

4.2.5. Moduł analizy danych

Ta część aplikacji przeznaczona jest do wykonywania różnego rodzaju raportów analitycznych wykorzystujących głównie mechanizm tabel przestawnych (*Pivot Tabel*). Przed utworzeniem tabel przestawnych konieczne będzie wykonanie pewnych prac przygotowawczych obejmujących utworzenie potrzebnych pól wraz z funkcjami wyliczającymi wartości tych pól.

Forma *frmAnaliza* została utworzona w celu udostępnienia użytkownikowi łatwego dostępu do szeregu różnych analiz obejmujących trzy grupy zagadnień: analizę poniesionych wydatków, analizę wpływów oraz saldo wydatków i przychodów.

4.2.4.1. Przygotowanie danych do analizy

Raporty tabel przestawnych będą wykonywane generalnie na dwóch tabelach: *FakturyDetale* i *Sprzedaz*. Pomocniczo będą także wykorzystywane dane zgromadzone w innych tabelach.

W momencie uruchomienia formy *frmAnaliza* w arkuszach przechowujących tabele *FakturyDetale* oraz *Sprzedaz* muszą być utworzone te dodatkowe pola, które mamy zamiar wykorzystywać w raportach tabel przestawnych czy w procedurach wyliczających saldo rozchodów i przychodów. Moim zamiarem jest budowanie tabel przestawnych uwzględniając takie klasyfikacje jak: miesiące, dostawcy (odbiorcy) produktów, produkty i kategorie, tym samym muszę utworzyć odpowiednie pola.

W przypadku tabeli *FakturyDetale* w kolumnach I:N muszą być dynamicznie zbudowane następujące pola (z formułami utworzonymi w komórkach wiersza drugiego).

KwotaVAT – dla wyliczenia zapłaconego podatku VAT z formułą:

=ZAOKR(F2*G2;2),

Brutto – dla wyliczenia faktycznych kosztów zakupu z formułą: =F2+I2,

Kategoria – dla wyznaczenia kategorii zakupionego produktu z formułą:

=WYSZUKAJ.PIONOWO(C2;Produkty!\$A\$2:\$C\$31;2;FAŁSZ),

Data – dla ujednoczenia daty zakupu (w przypadku zakupów fakturowanych data nie była rejestrowana w polu *DataZakupu*) z formułą: =JEŻELI(B2>-1; WYSZUKAJ.PIONOWO(B2;Faktury!\$A\$2:\$D\$61;4;FAŁSZ);H2),

Mc – dla udostępnienia numeru miesiąca dokonania zakupu z formułą wykorzystującą wcześniej ujedniczoną datę: =MIESIĄC(L2),

Dostawca – dla udostępnienia identyfikatora dostawcy na podstawie identyfikatora produktu z formułą: =JEŻELI(B2=-1;-1;WYSZUKAJ.PIONOWO(B2;Faktury!\$A\$2:\$D\$61;2;FAŁSZ))

W tabeli *Sprzedaz* w kolumnach I:L trzeba utworzyć następujące pola:

KwotaVat – dla naliczania należnej kwoty podatku VAT (do odzyskania) z formułą:
=JEŻELI(H2=1;ZAOKR(F2*0,03;2);0),

Brutto – dla wyznaczenia wartości brutto sprzedanych produktów z formułą: =F2+I2,

Kategoria – dla ustalenia kategorii odbiorcy produktów z formułą:
=WYSZUKAJ.PIONOWO(C2;Produkty!\$G\$2:\$I\$31;2;FAŁSZ),

Mc – dla ustalenia miesiąca sprzedaży produktu z formułą: =MIESIĄC(G2).

Standardowo w skoroszycie *NowyRejestrRolnika.xls* pola te nie występują, po prostu nie jest to konieczne. Będziemy je tworzyć w momencie, gdy będą potrzebne, podobnie po wykorzystaniu będziemy je kasować.

Zadanie tworzenia tych pól jest realizowane przez procedurę inicjalizacyjną formy *frmAnaliza*, przy czym przy budowaniu formuł w arkuszu wykorzystywana będzie adresacja względna typu W1C1.

```
Private Sub UserForm_Initialize()
Dim tP As String, tF As String
Dim ileP As Integer, ileF As Integer, myOb As Range
ZabezpieczArkusze False ' odbezpiecza arkusze
Worksheets("Produkty").Activate
ileP = Application.WorksheetFunction.CountA(Range("A:A"))
tP = "Produkty!R2C1:R" & LTrim(Str(ileP)) & "C3"
' ustalenie adresu tabeli Faktury
Worksheets("Faktury").Activate
ileF = Application.WorksheetFunction.CountA(Range("A:A"))
tF = "Faktury!R2C1:R" & LTrim(Str(ileF)) & "C4"
' aktywowanie tabeli detali
Worksheets("FakturyDetale").Activate
Set myOb = Range("I1")
myOb.Offset(0, 0) = "KwotaVat"
myOb.Offset(0, 1) = "Brutto"
myOb.Offset(0, 2) = "Kategoria"
myOb.Offset(0, 3) = "Data"
myOb.Offset(0, 4) = "Mc": myOb.Offset(0, 5) = "Dostawca"
' budowanie formuł
Range("I2").Select
ActiveCell.FormulaR1C1 = "=ROUND(RC[-3]*RC[-2],2)"
Range("J2").Select
ActiveCell.FormulaR1C1 = "=RC[-4]+RC[-1]"
Range("K2").Select
ActiveCell.FormulaR1C1 = "=VLOOKUP(RC[-8]," & tP & _
    ",2,FALSE)"
Range("L2").Select
ActiveCell.FormulaR1C1 = _
```

```

        "=IF(RC[-10]>-1,VLOOKUP(RC[-10]," & tF & _
            ",4,FALSE),RC[-4])"
Range("M2").Select
ActiveCell.FormulaR1C1 = "=MONTH(RC[-1])"
Range("N2").Select
ActiveCell.FormulaR1C1 = _
    "=IF(RC[-12]=-1,-1,VLOOKUP(RC[-12]," & tF & _
        ",2,FALSE))"
' przygotowanie do wykonania operacji FillDown
i = Application.WorksheetFunction.CountA(Range("A:A")) - 1
Set myOb = Range("I2").Resize(i, 6)
Set myProdukty = Range("A1").Resize(i + 1, 14)
myOb.Select
Selection.FillDown
Range("i2").Select
' przygotowanie pól w tabeli Sprzedaz
Worksheets("Produkty").Activate
ileP = Application.WorksheetFunction.CountA(Range("G:G"))
tP = "Produkty!R2C7:R" & LTrim(Str(ileP)) & "C9"
'
Worksheets("Sprzedaz").Activate
Set myOb = Range("I1")
myOb.Offset(0, 0) = "KwotaVat"
myOb.Offset(0, 1) = "Brutto"
myOb.Offset(0, 2) = "Kategoria"
myOb.Offset(0, 3) = "Mc"
'
Range("I2").Select
ActiveCell.FormulaR1C1 = _
    "=IF(RC[-1]=1,ROUND(RC[-3]*0.03,2),0)"
Range("J2").Select
ActiveCell.FormulaR1C1 = "=RC[-4]+RC[-1]"
Range("K2").Select
ActiveCell.FormulaR1C1 = "=VLOOKUP(RC[-8]," & tP & _
    ",2,FALSE)"
Range("L2").Select
ActiveCell.FormulaR1C1 = "=MONTH(RC[-5])"
i = Application.WorksheetFunction.CountA(Range("G:G")) - 1
Set myOb = Range("I2").Resize(i, 4)
myOb.Select
Selection.FillDown
Range("i2").Select
' zmienna mySprzedaz zawiera referencje do tabeli Sprzedaz
Set mySprzedaz = Range("A1").Resize(i + 1, 12)

```

```

' zmienna myMiesiac dostaje referencje do obszaru
' zawierajacego słownie nazwy miesięcy
Set myMiesiac = Range("X2:Y13")
MultiPage1.Value = 0
UstalRozmiar 263, 205, 216
PrzygotujArkuszFakturyDetale ' będzie opisana dalej
End Sub

```

Efekt działania procedury UserForm_Initialize uruchamianej w momencie otwierania formy frmAnaliza pokazany jest poniżej.

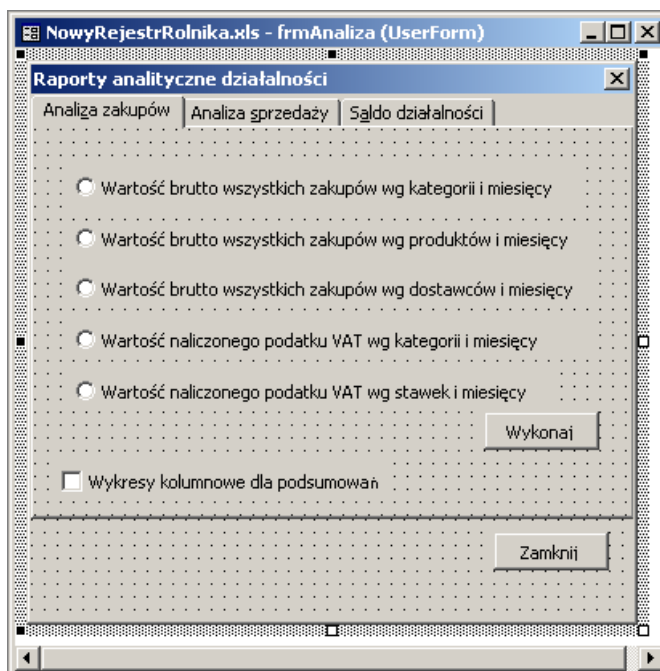
	E	F	G	H	I	J	K	L	M	N
1	ID_Jednosci	Cena	StawkaVat	DataZakupu	KwotaVat	Brutto	Kategorie	Data	Mc	Dostawca
23	17	46,00	22%		10,12	56,12	9	2003-03-15	3	8
24	17	56,00	7%		3,92	59,92	9	2003-03-15	3	8
25	16	156,00	22%		34,32	190,32	13	2003-01-28	1	12
26	16	245,00	7%		17,15	262,15	13	2003-01-28	1	12
27	16	45,00	22%		9,9	54,90	13	2003-02-25	2	12
28	12	480		2003-03-28	0	480,00	14	2003-03-28	3	-1
29	12	600,00		2003-04-28	0	600,00	14	2003-04-28	4	-1
30	5	600,00	22%		132	732,00	3	2003-03-23	3	5

	D	E	F	G	H	I	J	K	L	M
1	ID_jednostki	Ilosc	WartoscN	DataS	Kontraktacja	KwotaVat	Brutto	Kategorie	Mc	
2	3	34	1250	2003-12-04	1	37,5	1287,5		6	12
3	5	23	234	2003-12-04	0	0	234		9	12
4	4	234	12456	2003-12-06	1	373,68	12829,68		13	12
5	4	24	12	2003-12-06	0	0	12		13	12
6	3	23	21000	2003-10-10	1	630	21630		9	10
7	3	55	45000	2003-11-10	1	1350	46350		9	11
8	8	40	55	2003-11-14	0	0	55		9	11
9	3	12	24000	2003-07-14	1	720	24720		10	7
10	3	8	14000	2003-07-15	1	420	14420		10	7
11	3	34	50000	2003-07-17	1	1500	51500		10	7

4.2.4.2. Ogólne procedury formy frmAnaliza

Forma frmAnaliza, podobnie jak frmZakupy i frmMojaSprzedaz została zaprojektowana jako wielozakładkowa. Poszczególne zakładki tej formy mają pozwolić na wykonanie różnych analiz z zasygnalizowanych na początku tego rozdziału trzech grup

zagadnień. Poniżej pokazany jest widok projektu tej formy, jej cechą charakterystyczną jest to, że dominującym typem kontrolki wykorzystanych w tej formie jest `OptionButton`. Kontrolki te uzupełnione są polami wyboru i standardowymi przyciskami. Wspólną kontrolką dla wszystkich zakładek jest przycisk o nazwie `cmdZamknij`.



Procedura `UstalRozmiar` dostosowuje wysokość tej formy, wysokość kontrolki `MultiPage1` oraz położenie przycisku `cmdZamknij` do aktualnie wybranej zakładki.

```
Private Sub UstalRozmiar(w1 As Integer, w2 As Integer, _
    w3 As Integer)
    frmAnaliza.Height = w1
    MultiPage1.Height = w2
    cmdZamknij.Top = w3
End Sub
```

Procedura `MultiPage1_Change` wywoływana jest w momencie wyboru przez użytkownika określonej zakładki, a jej zadaniem jest modyfikacja wymiarów formy oraz uaktywnienie potrzebnego arkusza.

```
Private Sub MultiPage1_Change()
    Application.ScreenUpdating = False
    Select Case MultiPage1.Value
        Case 0 'Zakupy
```

```
        UstalRozmiar 263, 205, 216
        PrzygotujArkuszFakturyDetale
    Case 1 ' Sprzedaz wlasna
        UstalRozmiar 263, 205, 216
        PrzygotujArkuszSprzedaz
    Case 2
        UstalRozmiar 238, 175, 186
    End Select
End Sub
```

Dwie procedury pomocnicze wywoływane przez `MultiPage1_Change`, czyli `PrzygotujArkuszFakturyDetale` oraz `PrzygotujArkuszSprzedaz` mają za zadanie przygotowanie zmiennych obszarowych wykorzystywanych do opisanie wierszy i kolumn wykonanego raportu tabeli przestawnej.

```
Private Sub PrzygotujArkuszFakturyDetale()
    Worksheets("Dostawcy").Activate
    i = Application.WorksheetFunction.CountA(Range("A:A"))
    Set myDostawcy = Range("A2").Resize(i - 1, 2)
    Worksheets("Produkty").Activate
    ileP = Application.WorksheetFunction.CountA(Range("A:A"))
    Set myProdukty = Range("A2").Resize(ileP - 1, 3)
    Worksheets("Kategorie").Activate
    i = Application.WorksheetFunction.CountA(Range("A:A"))
    Set myKategorie = Range("A2").Resize(i - 1, 2)
    Worksheets("FakturyDetale").Activate
End Sub

Private Sub PrzygotujArkuszSprzedaz()
    Worksheets("Dostawcy").Activate
    i = Application.WorksheetFunction.CountA(Range("G:G"))
    Set myOdbiorcy = Range("G2").Resize(i - 1, 2)
    Worksheets("Produkty").Activate
    ileP = Application.WorksheetFunction.CountA(Range("G:G"))
    Set myProdukty = Range("G2").Resize(ileP - 1, 3)
    Worksheets("Kategorie").Activate
    i = Application.WorksheetFunction.CountA(Range("G:G"))
    Set myKategorie = Range("G2").Resize(i - 1, 2)
    Worksheets("Sprzedaz").Activate
End Sub
```

W module formy frmAnaliza utworzyłem kilka ważnych procedur realizujących istotne, dla funkcjonalności tej formy, zadania. Wszystkie wykonane analizy będą zapisywane w nowych arkuszach, które otrzymają nazwę *Raport* z aktualnym numerem. Zadaniem pokazanej niżej procedury jest właśnie nadanie arkuszowi takiej nazwy.

```
Private Sub ZmienNazweArkusza()
    On Error Resume Next
    Dim start As Integer
    Do
        start = start + 1
        Err.Clear
        ActiveSheet.Name = "Raport" & LTrim(Str(start))
    Loop Until Err.Number = 0
End Sub
```

W trakcie pracy z formą frmAnaliza tworzone są kolejne arkusze raportów. Procedura UsunArkusze odpowiada za ustalenie liczby arkuszy tego typu i za ewentualne ich skasowanie przed utworzeniem kolejnego raportu. Procedura wykorzystuje pętlę typu For Each do przeglądania kolekcji arkuszy.

```
Private Sub UsunArkusze()
    Dim IleA As Integer, t As String, myS As Worksheet, Style As Integer
    IleA = 0
    ' ustalenie liczby raportów w kolekcji arkuszy
    For Each myS In Worksheets
        If Len(myS.Name) > 6 Then
            If Left(myS.Name, 6) = "Raport" Then
                IleA = IleA + 1
            End If
        End If
    Next
    ' jeżeli są takie arkusze, to zapytaj co z nimi
    If IleA > 0 Then
        Style = vbInformation + vbYesNo + vbDefaultButton2
        t = "Utworzyłeś już" & Str(IleA) & _
            " arkuszy raportów. Czy mam je skasować ?"
        i = MsgBox(Prompt:=t, Buttons:=Style, _
            Title:="Nowy rejestr")
        If i = vbYes Then
            For Each myS In Worksheets
                If Len(myS.Name) > 6 Then
                    If Left(myS.Name, 6) = "Raport" Then
                        myS.Activate
                    End If
                End If
            Next
        End If
    End If
End Sub
```

```

        myS.Delete
    End If
End If
Next
End If
End If
End Sub

```

Zadaniem stosunkowo dużej i skomplikowanej procedury DajWykresKolumnowy jest zbudowanie eleganckiego wykresu kolumnowego wraz ze wstępnym sformatowaniem.

```

Private Sub DajWykresKolumnowy(myObszar As Range, _
    strGdzie As String, flaga As Boolean, _
    tytulWykresu As String)
    Dim myOsX As Range, jak As Integer, t As String, _
    tytulX As String
    ' ustalenie sposobu wykonania wykresu (gdzie są serie)
    If flaga = True Then
        Set myOsX = Range(myObszar.Cells(1, 2).Address). _
            Resize(1, myObszar.Columns.Count - 1)
        jak = xlRows
    Else
        Set myOsX = Range(myObszar.Cells(2, 1).Address). _
            Resize(myObszar.Rows.Count - 1, 1)
        jak = xlColumns
    End If
    tytulX = myObszar.Cells(1, 1)
    ' utworzenie nowego wykresu
    Charts.Add
    ' okreslenie jego typu
    ActiveChart.ChartType = xlColumnClustered
    ' ustawienie źródła danych i położenia serii
    ActiveChart.SetSourceData Source:=myObszar, PlotBy:=jak
    ' usunięcie z serii pierwszej z nich
    ActiveChart.SeriesCollection(1).Delete
    ' ustawienie etykiet osi X
    ActiveChart.SeriesCollection(1).XValues = myOsX
    ActiveChart.Location Where:=xlLocationAsObject, _
        Name:=strGdzie
    ' zdefiniowanie tytułu i jego atrybutów
    With ActiveChart
        .HasTitle = True
        .ChartTitle.Characters.Text = tytulWykresu
    End With
    ActiveChart.ChartTitle.Select

```

```
Selection.AutoScaleFont = True
With Selection.Font
    .Name = "Arial CE"
    .FontStyle = "Kursywa"
    .Size = 10
End With
' usunięcie legendy
ActiveChart.HasLegend = False
' wyselekcjonowanie i ustalenie formatu dla osi Y
ActiveChart.Axes(xlValue).Select
Selection.TickLabels.NumberFormat = "#,##0"
' ustawienie innych atrybutów tej osi
Selection.TickLabels.AutoScaleFont = True
With Selection.TickLabels.Font
    .Name = "Arial CE"
    .FontStyle = "Normalny"
    .Size = 9
End With
' wyselekcjonowanie osi X i ustawienie jej atrybutów
ActiveChart.Axes(xlCategory).Select
Selection.TickLabels.AutoScaleFont = True
With Selection.TickLabels.Font
    .Name = "Arial CE"
    .FontStyle = "Normalny"
    .Size = 9
End With
' opisanie osi X wraz z ustawieniem atrybutów
With ActiveChart
    .Axes(xlCategory, xlPrimary).HasTitle = True
    .Axes(xlCategory, _
        xlPrimary).AxisTitle.Characters.Text = tytułX
End With
Selection.AutoScaleFont = True
With Selection.Font
    .Name = "Arial CE"
    .FontStyle = "Normalny"
    .Size = 8
End With
' ustalenie nazwy utworzonego wykresu dla potrzeb
' jego przesunięcia
t = ActiveChart.Name
t = Mid(t, InStr(1, t, " ") + 1, _
    Len(t) - InStr(1, t, " "))
' jeden z dwu wykresów zostanie trochę przesunięty
```

```

    If flaga Then
        ActiveSheet.Shapes(t).IncrementLeft 75#
        ActiveSheet.Shapes(t).IncrementTop -37.5
    End If
    myObszar.Cells(1, 1).Select
End Sub

```

Procedura ZbudujTabele realizuje zadanie zbudowania raportu tabeli przestawnej na podstawie przekazanych argumentów wraz z pełnym opisaniem tabeli, jej sformatowaniem i ewentualnym wykonaniem wykresów dla podsumowań tabeli. Z uwagi na zakres wykonywanych prac jest to dość złożona procedura.

```

Private Sub ZbudujTabele(myDane As Range, tW As String, _
    tK As String, tD As String, funkcja As String, _
    wykres As Boolean, _
    myWyszukajK As Range, kolumnaK As Integer, _
    myWyszukajW As Range, kolumnaW As Integer)
Dim ileW As Integer, ileK As Integer, myOb As Range, _
    t As String
ActiveSheet.PivotTableWizard SourceType:=xlDatabase, _
    SourceData:= myDane, TableDestination:="", _
    TableName:="Tabela przestawna"
ActiveSheet.PivotTables("Tabela przestawna"). _
    AddFields RowFields:=tW, ColumnFields:=tK
'
With ActiveSheet.PivotTables("Tabela przestawna"). _
    PivotFields(tD)
    .Orientation = xlDataField
    .Caption = funkcja & ": " & tD
    ' .Function = xlSum jest domyślna
End With
ActiveSheet.PivotTables("Tabela przestawna"). _
    PivotSelect "", xlDataAndLabel
Selection.Copy
Selection.PasteSpecial Paste:=xlValues, _
    Operation:=xlNone, SkipBlanks:= False, _
    Transpose:=False
Range("A1").Select
' zakonczenie kopiowania
Application.CutCopyMode = False
ileW = Application.WorksheetFunction.CountA(Range("A:A"))
ileK = Application.WorksheetFunction.CountA(Range("2:2"))
Range("A2").Offset(0, ileK - 1).Value = "Razem"
Range("A1").Offset(ileW - 1).Value = "Razem"

```

```

Range("B3").Offset(0, 0).Resize(ileW - 2, _
    ileK - 1).Select
Selection.NumberFormat = "#,##0.00"
' przygotowanie danych i wykonanie wykresow
If wykres = True Then
    Set myOb = Union(Range("B2").Resize(1, ileK - 2), _
        Range("B2").Offset(ileW - 2, 0). _
            Resize(1, ileK - 2))
    myOb.Copy Range("B2").Offset(0, ileK + 1)
    Range("B2").Offset(0, ileK) = Range("B1").Value
    Range("B2").Offset(1, ileK) = "Razem"
    Set myOb = Range("B2").Offset(0, ileK). _
        Resize(2, ileK - 1)
    myOb.Cells(1, 1).Select
    t = Range("A1").Value
    DajWykresKolumnowy myOb, ActiveSheet.Name, True, _
        Range("A1").Value
    Set myOb = Union(Range("A2").Resize(ileW - 2, 1), _
        Range("A2").Offset(0, ileK - 1). _
            Resize(ileW - 2, 1))
    Range("A2").Offset(ileW + 1, 0).Value = _
        "Pomocnicze dane dla wykresu"
    myOb.Copy Range("A2").Offset(ileW + 2, 0)
    Set myOb = Range("A2").Offset(ileW + 2, 0). _
        Resize(ileW - 2, 2)
    DajWykresKolumnowy myOb, ActiveSheet.Name, False, _
        Range("A1").Value
    Columns("B:B").Select
    Selection.Columns.AutoFit
End If
' opis wierszy, ale wtedy, gdy kolumnaK > 0
If kolumnaK > 0 Then
    Set myOb = Range("A3").Resize(ileW - 3, 1)
    For i = 1 To myOb.Rows.Count
        If myOb.Cells(i) = -1 Then
            myOb.Cells(i) = "nieustalony"
        Else
            myOb.Cells(i) = _
                Application.WorksheetFunction.VLookup( _
                    myOb.Cells(i), myWyszukajK, kolumnaK, False)
        End If
    Next i
    Columns("A:A").Select
    Selection.Columns.AutoFit

```

```

End if
' opis kolumn, ale wtedy, gdy kolumnaW > 0
If kolumnaW > 0 Then
  Set myOb = Range("B2").Resize(1, ileK - 2)
  For i = 1 To myOb.Columns.Count
    myOb.Cells(i) = Application.WorksheetFunction. _
      VLookup(myOb.Cells(i), myWyszukajW, _
        kolumnaW, False)
  Next i
End If
' dopasowanie rozmiarów
Range("B2").Resize(ileW - 1, ileK - 1).Select
Selection.Columns.AutoFit
Range("A1").Select
End Sub

```

4.2.4.3. Analiza zakupów produktów i usług

Zakładka Page1 kontrolki MultiPage1 zawiera pięć przycisków radiowych o nazwach odpowiednio optP1Analiza1 do optP1Analiza5. Każdy z tych przycisków odpowiada za wykonanie określonej analizy poniesionych wydatków na zakup środków produkcji i usług. Pole wyboru o nazwie chkP1Wykres pozwala na zadysponowanie wykonania wykresów dla podsumowań tabeli przestawnej. Przycisk cmdP1 uruchamia procedurę wywołującą przedstawioną wyżej procedurę ZbudujTabele z odpowiednimi argumentami.

Przy wykonywaniu analiz dla kwoty podatku VAT trzeba będzie rozwiązać dodatkowo problem ograniczenia rekordów tabeli *FakturyDetale* do tych tylko, dla których kwota VAT jest niezerowa. Problem ten został rozwiązany poprzez sortowanie malejące rekordów tej tabeli wg pola *KwotaVAT*, a następnie ograniczenie obszaru przekazywanego jako źródła danych dla tabeli przestawnej do takiego, który obejmie tylko niezerowe rekordy w tym polu. Po przygotowaniu raportu dla tej analizy tabela *FakturyDetale* zostanie ponownie przesortowana w celu przywrócenia sytuacji wyjściowej. Tym razem obszar tabeli będzie sortowany rosnąco wg pola *ID_Pozycji*.

```

Private Sub cmdP1Wykonaj_Click()
  Dim myDaneW As Range, t As String, ileK As Integer
  UsunArkusze
  If optP1Analiza1.Value = True Then
    ZbudujTabele myFakturyDetale, "Kategoria", "Mc", _
      "Brutto", "Suma", chkP1Wykres.Value, _
      myKategorie, 2, myMiesiac, 2
  End If

```

```
If optPlAnaliza2.Value = True Then
    ZbudujTabele myFakturyDetale, "ID_Projektu", "Mc", _
        "Brutto", "Suma", chkPlWykres.Value, _
        myProdukty, 3, myMiesiac, 2
End If
If optPlAnaliza3.Value = True Then
    ZbudujTabele myFakturyDetale, "Dostawca", "Mc", _
        "Brutto", "Suma", chkPlWykres.Value, _
        myDostawcy, 2, myMiesiac, 2
End If
If optPlAnaliza4.Value = True Or _
    optPlAnaliza5.Value = True Then
    ' trzeba sortowac tabele FakturyDetale malejaco wg
    ' polaKwotaVat
    Range("A1").Select
    Selection.Sort Key1:=Range("I2"), _
        Order1:=xlDescending, Header:=xlGuess, _
        OrderCustom:=1, MatchCase:=False, _
        Orientation:=xlTopToBottom
    ' ustalenie o ile rekordów zmniejszyć obszar
    i = Application.WorksheetFunction.CountA( _
        Range("H:H")) - 1
    Set myDaneW = Range("A1").Resize( _
        myFakturyDetale.Rows.Count - i, 14)
End If
If optPlAnaliza4.Value = True Then
    ZbudujTabele myDaneW, "Kategoria", "Mc", _
        "KwotaVat", "Suma", chkPlWykres.Value, _
        myKategorie, 2, myMiesiac, 2
End If
If optPlAnaliza5.Value = True Then
    ZbudujTabele myDaneW, "Mc", "StawkaVat", "KwotaVat", _
        "Suma", chkPlWykres.Value, myMiesiac, 2, _
        myMiesiac, 0
End If
ZmienNazweArkusza
' zapamiętanie zmienionej nazwy arkusza
t = ActiveSheet.Name
' jeżeli trzeba, to jeszcze raz sortuj
If optPlAnaliza4.Value = True Or _
    optPlAnaliza5.Value = True Then
    ' przywracamy wyjściowy porządek
    Worksheets("FakturyDetale").Activate
    Range("A1").Select
```

```

        Selection.Sort Key1:=Range("A2"), Order1:= _
            xlAscending, Header:=xlGuess, OrderCustom:=1, _
            MatchCase:=False, Orientation:=xlTopToBottom
    End If
    PrzygotujZamkniecieFormy
    Worksheets(t).Activate
    Application.ScreenUpdating = True
    End
End Sub

```

Procedura PrzygotujZamkniecieFormy przygotowuje zamknięcie formy frmAnaliza, czyli usuwa zawartość dynamicznych pól (zbudowanych przy otwieraniu formy) oraz zabezpiecza istotne arkusze.

```

Private Sub PrzygotujZamkniecieFormy()
    Worksheets("FakturyDetale").Activate
    Columns("I:N").Select: Selection.ClearContents
    Range("A2").Select: Worksheets("Sprzedaz").Activate
    Columns("I:L").Select: Selection.ClearContents
    Range("A2").Select
End Sub

```

Poniżej widok formy frmAnaliza, zakładka Analiza zakupów aktywna, mamy zamiar przygotować raport tabeli przestawnej pokazujący wartość brutto zakupionych produktów i usług w ujęciu „kategorie x miesiąc”. Pole wyboru wykonania wykresów jest aktywne, można więc uruchomić wykonanie analizy.

Raporty analityczne działalności

Analiza zakupów | Analiza sprzedaży | Saldo działalności

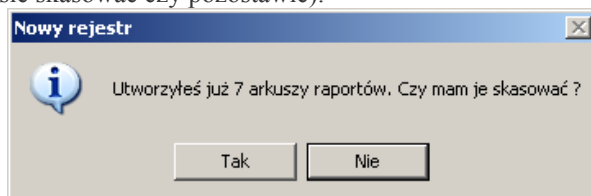
Wartość brutto wszystkich zakupów wg kategorii i miesięcy
 Wartość brutto wszystkich zakupów wg produktów i miesięcy
 Wartość brutto wszystkich zakupów wg dostawców i miesięcy
 Wartość naliczonego podatku VAT wg kategorii i miesięcy
 Wartość naliczonego podatku VAT wg stawek i miesięcy

Wykresy kolumnowe dla podsumowań

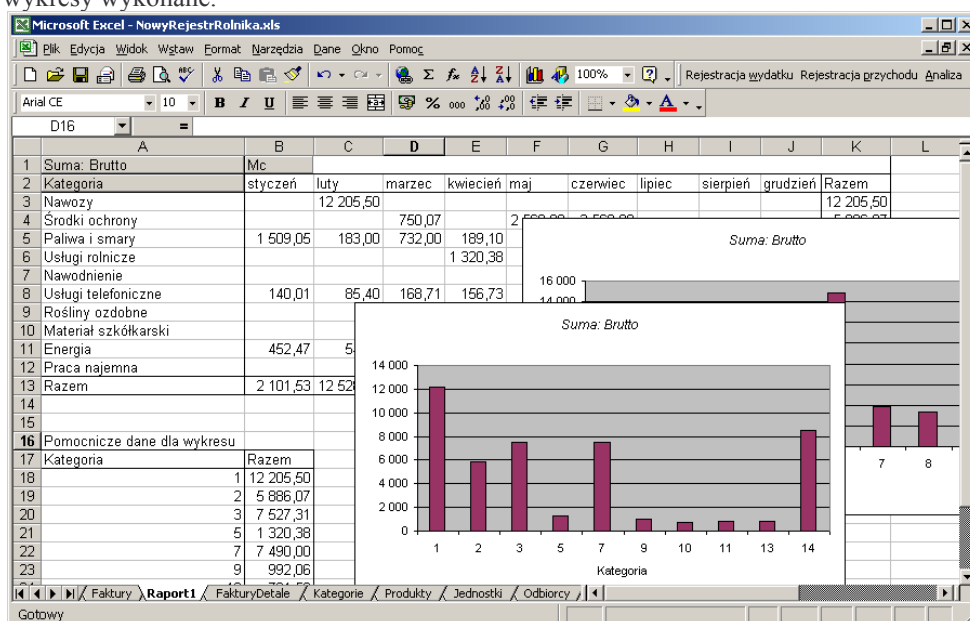
Wykonaj

Zamknij

Procedura cmdP1Wykonaj_Click zaczyna swoją pracę od wywołania procedury UsunArksze, która przeglądając kolekcję arkuszy w bieżącym skoroszycie stwierdza, że raporty były już budowane, wyświetla więc stosowny komunikat z pytaniem, co z tymi arkuszami (w sensie skasować czy pozostawić).

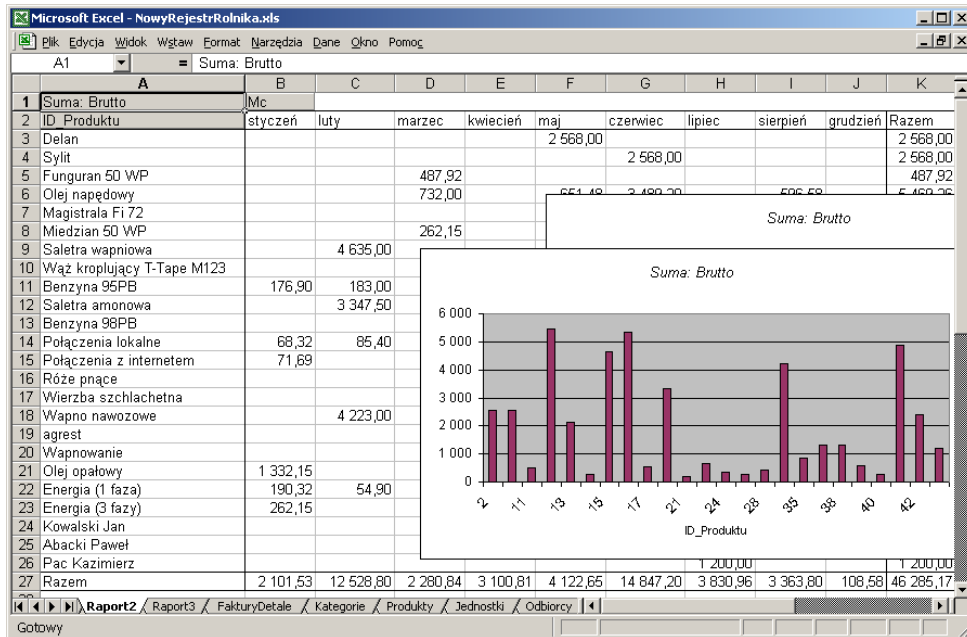


Po wybraniu opcji 'Tak' aplikacja usuwa dotychczasowe raporty i buduje kolejny rozpoczynając jego numerację od 1. Poniżej pokazany jest widok arkusza Excela z wykonanym raportem. Jak widzimy raport jest sformatowany, wiersze i kolumny opisane, wykresy wykonane.

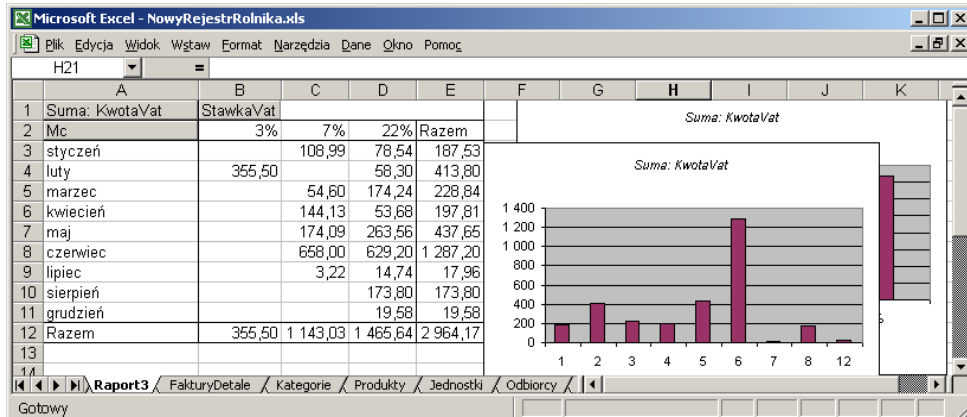


W analogiczny sposób mogą być wykonane analizy wartości brutto kosztów zakupu środków produkcji i usług czy kwot naliczonego podatku VAT w innych, przygotowanych w formie, układach. Wybrane dwie dalsze analizy prezentują na kolejnej stronie.

Analiza kosztów brutto w ujęciu „produkty x miesiące.



Analiza naliczonego podatku VAT w ujęciu „miesiące x stawka VAT”.



4.2.4.4. Analiza sprzedaży własnej

Zakładka Analiza sprzedaży jest w zasadzie obsługiwana przez już przedstawione procedury. Jediną specyficzną procedurą dla tej zakładki jest pokazana niżej procedura cmdP2Wykonaj_Click uruchamiana po wyborze przycisku 'Wykonaj'.

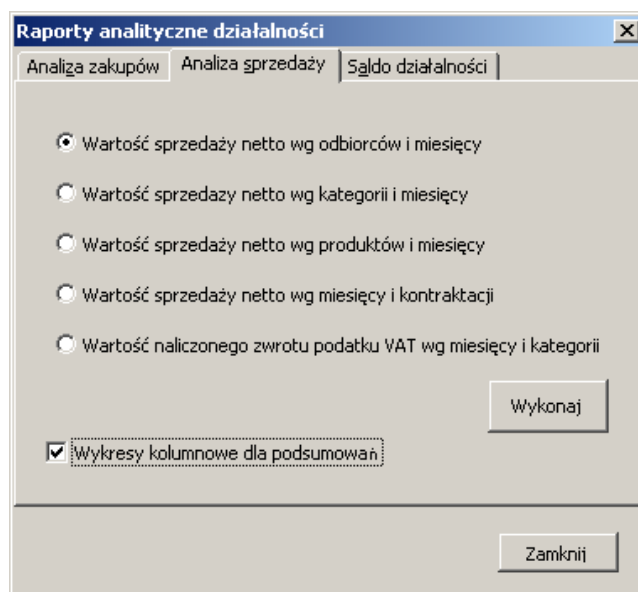
```

Private Sub cmdP2Wykonaj_Click()
Dim myDaneW As Range, t As String
UsunArkusze
If optP2Analiza1.Value = True Then
    ZbudujTabele mySprzedaz, "ID_Odbiorca", "Mc", _
        "WartoscN", "Suma", chkP2Wykres.Value, _
        myOdbiorcy, 2, myMiesiac, 2
End If
If optP2Analiza2.Value = True Then
    ZbudujTabele mySprzedaz, "Kategoria", "Mc", _
        "WartoscN", "Suma", chkP2Wykres.Value, _
        myKategorie, 2, myMiesiac, 2
End If
If optP2Analiza3.Value = True Then
    ZbudujTabele mySprzedaz, "ID_Produkt", "Mc", _
        "WartoscN", "Suma", chkP2Wykres.Value, _
        myProdukty, 3, myMiesiac, 2
End If
If optP2Analiza4.Value = True Then
    ZbudujTabele mySprzedaz, "Mc", "Kontraktacja", _
        "WartoscN", "Suma", chkP2Wykres.Value, _
        myMiesiac, 2, myMiesiac, 0
End If
If optP2Analiza5.Value = True Then
    ' trzeba sortowac malejaca po polu KwotaVat
    Range("A1").Select
    Selection.Sort Key1:=Range("I2"), Order1:= _
        xlDescending, Header:=xlGuess, _
        OrderCustom:=1, MatchCase:=False, _
        Orientation:=xlTopToBottom
    ' pytanie, ile niezerowych rekordów w kolumnie I
    ' tyle, ile jedynek w kolumnie H
    i = Application.WorksheetFunction. _
        CountA(Range("H:H")) - 1
    Set myDaneW = Range("A1").Resize( _
        Application.WorksheetFunction.Sum( _
            Range("H2").Resize(i, 1)) + 1, 12)

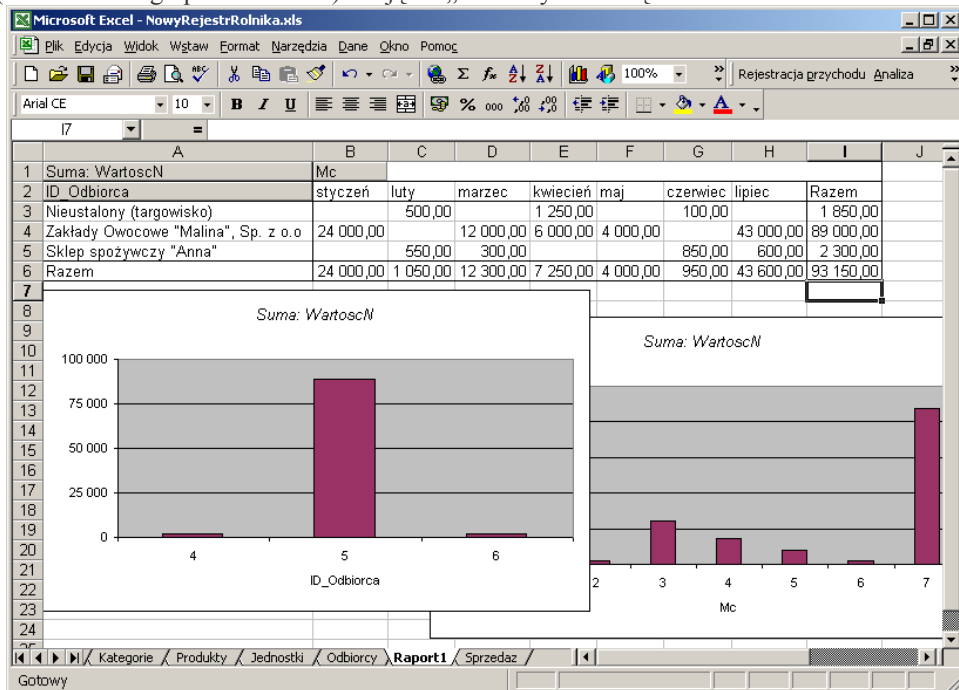
```

```
ZbudujTabele myDaneW, "Mc", "Kategoria", "KwotaVat", _  
    "Suma", chkP2Wykres.Value, myMiesiac, 2, _  
    myKategorie, 2  
End If  
ZmienNazweArkusza  
,  
t = ActiveSheet.Name  
If optP2Analiza5.Value = True Then  
    Worksheets("Sprzedaz").Activate  
    Range("A1").Select  
    Selection.Sort Key1:=Range("A2"), Order1:= _  
        xlAscending, Header:=xlGuess, _  
        OrderCustom:=1, MatchCase:=False, _  
        Orientation:=xlTopToBottom  
End If  
PrzygotujZamkniecieFormy  
Worksheets(t).Activate  
Application.ScreenUpdating = True  
End  
End Sub
```

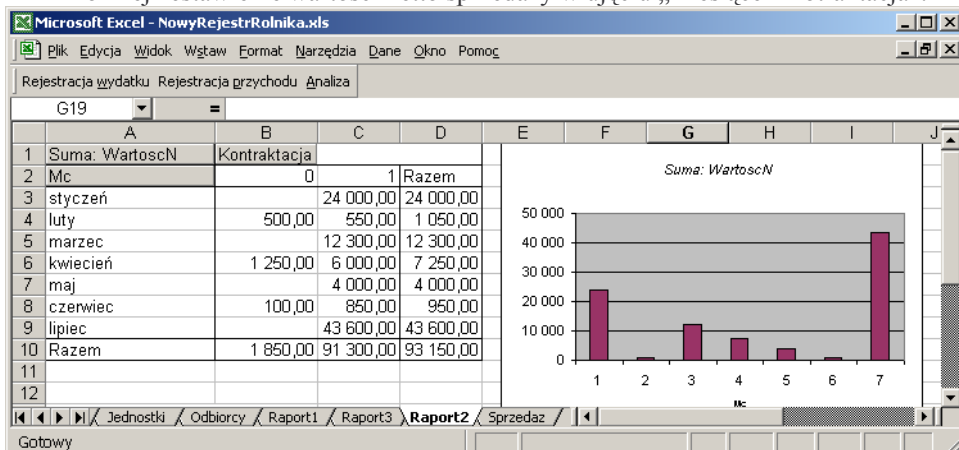
Poniżej pokazane jest okno frmAnaliza, zakładka Analiza sprzedaży jest aktywna, mamy zamiar wykonać pierwszą z dostępnych analiz wraz z przygotowaniem wykresów.



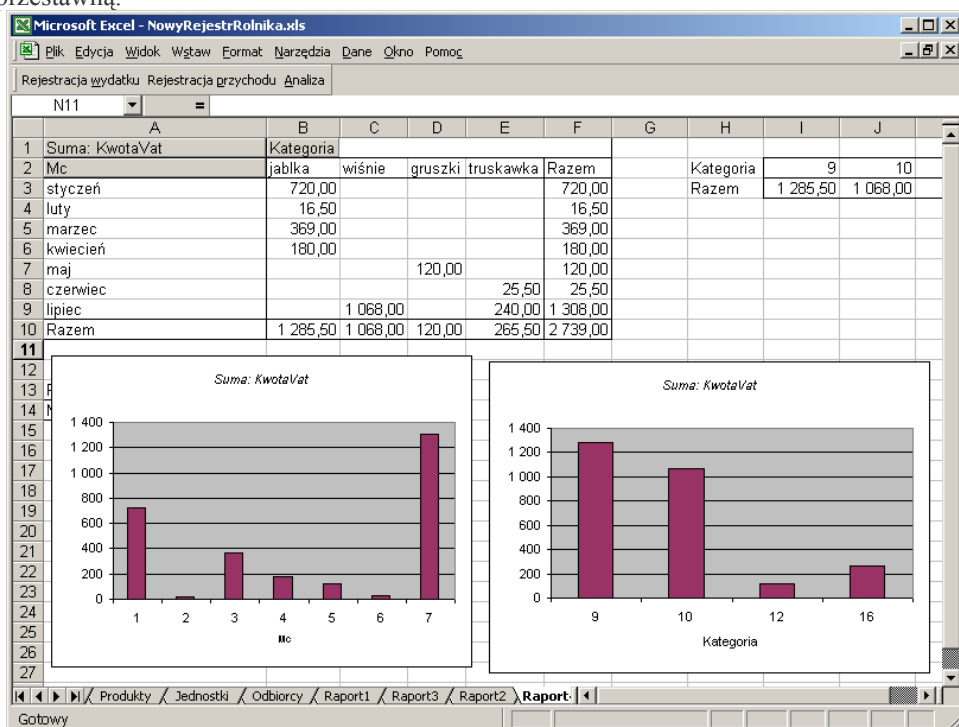
A to już efekt pracy naszej aplikacji. Pokazane jest zestawienie przychodów netto (bez naliczonego podatku VAT) w ujęciu „odbiorcy x miesiąc”.



Poniżej zestawienie wartości netto sprzedaży w ujęciu „miesiące x kontraktacja”.

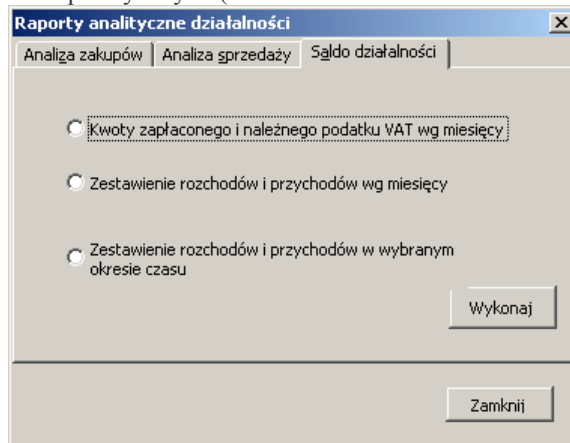


Przy wykonywaniu zestawienia należnego podatku VAT (za sprzedaż kontraktowaną) rozszerzona tabela *Sprzedaz* jest sortowana malejąco wg pola *Kontraktacja* w celu ograniczenia obszaru danych źródłowych przekazywanego do procedury budującej tabelę przestawną.



4.2.4.5. Bilans wydatków i przychodów

Ostatnia zakładka formy `frmAnaliza` pozwala na wykonywanie analiz porównawczych rozchodów i przychodów finansowych w zakresie pozycji brutto oraz kwoty podatku VAT, generalnie w układzie miesięcznym. Ostatnia z proponowanych analiz została zamieszczona bardziej dla chęci pokazania jeszcze jednej, eleganckiej kontrolki, niż z rzeczywistych potrzeb praktycznych (a może i takie uzasadnienie też można znaleźć).



W momencie wybrania zakładki Saldo działalności wykonywana jest specjalna procedura `PrzygotujAnalize`, jej zadaniem jest zestawienie w zmiennej tablicowej X sumarycznych wartości netto i kwot podatku VAT zarówno po stronie wydatków (tabela *FakturyDetale*) jak i przychodów (*Sprzedaz*) w ujęciu miesięcznym. W kolejnym kroku, w zależności od wybranego zestawienia, dane te będą prezentowane w nowym arkuszu w postaci raportu tabeli przestawnej.

```
Private Sub PrzygotujAnalize()
    Dim k As Integer
    Application.ScreenUpdating = False
    Worksheets("FakturyDetale").Activate
    Range("A1").Select
    Selection.Sort Key1:=Range("M2"), Order1:=xlAscending, _
        Header:=xlGuess, OrderCustom:=1, MatchCase:=False, _
        Orientation:=xlTopToBottom
    Worksheets("Sprzedaz").Activate
    Range("A1").Select
    Selection.Sort Key1:=Range("L2"), Order1:=xlAscending, _
        Header:=xlGuess, OrderCustom:=1, MatchCase:=False, _
        Orientation:=xlTopToBottom
```

```

For i = 2 To myFakturyDetale.Rows.Count
    k = myFakturyDetale.Cells(i, 13)
    X(k, 1) = X(k, 1) + myFakturyDetale.Cells(i, 6)
    X(k, 2) = X(k, 2) + myFakturyDetale.Cells(i, 9)
Next i
For i = 1 To 12
    X(i, 3) = X(i, 1) + X(i, 2)
    If X(i, 0) = 0 Then X(i, 0) = 0 Else X(i, 0) = 1
Next i
'
For i = 2 To mySprzedaz.Rows.Count
    k = mySprzedaz.Cells(i, 12)
    X(k, 4) = X(k, 4) + myFakturyDetale.Cells(i, 6)
    X(k, 5) = X(k, 5) + myFakturyDetale.Cells(i, 9)
Next i
For i = 1 To 12
    X(i, 6) = X(i, 4) + X(i, 5)
    If X(i, 0) = 0 And X(i, 6) > 0 Then X(i, 0) = 1
Next i
End Sub

```

Pierwsze dwie analizy są bardzo podobne, jedyna różnica między nimi polega na wykorzystaniu innego pola umieszczonego w obszarze 'dane' tabeli przestawnej. W pierwszym przypadku jest to naliczona kwota podatku VAT, w drugim wartość brutto wydatków i wpływów. Z tego powodu warto przygotować jedną, wspólną procedurę wykonującą taką analizę. Pokazana poniżej procedura AnalizaSalda wykonuje taką pracę, dodatkowo, w zależności od zapotrzebowania wyrażonego argumentem flaga wykonuje elegancki wykres przestrzenny porównując zapłacone i należne kwoty podatku VAT (lub wydatki i wpływy brutto) w kolejnych miesiącach

```

Private Sub AnalizaSalda(t1 As String, kol As Integer, _
    t2 As String, t3 As String, flaga As Boolean)
    Dim myOb As Range, k As Integer, ileW As Integer, _
        t As String
    Dim X1 As Currency, X2 As Currency
    Set myOb = Range("A1")
    myOb.Offset(0, 0) = t1 ' tytuł zestawienia
    myOb.Offset(2, 0) = "Miesiąc"
    myOb.Offset(2, 1) = t2 ' opis pierwszej kolumny
    myOb.Offset(2, 2) = t3 ' opis drugiej kolumny
    myOb.Offset(2, 3) = "Różnica"
    k = 3
    For i = 1 To 12
        If X(i, 0) = 1 Then
            myOb.Offset(k, 0) = _

```

```

        Application.WorksheetFunction. _
            VLookup(i, myMiesiac, 2, False)
        ileW = ileW + X(i, 0)
        myOb.Offset(k, 1) = X(i, kol)
        myOb.Offset(k, 2) = X(i, kol + 3)
        myOb.Offset(k, 3) = X(i, kol + 3) - X(i, kol)
        X1 = X1 + X(i, kol)
        X2 = X2 + X(i, kol + 3)
        k = k + 1
    End If
Next i
myOb.Offset(k, 0) = "Razem"
myOb.Offset(k, 1) = X1
myOb.Offset(k, 2) = X2
myOb.Offset(k, 3) = X2 - X1

Range("B3").Resize(k, 3).Select
Selection.NumberFormat = " # #00.00"
Selection.Columns.AutoFit
'
t = ActiveSheet.Name
Worksheets(t).Activate
ZmienNazweArkusza
t = ActiveSheet.Name
Set myOb = Range("A4").Resize(ileW, 3)
myOb.Select

Worksheets(t).Activate
If flaga = True Then
    Charts.Add
    ActiveChart.ApplyCustomType _
        ChartType:=xl3DColumnClustered
    ActiveChart.SetSourceData Source:=myOb, _
        PlotBy:=xlColumns
    ActiveChart.SeriesCollection(1).Name = _
        """"Wydatki""""
    ActiveChart.SeriesCollection(2).Name = _
        """"Przychody""""
    ActiveChart.Location Where:=xlLocationAsObject, _
        Name:=t
' formatowanie osi
ActiveChart.Axes(xlCategory).Select
Selection.TickLabels.AutoScaleFont = True
Selection.TickLabels.Font.Size = 9

```

```

    '
    ActiveChart.Legend.Select
    Selection.AutoScaleFont = True
    Selection.Font.Size = 9
    '
    ActiveChart.Axes(xlValue).Select
    Selection.TickLabels.AutoScaleFont = True
    Selection.TickLabels.Font.Size = 9
    '
    Selection.TickLabels.NumberFormat = "#,##0"
End If
End Sub

```

Za wykonanie wybranej analizy odpowiada procedura cmdP3Wykonaj_Click, która przy wyborze pierwszych dwóch analiz wywołuje procedurę AnalizaSalda z odpowiednimi parametrami.

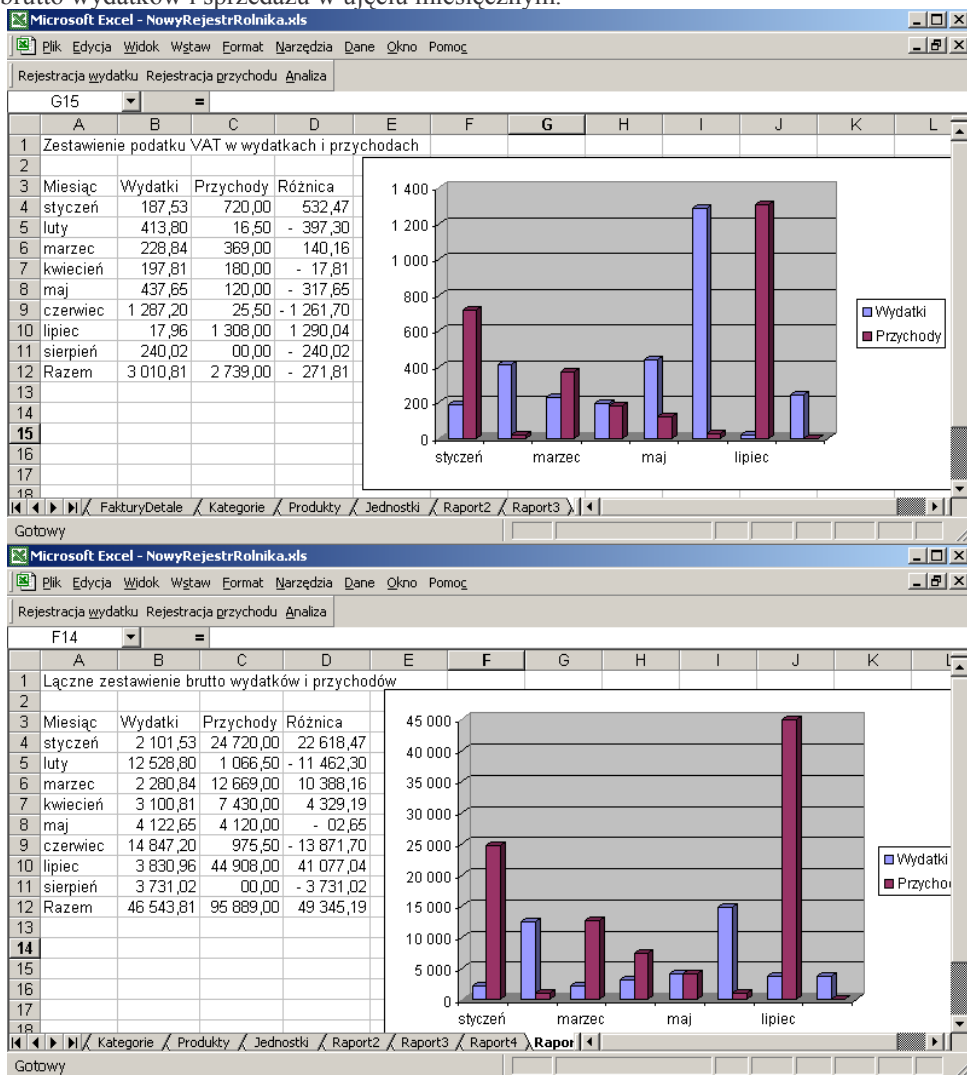
```

Private Sub cmdP3Wykonaj_Click()
    Dim myOb As Range, k As Integer, ileW As Integer
    Dim X1 As Currency, X2 As Currency, X3 As Currency
    UsunArkusze
    ' dodanie nowego arkusza do skoroszytu
    ActiveWorkbook.Sheets.Add
    If optP3Analiza1.Value = True Then
        AnalizaSalda "Zestawienie podatku VAT w wydatkach i
        przychodach", 2, "Wydatki", _
        "Przychody", chkP3Wykres
    End If
    If optP3Analiza2.Value = True Then
        AnalizaSalda "Łączne zestawienie brutto wydatków i
        przychodów", 3, "Wydatki", _
        "Przychody", chkP3Wykres
    End If
    ' jeżeli wybrana analiza między dwoma punktami czasowymi
    If optP3Analiza3.Value = True Then
        ' przeniesienie informacji o ewentualnym wykresie
        If chkP3Wykres.Value = True Then i = 1 Else i = 0
        Load frmUstalDate
        frmUstalDate.Show
    End If
    ' po zakończeniu pierwszych dwóch analiz sterowanie
    ' przekazywane jest w to miejsce, trzeba przygotować
    ' zakończenie pracy formy frmAnaliza
    Range("A1").Select
    Application.ScreenUpdating = True
    ' zamykamy formę

```

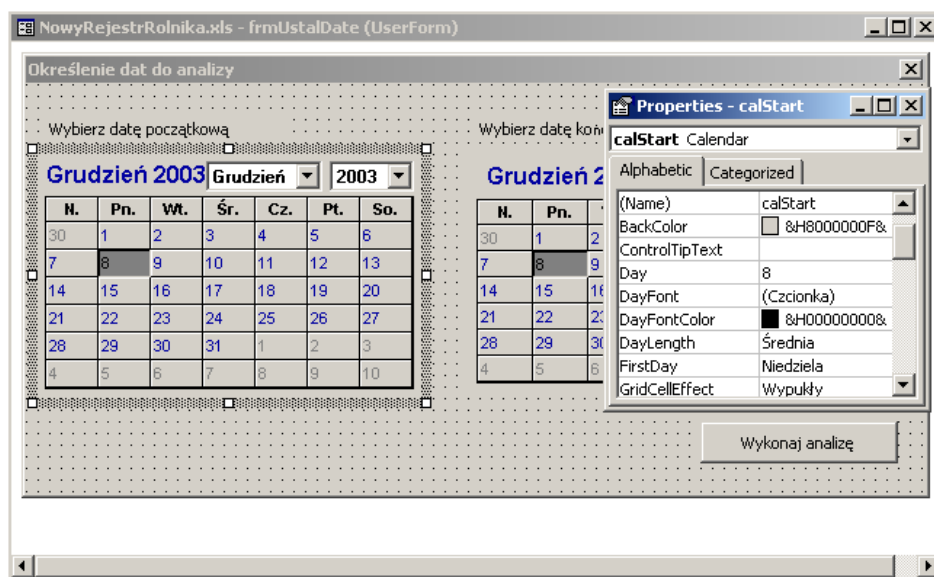
End
End Sub

Poniżej przykłady wykonania dwóch pierwszych analiz podatku VAT i wartość brutto wydatków i sprzedaży w ujęciu miesięcznym.



Przy wyborze trzeciej analizy procedura cmdP3Wykonaj_Click wywołuje dodatkową formę o nazwie frmUstalDate w celu ustalenia daty początkowej i końcowej analizy oraz jej wykonania.

Poniżej widok okna projektu formy frmUstalDate, na jej powierzchni umieściłem dwie kontrolki typu Calendar (rozszerzone, trzeba je doinstalować) o nazwach calStart oraz calKoniec plus przycisk polecenia cmdWykonaj. Listę kontroltek uzupełniają dwie etykiety opisujące oba kalendarze.



Szereg istotnych dla nas właściwości kontroltek typu Calendar nie może być zmodyfikowanych na etapie projektowania formy, będziemy musieli ustawiać te właściwości w trakcie wykonywania formy (dokładniej w momencie jej pokazania na ekranie). Do tej grupy właściwości należą np. wszystkie czcionki wykorzystywane do opisywania elementów kalendarza.

Pokazana niżej procedura UserForm_Initialize ustawia rozmiar czcionki tytułowej oraz początkową datę kontrolki calStart.

```
Private Sub UserForm_Initialize()
    If i = 1 Then fWykres = True Else fWykres = False
    calStart.TitleFont.Size = 10
    calKoniec.TitleFont.Size = 10
    ' ustawienie daty 1 styczeń na pierwszym kalendarzu
    calStart.Month = 1
    calStart.Day = 1
End Sub
```

Instrukcja:

```
If i = 1 Then fWykres = True Else fWykres = False
```

“odbiera” od zmiennej i informację o tym, czy zadysponowano wykonanie wykresu i przypisuje ją do zmiennej logicznej `fWykres` zadeklarowanej w sekcji deklaracji tej formy.

Poniżej widok tej formy w trakcie ustalania zakresu dat do wykonania analizy ...

Określenie dat do analizy

Wybierz datę początkową

Wybierz datę końcową

Luty 2003 Luty 2003

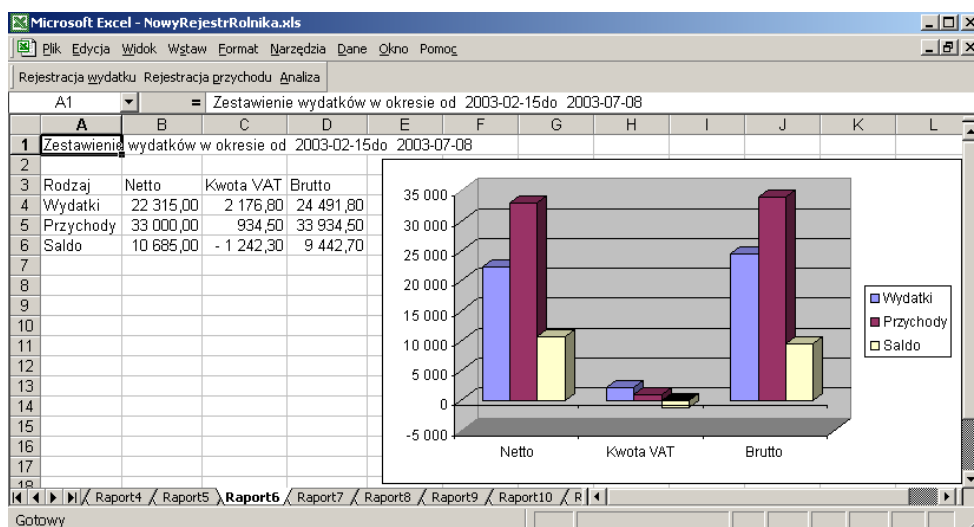
N.	Pn.	Wt.	Śr.	Cz.	Pt.	So.
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	1
2	3	4	5	6	7	8

Lipiec 2003 Lipiec 2003

N.	Pn.	Wt.	Śr.	Cz.	Pt.	So.
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

Wykonaj analizę

i sama analiza.



Analiza została wykonana przez procedurę `cmdWykonaj_Click` pokazaną niżej. Procedura zaczyna swoją pracę od odebrania od kontrolki `calStart` i `calKoniec` początkowej i końcowej daty analizy. Kolejny etap pracy polega na przejrzaniu tabel

FakturyDetale oraz *Sprzedaz* w celu wyłapania tych rekordów, dla których daty transakcji mieszczą się w zadanym przedziale czasowym. Na tych rekordach wykonywane są odpowiednie działania sumujące, a ich rezultat zwracany do dodanego wcześniej arkusza. W zależności od dyspozycji pochodzącej z formy *frmAnaliza* wykonywany jest lub nie wykres bilansowy.

```

Private Sub cmdWykonaj_Click()
Dim X(2, 3) As Currency, myOb As Range
Dim datStart As Date, datKoniec As Date
datStart = calStart.Value
datKoniec = calKoniec.Value
' sprawdzenie poprawności dat
If datStart < datKoniec Then
For i = 2 To myFakturyDetale.Rows.Count
If myFakturyDetale.Cells(i, 12) >= datStart And _
myFakturyDetale.Cells(i, 12) <= datKoniec Then
' netto
X(1, 1) = X(1, 1) + myFakturyDetale.Cells(i, 6)
' kwota Vat
X(1, 2) = X(1, 2) + myFakturyDetale.Cells(i, 9)
' brutto
X(1, 3) = X(1, 3) + myFakturyDetale.Cells(i, 10)
End If
Next i
'
For i = 2 To mySprzedaz.Rows.Count
If mySprzedaz.Cells(i, 7) >= datStart And _
mySprzedaz.Cells(i, 7) <= datKoniec Then
X(2, 1) = X(2, 1) + mySprzedaz.Cells(i, 6)
X(2, 2) = X(2, 2) + mySprzedaz.Cells(i, 9)
X(2, 3) = X(2, 3) + mySprzedaz.Cells(i, 10)
End If
Next i
' wyprowadzanie wyników analizy
Set myOb = Range("A1")
myOb.Offset(0, 0) = _
"Zestawienie wydatków w okresie od " & _
Str(datStart) & "do " & Str(datKoniec)
myOb.Offset(2, 0) = "Rodzaj"
myOb.Offset(2, 1) = "Netto"
myOb.Offset(2, 2) = "Kwota VAT"
myOb.Offset(2, 3) = "Brutto"
myOb.Offset(3, 0) = "Wydatki"
myOb.Offset(3, 1) = X(1, 1)

```

```
myOb.Offset(3, 2) = X(1, 2)
myOb.Offset(3, 3) = X(1, 3)
'
myOb.Offset(4, 0) = "Przychody"
myOb.Offset(4, 1) = X(2, 1)
myOb.Offset(4, 2) = X(2, 2)
myOb.Offset(4, 3) = X(2, 3)
'
myOb.Offset(5, 0) = "Saldo"
myOb.Offset(5, 1) = X(2, 1) - X(1, 1)
myOb.Offset(5, 2) = X(2, 2) - X(1, 2)
myOb.Offset(5, 3) = X(2, 3) - X(1, 3)
' formatowanie obszaru wyników
Range("B4:D6").Select
Selection.NumberFormat = " # #00.00"
Range("A3:d6").Select
Selection.Columns.AutoFit
'
't = ActiveSheet.Name
Worksheets(ActiveSheet.Name).Activate
ZmienNazweArkusza
t = ActiveSheet.Name
' wykonanie wykresu, jezeli trzeba
If fWykres = True Then
    Set myOb = Range("A3:D6")
    Charts.Add
    ActiveChart.ApplyCustomType _
        ChartType:=xl3DColumnClustered
    ActiveChart.SetSourceData Source:=myOb, _
        PlotBy:=xlRows
    ActiveChart.Location Where:=xlLocationAsObject, _
        Name:=t
    ' formatowanie osi i legendy
    ActiveChart.Axes(xlCategory).Select
    Selection.TickLabels.AutoScaleFont = True
    Selection.TickLabels.Font.Size = 9
    ActiveChart.Legend.Select
    Selection.AutoScaleFont = True
    Selection.Font.Size = 9
    ActiveChart.Axes(xlValue).Select
    Selection.TickLabels.AutoScaleFont = True
    Selection.TickLabels.Font.Size = 9
    Selection.TickLabels.NumberFormat = "#,##0"
    ActiveWindow.Visible = False
```

```
End If
,
Range("A1").Select
PrzygotujZamkniecieFormy
Worksheets(t).Activate
End
Else
MsgBox "Data początkowa musi być mniejsza od
końcowej!", , "Nowy rejestr"
End If
End Sub
```

W pokazanej wyżej procedurze miałem zamiar wykorzystać pomocniczą procedurę o nazwie `PrzygotujZamkniecieFormy`, która pierwotnie została utworzona w module formy `frmAnaliza`. Zamiar jej wykorzystania w module `frmUstalDate` **wymagał** przeniesienia procedury `PrzygotujZamkniecieFormy` **do modułu standardowego** wraz ze zmianą typu z prywatnego na publiczny.

Z tych samych powodów również procedura `ZmienNazweArkusza` musiała być przeniesiona do modułu standardowego.