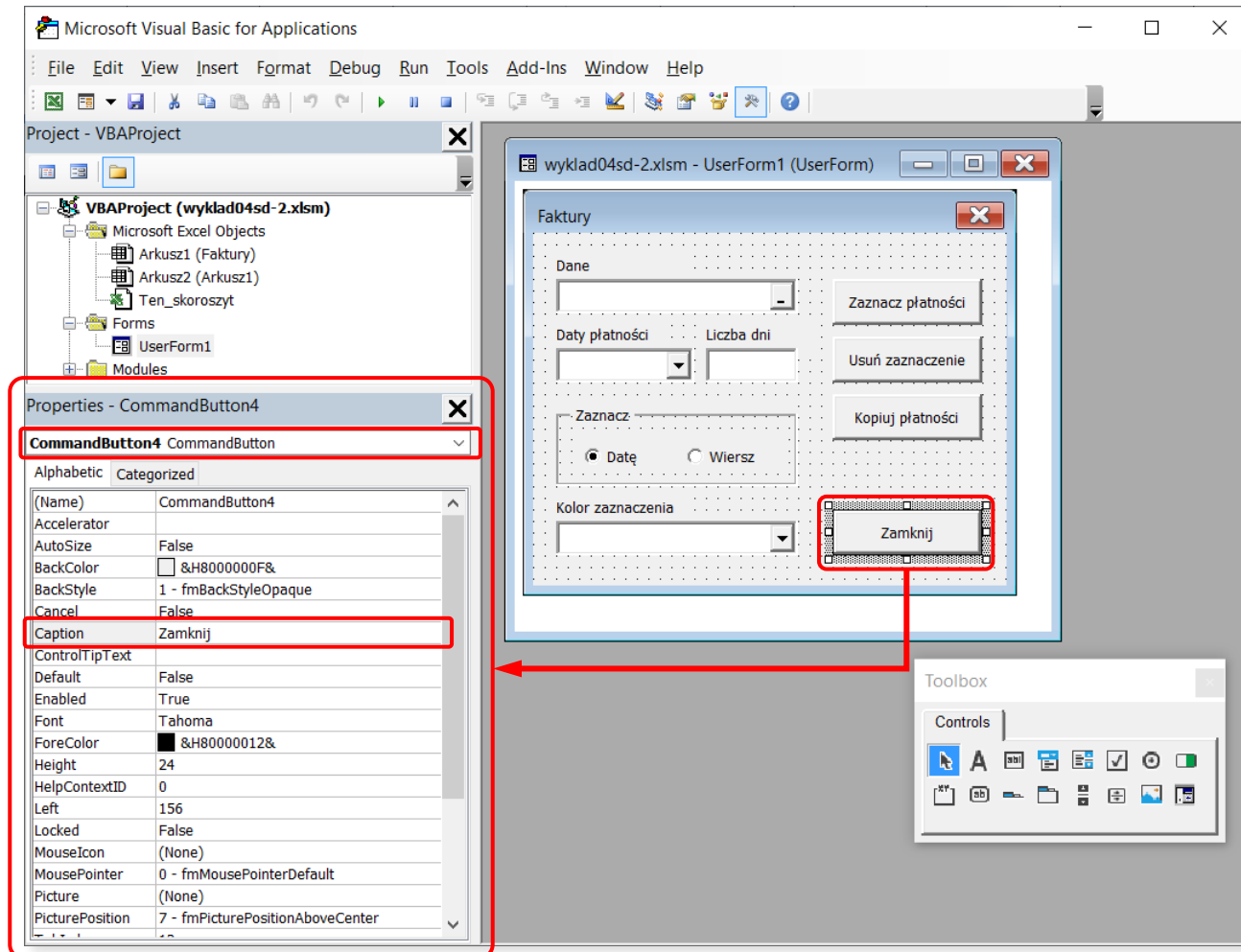
Projekt interfejsu





Uwaga: Przyjęto nazwy (własność **Name**), które VBA domyślnie przypisuje wstawianym formantom (zgodnie ze schematem `nazwa_klasy<nr>`).

Edycja własności formantów: okno Properties (View->Properties Window lub F4)



Zdarzenie – sygnał informujący o wystąpieniu określonej sytuacji rozpoznawanej i rejestrowanej przez system komputerowy. Może być generowane przez system lub działania podejmowane przez użytkownika. Typ zdarzenia określa jego wyzwalacz, czyli akcja lub sytuacja, która je wywołała.

Procedura zdarzeniowa – makro uruchamiane po wystąpieniu zdarzenia.

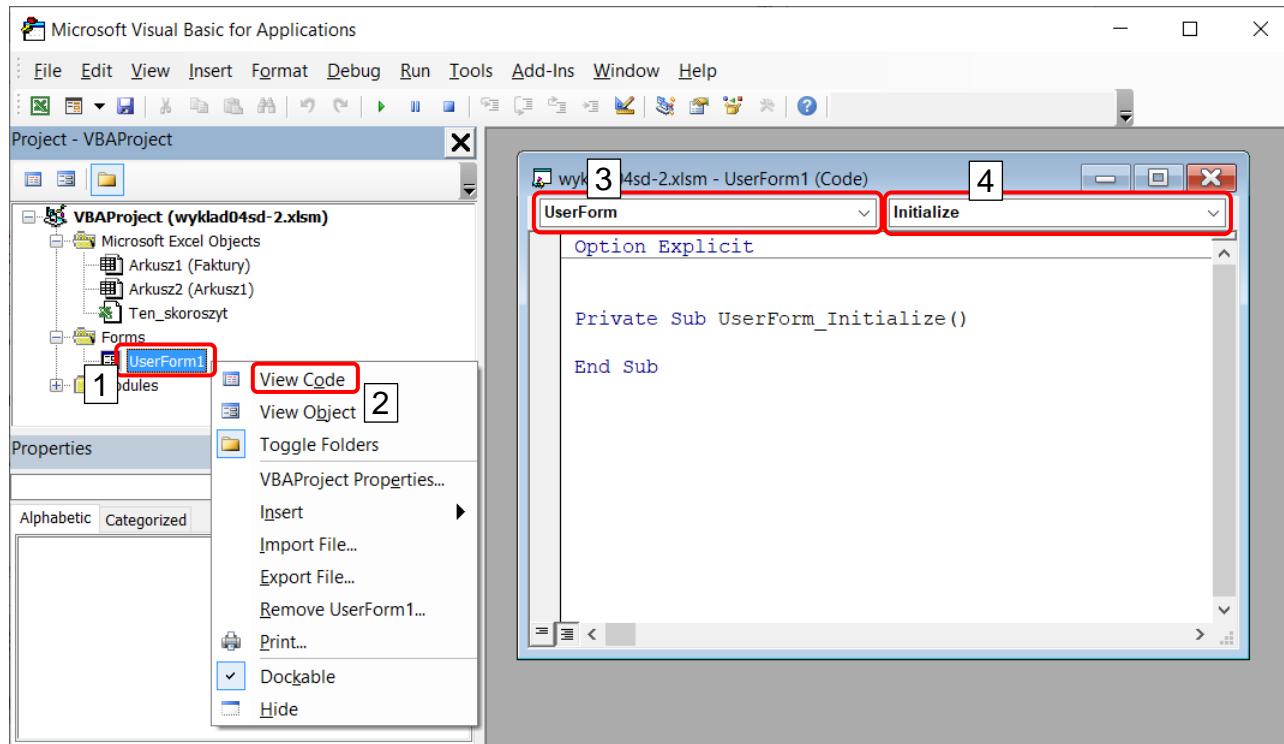
Przykładowe zdarzenia formantów

- ❑ Initialize*, Terminate* – zdarzenia formularza (obiekt UserForm), generowane odpowiednio po uruchomieniu i przed zamknięciem formularza. Zazwyczaj używane do zainicjalizowania pozostałych formantów oraz zapisania ich stanu.
- ❑ Enter, Exit* – zdarzenia formantów umożliwiających wprowadzanie danych, generowane w chwili wejścia (wskazania) i wyjścia z formantu.
- ❑ Change* – zdarzenie formantów umożliwiających wprowadzanie danych, generowane po każdej zmianie wartości wprowadzonej do formantu.
- ❑ Click* – zdarzenie generowane po kliknięciu lewym przyciskiem myszy, dostępne dla wszystkich formantów, najczęściej używane przez CommandButton do uruchomienia operacji powiązanej z przyciskiem.

* zdarzenia używane w przykładowej aplikacji

Uwaga: kod procedur zdarzeniowych związanych z formantami formularza musi być zapisany w module formularza. Procedury umieszczone w innych modułach nie będą uruchamiane.

Tworzenie procedury zdarzeniowej



1. Wybór formularza zawierającego formant, którego zdarzenie będzie definiowane.
2. Menu podręczne, opcja View Code.
3. Wybór obiektu (formantu) .
4. Wybór zdarzenia (powstaje szkielet procedury zdarzeniowej)

Alternatywa: kliknięcie prawym przyciskiem myszy na formancie i opcja View Code

Zdarzenie UserForm_Initialize

```
Private Sub UserForm_Initialize()  
Worksheets(WSInvoice).Activate           'aktywacja arkusza z danymi  
RefEdit1.Text = ActiveSheet.UsedRange.Address 'pobranie zakresu  
FillColumns           'wypełnienie listy dostępnych kolumn (patrz s.15)  
With ComboBox2           'wypełnienie listy kolorów zaznaczenia...  
    .AddItem "Czerwony"           '...i wybór pierwszego (indeks 0)  
    .AddItem "Cyjan"  
    .AddItem "Magenta"  
    .AddItem "Niebieski"  
    .AddItem "Zielony"  
    .AddItem "Żółty"  
    .ListIndex = 0  
End With  
End Sub
```

Stałe zdefiniowane w module (nazwy arkuszy)

```
Const WSInvoice As String = "Faktury"  
Const WSCopy As String = "Kopia"
```

Uwaga: użycie stałych w odwołaniach do arkuszy ułatwia parametryzację programu (można przyjąć inne nazwy arkuszy zmieniając wartości stałych).

Procedura FillColumns

Private Sub FillColumns()

Dim r **As** Range

Dim c **As** Range

ComboBox1.Clear

1 **Set** r = Range(RefEdit1.Text)

For Each c **In** r.Columns 2

3 ComboBox1.AddItem c.Address

Next

End Sub

	A	B	C	D	E
1	Lp.	Nazwa	Kwota	Data wystawienia	Data płatności
2	1	A-Z Elementy dekoracyjne	2 300,00 zł	16.02.2022	22.03.2022
3	2	Hydrostal Sp. z o.o. ZP	980,20 zł	04.02.2022	16.06.2022
4	3	Kaiser Schody Sp. z o.o.	3 575,00 zł	15.02.2022	12.04.2022
5	4	RECTOR Polska Sp. z o.o.	227,50 zł	28.02.2022	19.06.2022
6	5	ComfortHouse sp. z o.o.	1 232,80 zł	05.02.2022	28.05.2022
7	6	AB KLIMA Chigo.pl	2 500,00 zł	06.02.2022	09.03.2022
8	7	AS Instrument Polska	128,75 zł	19.02.2022	06.04.2022
9	8	Bikrol-Stal s.c.	3 750,00 zł	18.02.2022	23.05.2022
10	9	Hydrostal Sp. z o.o. ZP	235,50 zł	15.02.2022	22.06.2022
11	10	Karo. Bramy garażowe, ogrodzenia	3 200,00 zł	06.02.2022	23.05.2022
12	11	Paroc Panel System	783,25 zł	27.02.2022	13.03.2022
13	12	AS Instrument Polska	287,50 zł	28.02.2022	21.04.2022
14	13	ACO sp. z o.o. sp. komandytowa	1 872,90 zł	15.02.2022	15.04.2022
15	14	Bikrol-Stal s.c.	4 320,00 zł	24.02.2022	26.05.2022
16	15	AQUAFORM Inc. Sp.z o.o.	588,30 zł	17.02.2022	26.06.2022
17					

- \$A\$1:\$A\$16
- \$B\$1:\$B\$16
- \$C\$1:\$C\$16
- \$D\$1:\$D\$16
- \$E\$1:\$E\$16

3 AddItem

Fakтуры1

Dane

\$A\$1:\$E\$16

Daty płatności

Liczba dni

\$A\$1:\$A\$16

\$B\$1:\$B\$16

\$C\$1:\$C\$16

\$D\$1:\$D\$16

\$E\$1:\$E\$16

Wiersz

2 r.Columns (kolekcja kolumn w zakresie)

1 RefEdit1.Text (UsedRange, zmienna "r")
\$A\$1:\$E\$16

CommandButton1_Click (zaznaczenie płatności)

```
1 Private Sub CommandButton1_Click()  
2 Dim dates As Range, d As Range  
3 Dim t As Integer: Dim c As Long  
4 Range(RefEdit1.Text).Font.ColorIndex = xlColorIndexAutomatic  
5 Set dates = Range(ComboBox1.Text)  
6 Set dates = dates.SpecialCells(xlCellTypeConstants, xlNumbers)  
7 t = Int(TextBox1.Value)  
8 c = Choose(ComboBox2.ListIndex + 1, vbRed, ..., vbYellow)  
9 For Each d In dates  
10     If d.Value >= Date And d.Value <= Date + t Then  
11         If OptionButton1.Value Then d.Font.Color = c _  
12         Else ActiveSheet.UsedRange.Rows(d.Row).Font.Color = c  
13     End If  
14 Next  
15 End Sub
```


CommandButton1_Click – opis procedury zdarzenia

2–3. Deklaracja zmiennych:

`dates` – kolekcja komórek zawierających daty płatności, wskazana przez użytkownika w `ComboBox1`,

`d` – pojedyncza data (zmienna używana w pętli `For Each`),

`t` – przedział czasu (liczba dni), w którym płatności powinny być znalezione,

`C` – kolor zaznaczenia.

4. Usunięcie aktualnie ustawionego koloru.

5–8. Odczyt danych wejściowych (patrz s.18).

9–13. Pętla `For Each`, powtarza operację kolorowania dla każdej daty płatności (kolejne daty są umieszczane w zmiennej `d`):

9. Sprawdzenie czy data mieści się w zakresie,

11. Jeżeli użytkownik wybrał kolorowanie daty (`OptionButton1`) ustawienie koloru w pojedynczej komórce,

12. Jeżeli użytkownik wybrał kolorowanie wiersza (`OptionButton2`) ustawienie koloru w całym wierszu (`EntireRow`) używanego zakresu (`UsedRange`).

CommandButton1_Click – odczyt danych wejściowych

Kolumna z datami płatności

```
Set dates = Range(ComboBox1.Text)
```

```
Set dates = dates.SpecialCells(xlCellTypeConstants, xlNumbers)
```

Wartość wybrana w ComboBox1 (string) jest zamieniana na zakres (obiekt Range), odrzucane są wszystkie komórki, które nie zawierają wartości numerycznych.

Przedział czasowy

```
t = Int(TextBox1.Value)
```

Wartość wprowadzona w TextBox1 jest konwertowana na Integer (funkcja Int).

Kolor zaznaczenia

```
c = Choose(ComboBox2.ListIndex + 1, _  
vbRed, vbCyan, vbMagenta, vbBlue, _  
vbGreen, vbYellow)
```

Funkcja Choose wybiera element z listy, którego numer został podany jako pierwszy parametr.

Własność ListIndex określa numer elementu wybranego w ComboBox2 (numerując od zera), na tej podstawie określany jest wskazany kolor.

```
x = Choose(n, w1 w2, w3)
```

można zapisać jako:

```
If n = 1 Then  
    x = w1  
ElseIf n = 2 Then  
    x = w2  
ElseIf n = 3 Then  
    x = w3  
EndIf
```

CommandButton3_Click (kopiowanie płatności), cz.1

```
1 Private Sub CommandButton3_Click()  
2 Dim dates As Range, d As Range, r As Range  
3 Dim t As Integer, n As Integer  
4 On Error GoTo NoWorksheet  
5 Set r = Worksheets(WSCopy).UsedRange  
6 On Error GoTo 0  
7 Set dates = Range(ComboBox1.Text)  
8 Set dates = dates.SpecialCells(xlCellTypeConstants, xlNumbers)  
9 t = Int(TextBox1.Value)  
  
    ' zasadnicza część kodu, patrz s.20  
  
21 Exit Sub  
22 NoWorksheet:  
23 Worksheets.Add(After:=Worksheets(WSInvoice)).Name = WSCopy  
24 Worksheets(WSInvoice).Activate  
25 ActiveSheet.Rows(1).Copy Worksheets(WSCopy).Rows(1)  
26 Resume  
27 End Sub
```

CommandButton3_Click (kopiowanie płatności), cz.2

```
1 Private Sub CommandButton3_Click()  
2-9 ' deklaracja zmiennych i odczyt danych wejściowych (patrz s.19)  
10 n = 0  
11 For Each d In dates  
12     If d.Value >= Date And d.Value <= Date + t Then  
13         Set r = Worksheets(WSCopy).Rows(Worksheets(WSCopy). _  
                UsedRange.Rows.Count + 1)  
14         d.EntireRow.Copy r  
15         n = n + 1  
16         r.Cells(1, 1).Value = n  
17     End If  
18 Next  
19 Worksheets(WSCopy).Columns.AutoFit  
20 Worksheets(WSCopy).UsedRange.Font.ColorIndex=xlColorIndexAutomatic  
21-26 ' obsługa błędów (patrz s.19)  
27 End Sub
```

CommandButton3_Click – opis procedury zdarzenia

- 2-3. Deklaracja zmiennych:
 - dates, d – kolekcja komórek z datami, pojedyncza data (For Each),
 - r – zmienna pomocnicza określa miejsce kopiowania wiersza
 - t, n – przedział czasu, numer kopiowanego wiersza.
- 4-6. Próba odczytu zakresu z arkusza "Kopia" (WSCopy). Jeżeli arkusz nie istnieje wystąpi błąd (patrz 22-26) i zostanie utworzony.
- 7-9. Odczyt danych wejściowych, analogicznie do CommandButton1_Click.
- 10. Wyzerowanie licznika kopiowanych płatności (zmienna n)
- 11-18. Pętla For Each, wykonuje operację kopiowania dla kolejnych płatności:
 - 12. Sprawdzenie czy data mieści się w zakresie,
- 13-14. Pobranie wiersza docelowego do zmiennej r i skopiowanie aktualnej płatności,
- 15-16. Powiększenie licznika płatności i ustawienie numeru w pierwszej komórce.
- 19-20. Dopasowanie szerokości kolumn i usunięcie koloru w kopii.
- 21. Zakończenie procedury gdy nie wystąpił wyjątek,
- 22-26. Obsługa błędu z p.5 (patrz s.2-15). Jeżeli arkusz "Kopia" nie istnieje jest tworzony (Add), z odpowiednią nazwą (Name), a do pierwszego wiersza kopiowany jest nagłówek (Copy). Po obsłudze błędu program jest wznawiany.

Pozostałe zdarzenia, uruchomienie formularza

CommandButton2_Click – usuwanie zaznaczenia

```
Private Sub CommandButton2_Click()  
    Range(RefEdit1.Text).Font.ColorIndex = xlColorIndexAutomatic  
End Sub
```

CommandButton4_Click – zamykanie okna

```
Private Sub CommandButton4_Click()  
    Unload UserForm1  
End Sub
```

Uruchomienie formularza

```
Public Sub Faktury1()  
    UserForm1.Show  
End Sub
```

Uwaga: Makro uruchamiające formularz powinna być napisane w zwykłym module (nie w module formularza), co zapewni jego widoczność dla pozostałych elementów aplikacji.

Analiza działania formularza Faktury

Potencjalne błędy związane z użyciem formularza Faktury

1. Brak arkusza zawierającego dane płatności.
2. Nieprawidłowe dane wejściowe (zakres danych, kolumna z datami, liczba dni).
3. Lista dostępnych kolumn nie aktualizuje się po zmianie zakresu

Możliwe udoskonalenia

1. Usuwanie istniejącej kopii płatności (zabezpiecza przed dodawaniem nowych płatności do już istniejących).
2. Ostrzeżenie przed usuwaniem istniejącej kopii płatności (zabezpiecza przed skasowaniem poprzedniego raportu, który nie został przesłany).
3. Informacja o braku płatności w określonym przedziale czasowym (wyjaśnia dlaczego żadna płatność nie została zaznaczona/skopiowana).
4. Zapisanie stanu formularza przed zamknięciem i przywrócenie po powtórnym uruchomieniu (użytkownik nie musi ponownie ustawiać danych wejściowych).

Uwaga: prawidłowo zaprojektowany program powinien być zabezpieczony przed błędnymi danymi i zapewnić wygodę pracy. Nakład niezbędny do wprowadzenia takich udoskonalień może przekraczać czas przeznaczony na realizację podstawowej wersji projektu.

Rozszerzona wersja programu, jest dostępna w materiałach na stronie przedmiotu jako UserForm2

Komunikaty wyświetlane w programie

Zabezpieczenia i udoskonalenia opisane na s.23 wymagają wprowadzenia komunikatów związanych z powiadomieniami o błędach oraz informacjami przekazywanymi użytkownikowi. Część może powtarzać się w różnych punktach programu, stąd wygodne jest zdefiniowanie odpowiednich stałych zawierających treść komunikatu (ułatwia to ujednolicenia komunikatów w całym programie).

Stałe zdefiniowane w rozszerzonej wersji programu

```
Const WSInvoice As String = "Faktury"
```

```
Const WSCopy As String = "Kopia"
```

```
Const M_NoWorksheet As String = "Funkcja nie może być " & _  
    "uruchomiona. Nie znaleziono arkusza " & WSInvoice & "."
```

```
Const M_BadRange As String = "Nieprawidłowy zakres danych."
```

```
Const M_NoDate As String = "Wskazana kolumna nie zawiera dat."
```

```
Const M_BadNumber As String = "Należy wprowadzić liczbę."
```

```
Const M_NoInvoice As String = "Brak płatności w określonym " & _  
    "przedziale czasowym."
```

```
Const M_DelCopy As String = "W arkuszu " & WSCopy & " są już" & _  
    "dane. Kontynuacja operacji spowoduje ich usunięcie. Wykonać?"
```


Brak arkusza zawierającego dane płatności

Zdarzenie formularza Initialize

```
Private Sub UserForm_Initialize()  
    On Error GoTo WSNotFound          ' gdy nie można aktywować arkusza  
    Worksheets(WSInvoice).Activate    ' skok do etykiety WSNotFound  
    On Error GoTo 0                  ' przywrócenie standardowej obsługi błędów  
    ...  
Exit Sub  
WSNotFound:  
    MsgBox M_NoWorksheet, vbCritical  
    Unload UserForm2  
End Sub
```

Uruchomienie formularza

```
Public Sub Faktury2()  
    On Error Resume Next           ' gdy wystąpi błąd zostanie zignorowany  
    UserForm2.Show  
End Sub
```

Nieprawidłowe dane wejściowe

```
Private Sub CommandButton1_Click()  
...  
On Error GoTo BadRange 'jeżeli nieprawidłowy zakres skok do BadRange  
Set dates = Range(ComboBox1.Text)  
  
On Error GoTo NoDate 'jeżeli brak dat w zakresie skok do NoDate  
Set dates = dates.SpecialCells(xlCellTypeConstants, xlNumbers)  
  
On Error GoTo BadNumber 'jeżeli nieprawidłowa liczba skok do BadNumber  
t = Int(TextBox1.Value)  
...  
Exit Sub  
  
BadRange:  
    MsgBox M_BadRange, vbCritical: ComboBox1.SetFocus: Exit Sub  
  
NoDate:  
    MsgBox M_NoDate, vbExclamation: ComboBox1.SetFocus: Exit Sub  
  
BadNumber:  
    MsgBox M_BadNumber, vbExclamation: TextBox1.SetFocus  
  
End Sub
```

Uwaga: CommandButton3_Click analogicznie

Aktualizacja listy kolumn

```
Private DataChanged As Boolean      ' zmienna modułowa sygnalizująca  
                                       ' zmianę zakresu danych (RefEdit1)  
  
Private Sub RefEdit1_Change()        ' zaznaczenie zmiany zakresu, gdy  
    DataChanged = True                ' użytkownik zmodyfikował RefEdit1  
End Sub  
  
Private Sub RefEdit1_Exit(ByVal Cancel As MSForms.ReturnBoolean)  
    If DataChanged Then              ' ponowne wypełnienie listy kolumn  
        FillColumns                    ' gdy kursor opuszcza RefEdit1,  
    End If                             ' a dane uległy zmianie  
End Sub
```

Uwaga: zmiana listy dostępnych kolumn (formant ComboBox1) następuje gdy dane w RefEdit1 uległy zmianie (wystąpiło zdarzenie RefEdit1_Change ustawiając zmienną DataChange) i kursor opuszcza formant RefEdit1 (zdarzenie RefEdit1_Exit).

Usuwanie istniejącej kopii płatności

```
Private Sub CommandButton3_Click()  
Dim ans As Integer  
...  
On Error GoTo NoWorksheet  
With Worksheets(WSCopy).UsedRange  
    If .Rows.Count > 1 Then ' jeżeli w arkuszu są dane pytanie ...  
        ans = MsgBox(M_DelCopy, vbQuestion+vbYesNoCancel+vbDefaultButton2)  
        If ans <> vbYes Then Exit Sub ' ...i usuwanie gdy odpowiedź TAK  
        .Resize(.Rows.Count-1, .Columns.Count).Offset(1, 0).EntireRow.Clear  
    End If  
End With  
...  
End Sub
```

Uwaga: Jeżeli istnieje arkusz zawierający kopię płatności (nazwę określa stała `WSCopy`) i są w nim dane (tzn. zawiera więcej niż jeden wiersz, pierwszy wiersz jest nagłówkiem) program usunie cały zakres (metoda `Clear`). Przed kasowaniem danych użytkownik jest proszony o potwierdzenie wykonania operacji (`MsgBox` typu `vbQuestion`).

Informacja o braku płatności w określonym przedziale

```
Private Sub CommandButton1_Click()  
...  
Dim n As Integer  
...  
n = 0          ' licznik płatności w określonym przedziale  
For Each d In dates  
    If d.Value >= Date And d.Value <= Date + t Then  
        n = n + 1 ' powiększanie licznika gdy płatność została znaleziona  
        If OptionButton1.Value Then d.Font.Color = c _  
        Else ActiveSheet.UsedRange.Rows(d.Row).Font.Color = c  
    End If  
Next  
' informacja gdy brak płatności w określonym przedziale (n = 0)  
If n = 0 Then MsgBox M_NoInvoice, vbInformation  
...  
End Sub
```

CommandButton3_Click analogicznie

Zapisanie i przywrócenie stanu formularza

Zapis i odczyt ustawień

```
SaveSetting(AppName As String, Section As String, Key As String,  
            Setting As String)
```

```
GetSetting(AppName As String, Section As String, Key As String,  
           [Default]) As String
```

UserForm_Terminate – zapis ustawień formularza

```
SaveSetting "Invoice", "Form", "DateCol", ComboBox1.ListIndex
```

```
SaveSetting "Invoice", "Form", "Days", TextBox1.Value
```

```
If OptionButton1.Value Then SaveSetting "Invoice", "Form", "Sel", 1 _
```

```
Else SaveSetting "Invoice", "Form", "Sel", 2
```

UserForm_Initialize() – przywrócenie ustawień formularza

```
ComboBox1.ListIndex = GetSetting("Invoice", "Form", "DateCol", -1)
```

```
TextBox1.Value = GetSetting("Invoice", "Form", "Days", "")
```

```
If GetSetting("Invoice", "Form", "Sel", 1) = 1 Then _
```

```
OptionButton1.Value = True Else OptionButton2.Value = True
```

Uwaga: zapis/przywrócenie elementu wybranego w ComboBox2 analogicznie do ComboBox1.