

IDŹ DO

PRZYKŁADOWY ROZDZIAŁ



SPIS TREŚCI

KATALOG KSIĄŻEK

KATALOG ONLINE

ZAMÓW DRUKOWANY KATALOG

TWÓJ KOSZYK

DODAJ DO KOSZYKA

CENNIK I INFORMACJE

ZAMÓW INFORMACJE
O NOWOŚCIACH

ZAMÓW CENNIK

CZYTELNIA

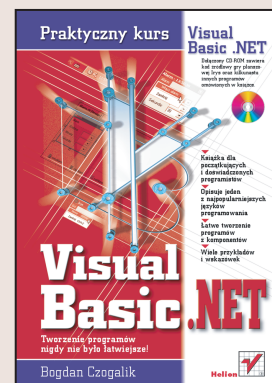
FRAGMENTY KSIĄŻEK ONLINE

Praktyczny kurs Visual Basic .NET

Autor: Bogdan Czogalik

ISBN: 83-7361-386-2

Format: B5, stron: 408



Visual Basic .NET – jeden z najbardziej uniwersalnych i popularnych języków programowania – służy do tworzenia wszelkiego rodzaju programów: edytorów tekstu i grafiki, aplikacji internetowych działających w przeglądarkach, baz danych, gier i wielu innych. Utworzone w języku Visual Basic .NET programy można uruchamiać nie tylko w tradycyjnych komputerach stacjonarnych i przenośnych, ale także w palmtopach.

Wielką zaletą języka Visual Basic .NET jest to, że jest składnikiem pakietu programistycznego Visual Studio .NET, dzięki któremu programista ma do dyspozycji bogaty zestaw narzędzi ułatwiających tworzenie programów. Niniejsza książka – przeznaczona zarówno dla początkujących, jak i doświadczonych programistów – wyjaśnia krok po kroku, jak tworzyć programy przy użyciu narzędzi języka Visual Basic .NET.

Książka omawia m.in.:

- Około 30 formantów, czyli komponentów, z których tworzy się programy
- Klasy, struktury oraz interfejsy
- Składowe klasy i struktur: pola (zmiennie i stałe), tablice, metody i właściwości
- Standardowe elementy okien programów, takie jak menu i paski narzędzi
- Standard graficzny GDI+
- Dostęp do folderów i plików

Dołączony CD-ROM zawiera kod źródłowy gry planszowej Irys oraz kilkunastu innych programów omówionych w książce.

Tworzenie programów nigdy nie było łatwiejsze.

- Książka dla początkujących i doświadczonych programistów
- Opisuje jeden z najpopularniejszych języków programowania
- Łatwe tworzenie programów z komponentów
- Wiele przykładów i wskazówek



Spis treści

Wprowadzenie	9
Rozdział 1. Kalendarz — Twój pierwszy program.....	15
Tworzenie pakietu.....	15
Tworzenie projektu	16
Formularze	17
Dodanie formularza do projektu	17
Zmiana właściwości formularza	18
Formanty	20
Dodanie formantu do formularza.....	21
Zmiana właściwości formantu	22
Kod programu Kalendarz	24
Program Kalendarz w działaniu	26
Rozdział 2. Visual Basic .NET od podszewki	29
Metody	31
Tworzenie metody, która nie zwraca wartości i nie ma zestawu parametrów.....	35
Tworzenie metody, która nie zwraca wartości i ma zestaw parametrów	35
Tworzenie metody, która zwraca wartość i nie ma zestawu parametrów.....	36
Tworzenie metody, która zwraca wartość i ma zestaw parametrów	37
Tworzenie metody, która ma więcej niż jedną wersję	37
Zmienne.....	39
Dostęp do składowych klas.....	43
Składowa instancyjna.....	43
Składowa współdzielona.....	45
Właściwości	46
Tworzenie właściwości do zapisu i do odczytu	48
Tworzenie właściwości tylko do zapisu	49
Tworzenie właściwości tylko do odczytu	50
Kontrola przepływu programu	51
Instrukcja For... Next	51
Instrukcja Do... Loop	53
Instrukcja While... End While.....	56
Instrukcja If... Then... Else.....	57
Instrukcja Select Case.....	58
Instrukcja Try...Catch...Finally	60
Instrukcja Exit.....	61
Instrukcja Return.....	65
Stałe.....	65
Enumeracje.....	67

Tablice	69
Tablica jednowymiarowa	70
Metody i właściwości klasy Array	75
Metoda Clear	76
Metoda Copy	77
Metoda IndexOf	81
Metoda LastIndexOf	82
Metoda Reverse	84
Metoda Sort	87
Metoda GetLowerBound oraz metoda GetUpperBound	90
Metoda GetLength	91
Właściwość Length	92
Właściwość Rank	93
Tablice a metody	95
Tworzenie metody, która zwraca tablicę wartości i nie ma zestawu parametrów	95
Tworzenie metody, która zwraca tablicę wartości i ma zestaw parametrów	96
Tworzenie metody z parametrem, do którego można przekazać dowolną liczbę wartości	98
Tablica wielowymiarowa	99
Tablica tablic	105
Metoda rekurencyjna	109
Metoda służąca do tworzenia komunikatów	110
Metoda Main	114
Tworzenie metody Main	115
Program z metodą Main w działaniu	118
Dwa sposoby przekazywania wartości do parametrów	120
Klasy	122
Klasa bazowa a klasa potomna	123
Klasa finalna	138
Konstruktor klasy	138
Formularz i formant jako instancje klasy	141
Interfejsy	145
Struktury	152
Zbiory komponentów	155
Przegląd operatorów	157
Zestaw operatorów do wykonywania działań arytmetycznych	157
Zestaw operatorów do przypisywania wartości do zmiennej lub właściwości	158
Zestaw operatorów do porównywania wyrażeń	159
Zestaw operatorów do łączenia łańcuchów znaków	160
Zestaw operatorów do wykonywania operacji logicznych	160
Zestaw dodatkowych operatorów	162
Regiony w plikach z kodem źródłowym	162
Rozdział 3. Tworzenie programów sterowanych zdarzeniami	163
Tworzenie programu Zdarzenia	163
Formularz programu Zdarzenia	163
Kod potrzebny do uruchomienia programu Zdarzenia	164
Formant A	165
Formant B	166
Formant C	168
Kod potrzebny do uruchomienia metod UtwórzFormantB oraz UtwórzFormantC tworzących formanty	170
Program Zdarzenia w działaniu	170
Sygnatury metod dołączanych do zdarzeń formularzy i formantów	171

Rozdział 4. Matematyka	173
Metoda Abs	175
Metoda Sign	175
Metoda Floor	176
Metoda Ceiling	177
Metoda Min	178
Metoda Max	179
Metoda Pow	179
Metoda Sqrt	180
Metoda Round	181
Metoda Log10	182
Metoda Log	183
Metoda Sin	184
Metoda Cos	185
Metoda Tan	186
Metoda Sinh	186
Metoda Cosh	187
Metoda Tanh	188
Stała PI	189
Stała E	189
Rozdział 5. Tekst	191
Metoda Compare	192
Metoda Equals	193
Metoda Concat	194
Metoda Format	196
Metoda Join	198
Pole Empty	200
Metoda Split	201
Metoda ToLower	202
Metoda ToUpper	203
Właściwość Length	204
Właściwość Chars	205
Metoda Substring	206
Metoda Insert	207
Metoda Replace	208
Metoda Remove	209
Lokalizacja programów	209
Formularz programu Plus	210
Formanty programu Plus	210
Pliki z materiałami pomocniczymi programu Plus	211
Kod programu Plus	213
Program Plus w działaniu	215
Identyfikatory języków	217
Nazwy plików RESX	218
Rozdział 6. Daty i godziny	219
Właściwość Now	220
Właściwość UtcNow	221
Właściwość Today	222
Metoda Parse	223
Metoda DaysInMonth	223
Metoda IsLeapYear	224
Właściwość DayOfWeek	225

Właściwość DayOfYear	226
Metoda New (konstruktor klasy Date)	227
Konwersja dat i godzin na łańcuchy znaków	229
Metoda ToLongDateString	230
Metoda ToShortDateString	230
Metoda ToLongTimeString	230
Metoda ToShortTimeString	231
Właściwość Day	231
Właściwość Month	232
Właściwość Year	233
Właściwość Hour	233
Właściwość Minute	234
Właściwość Second	235
Metoda AddDays	236
Metoda AddMonths	237
Metoda AddYears	238
Metoda AddHours	239
Metoda AddMinutes	240
Metoda AddSeconds	241
Rozdział 7. Dostęp do folderów i plików	243
Klasa Directory	245
Metoda GetLogicalDrives	245
Metoda GetDirectories	246
Metoda GetFiles	247
Metoda CreateDirectory	250
Metoda Exists	251
Metoda Delete	252
Metoda Move	253
Metoda GetCreationTime	254
Metoda GetLastAccessTime	255
Metoda GetLastWriteTime	255
Metoda SetCreationTime	256
Metoda SetLastAccessTime	257
Metoda SetLastWriteTime	258
Klasa File	260
Metoda Exists	260
Metoda Delete	261
Metoda Move	261
Metoda Copy	262
Metoda GetCreationTime	264
Metoda GetLastAccessTime	265
Metoda GetLastWriteTime	265
Metoda SetCreationTime	266
Metoda SetLastAccessTime	267
Metoda SetLastWriteTime	268
Zapisywanie danych do plików i odczytywanie danych z plików	270
Klasa FileStream	270
Klasa StreamWriter	271
Klasa StreamReader	273
Klasa BinaryWriter	274
Klasa BinaryReader	276

Rozdział 8. Kolory i grafika	279
Kolory.....	279
Struktura Color.....	279
Tworzenie programu Kolory.....	280
Program Kolory w działaniu.....	286
Grafika.....	286
Struktura Point.....	288
Struktura Rectangle.....	288
Klasa Pens.....	289
Klasa Brushes.....	289
Klasa Font.....	289
Klasa Icon.....	290
Metoda DrawRectangle.....	290
Metoda FillRectangle.....	291
Metoda DrawEllipse.....	292
Metoda FillEllipse.....	294
Metoda DrawPie.....	295
Metoda FillPie.....	296
Metoda DrawArc.....	297
Metoda DrawLine.....	298
Metoda DrawCurve.....	299
Metoda DrawClosedCurve.....	301
Metoda FillClosedCurve.....	302
Metoda DrawPolygon.....	303
Metoda FillPolygon.....	305
Klasa SolidBrush.....	306
Klasa HatchBrush.....	307
Klasa TextureBrush.....	309
Klasa LinearGradientBrush.....	311
Klasa PathGradientBrush.....	312
Klasa Pen.....	314
Metoda DrawString.....	317
Metoda DrawImage.....	318
Metoda DrawIcon.....	320
Rozdział 9. Standardowe elementy okien programów	323
Formularz A i jego formanty.....	323
Formant TextBox.....	324
Formant ListBox.....	324
Formant ComboBox.....	324
Formant DomainUpDown.....	324
Formant NumericUpDown.....	325
Formant DateTimePicker.....	325
Formant GroupBox.....	325
Formant RadioButton.....	326
Formant CheckBox.....	326
Formant Label.....	326
Formant Button.....	327
Kolejność klawisza Tab.....	328
Kod formularza A.....	328
Formularz B i jego formanty.....	334
Formant ToolTip.....	334
Formant MainMenu.....	334
Formant ContextMenu.....	337

Formant ImageList.....	339
Formant ToolBar.....	339
Formant StatusBar.....	340
Formant ProgressBar.....	340
Formant Button.....	342
Kod formularza B.....	343
Formularz C i jego formanty.....	348
Formant TreeView.....	348
Formant ListView.....	349
Formant Splitter.....	349
Formant ImageList.....	349
Kod formularza C.....	350
Formanty do tworzenia standardowych elementów okien programów w działaniu.....	359
Rozdział 10. Tworzenie programów z formantów.....	363
Tworzenie programu Zegar.....	363
Formularz programu Zegar.....	363
Formanty programu Zegar.....	364
Kod programu Zegar.....	365
Program Zegar w działaniu.....	366
Tworzenie programu Czas.....	366
Formularz programu Czas.....	367
Formanty programu Czas.....	367
Kod programu Czas.....	368
Program Czas w działaniu.....	370
Tworzenie programu Alarm.....	370
Formularz programu Alarm.....	370
Formanty programu Alarm.....	371
Kod programu Alarm.....	373
Program Alarm w działaniu.....	381
Tworzenie programu Irys.....	382
Formularz programu Irys.....	383
Formanty programu Irys.....	383
Kod programu Irys.....	384
Program Irys w działaniu.....	394
Skorowidz.....	397

Rozdział 3.

Tworzenie programów sterowanych zdarzeniami

Visual Basic .NET służy do tworzenia programów sterowanych zdarzeniami, czyli takich, które wykonują zadania w odpowiedzi na zdarzenia wywoływane przez użytkownika, system operacyjny lub inny program.

Zdarzenia odgrywają kluczową rolę w programowaniu. Ilekroć użytkownik przesuwają mysz, naciska klawisz lub naciska przycisk myszy, zachodzi zdarzenie rejestrowane przez system operacyjny i aktywny program.

Formularz i większość formantów udostępnia zestaw zdarzeń. Tworząc program, programista dołącza metody do wybranych zdarzeń. Jeżeli na przykład programista dołączył metodę `MojaMetoda` do zdarzenia `Click` formantu `MójFormant`, metoda `MojaMetoda` jest wykonywana ilekroć zachodzi zdarzenie `Click` formantu `MójFormant`, czyli ilekroć użytkownik klika formant `MójFormant`.

Metoda, która ma zostać dołączona do zdarzenia, nie może zwracać wartości i musi posiadać sygnaturę odpowiednią dla tego zdarzenia. Jest kilka sposobów dołączania metody do zdarzenia. To, w jaki sposób dołączamy metodę do zdarzenia, zależy od tego, w jaki sposób tworzymy obiekt, który udostępnia to zdarzenie.

Tworzenie programu Zdarzenia

Uruchom Visual Studio .NET, utwórz pakiet i zapisz go pod nazwą *Zdarzenia*, a następnie dodaj do niego projekt języka Visual Basic i zapisz go pod nazwą *Zdarzenia*.

Formularz programu Zdarzenia

Program Zdarzenia będzie składał się z jednego formularza.

1. Do projektu *Zdarzenia* dodaj formularz i zapisz go pod nazwą *Zdarzenia.vb*.
2. Aby otworzyć arkusz właściwości, z menu *View* wybierz pozycję *Properties Window*. Upewnij się, że formularz *Zdarzenia* jest otwarty w oknie projektowym i kliknij jego pasek tytułu, aby go wybrać.
3. W arkuszu właściwości właściwość *FormBorderStyle* formularza ustaw na wartość *FixedSingle*, właściwość *Text* na wartość *Zdarzenia*, właściwość *Name* na wartość *Zdarzenia*, właściwość *Size* na wartość *225;158*, właściwość *StartPosition* na wartość *CenterScreen*, a właściwość *MaximizeBox* na wartość *False*.
4. Aby zapisać zmiany, z menu *File* wybierz pozycję *Save All*.

Kod potrzebny do uruchomienia programu Zdarzenia

Aby można było uruchomić program *Zdarzenia*, musi on posiadać metodę *Main*.

1. Wyświetl okno *Solution Explorer* (w tym celu z menu *View* wybierz pozycję *Solution Explorer*).
2. Aby otworzyć edytor kodu formularza *Zdarzenia*, zaznacz jego nazwę w oknie *Solution Explorer* i z menu *View* wybierz pozycję *Code*.

Edytor kodu zawiera kod wygenerowany przez Visual Basic .NET potrzebny do utworzenia formularza *Zdarzenia*. Większość tego kodu jest ukryta wewnątrz regionu.

3. Powyżej wiersza kodu `Public Class Zdarzenia` wpisz kod przedstawiony na wydruku 3.1.

Wydruk 3.1. Kod umieszczony przed definicją klasy *Zdarzenia*

```
Option Compare Binary
Option Explicit On
Option Strict On

Imports System.Windows.Forms
```

4. Wewnątrz klasy *Zdarzenia* poniżej wiersza kodu `Inherits System.Windows.Forms.Form` dodaj kod metody *Main* przedstawiony na wydruku 3.2.

Wydruk 3.2. Kod metody *Main* programu *Zdarzenia*

```
Public Shared Sub Main()
    Application.Run(New Zdarzenia())
End Sub
```

Metoda *Main*, punkt startowy programu *Zdarzenia*, wyświetla formularz *Zdarzenia*.

5. Aby zapisać zmiany, z menu *File* wybierz pozycję *Save All*.

Formant A

Teraz do formularza *Zdarzenia* dodamy formant *Button*, posługując się wskaźnikiem myszy, a następnie do jego zdarzenia *Click* dołączymy metodę *A*, korzystając z instrukcji *Handles*.

Dodanie formantu

1. Wyświetl formularz *Zdarzenia* w oknie projektowym — w tym celu w oknie *Solution Explorer* zaznacz nazwę formularza *Zdarzenia* i z menu *View* wybierz pozycję *Designer*.
2. Do formularza *Zdarzenia* dodaj formant *Button* — w tym celu kliknij ikonę reprezentującą formant *Button* w oknie *Toolbox*, umieść wskaźnik myszy nad formularzem, naciśnij lewy przycisk myszy, a następnie narysuj prostokąt w formularzu.

Formant *Button* służy do uruchamiania akcji lub zbioru akcji.

3. Aby otworzyć arkusz właściwości, z menu *View* wybierz pozycję *Properties Window*.
4. Wybierz (kliknij) dodany do formularza *Zdarzenia* formant *Button* i w arkuszu właściwości ustaw jego właściwość *Cursor* na wartość *Default*, właściwość *Text* na wartość *A*, właściwość *Name* na wartość *btnA*, właściwość *Location* na wartość *75;16*, a właściwość *Size* na wartość *75;23*.
5. Aby zapisać zmiany, z menu *File* wybierz pozycję *Save All*.

Metoda dołączona do zdarzenia

1. Wyświetl edytor kodu formularza *Zdarzenia* — w tym celu w oknie *Solution Explorer* zaznacz nazwę formularza *Zdarzenia* i z menu *View* wybierz pozycję *Code*.
2. Wewnątrz klasy *Zdarzenia* poniżej innych metod dodaj kod metody *A* przedstawiony na wydruku 3.3.

Wydruk 3.3. Kod metody *A* dołączonej do zdarzenia *Click* formantu *btnA*

```
Private Sub A( _  
    ByVal sender As System.Object, _  
    ByVal e As System.EventArgs) _  
    Handles btnA.Click  
  
    MessageBox.Show( _  
        "Życie jest snem.", _  
        "Pedro Calderon de la Barca", _  
        MessageBoxButtons.OK, _  
        MessageBoxIcon.Information)  
End Sub
```

Metoda A jest dołączona do zdarzenia Click formantu btnA, o czym świadczy instrukcja Handles btnA.Click kończąca nagłówek tej metody.

Instrukcja Handles pozwala dołączyć metodę do wybranego zdarzenia formularza lub formantu.

Składnia instrukcji Handles:

```
Sub NazwaMetody() Handles NazwaObiektu.NazwaZdarzenia
```

Kod umieszczony wewnątrz metody A korzysta z metody Show klasy MessageBox do wyświetlenia okna komunikatu z tekstem, tytułem, przyciskiem oraz ikoną.

3. Aby zapisać zmiany, z menu *File* wybierz pozycję *Save All*.

Formant B

Teraz do formularza *Zdarzenia* dodamy drugi formant Button, pisząc kod wewnątrz klasy *Zdarzenia*, a następnie do jego zdarzenia Click dołączymy metodę B, korzystając z instrukcji Handles.

Dodanie formantu

1. Wyświetl edytor kodu formularza *Zdarzenia* — w tym celu w oknie *Solution Explorer* zaznacz nazwę formularza *Zdarzenia* i z menu *View* wybierz pozycję *Code*.
2. Wewnątrz klasy *Zdarzenia* poniżej innych metod dodaj kod przedstawiony na wydruku 3.4.

Wydruk 3.4. Kod tworzący formant btnB

```
Private WithEvents btnB As Button

Private Sub UtwórzFormantB()
    btnB = New Button()

    btnB.Cursor = Cursors.Hand
    btnB.Text = "B"
    btnB.Name = "btnB"
    btnB.Left = 75
    btnB.Top = 48
    btnB.Width = 75
    btnB.Height = 23

    Me.Controls.Add(btnB)
End Sub
```

Jeżeli za pomocą kodu chcesz utworzyć formant Button, by potem do jego zdarzenia dołączyć metodę korzystając z instrukcji Handles, najpierw musisz zadeklarować zmienną (typ danych Button) korzystając ze słowa kluczowego WithEvents (formant zostaje utworzony dopiero wewnątrz metody UtwórzFormantB):

```
Private WithEvents btnB As Button
```

Metoda `UtwórzFormantB` zawiera: wiersz kodu, który tworzy instancję klasy `Button`, czyli formant `Button`:

```
btnB = New Button()
```

siedem wierszy kodu ustawiających właściwości formantu `Button` (w przypadku dodawania formantu do formularza za pomocą wskaźnika myszy właściwości te ustawia się w arkuszu właściwości):

```
btnB.Cursor = Cursors.Hand  
btnB.Text = "B"  
btnB.Name = "btnB"  
btnB.Left = 75  
btnB.Top = 48  
btnB.Width = 75  
btnB.Height = 23
```

oraz wiersz kodu dodający formant `Button` do zbioru formantów formularza:

```
Me.Controls.Add(btnB)
```

3. Aby zapisać zmiany, z menu *File* wybierz pozycję *Save All*.

Metoda dołączona do zdarzenia

1. Wyświetl edytor kodu formularza *Zdarzenia* — w tym celu w oknie *Solution Explorer* zaznacz nazwę formularza *Zdarzenia* i z menu *View* wybierz pozycję *Code*.
2. Wewnątrz klasy *Zdarzenia* poniżej innych metod dodaj kod metody `B` przedstawiony na wydruku 3.5.

Wydruk 3.5. Kod metody `B` dołączonej do zdarzenia *Click* formantu `btnB`

```
Private Sub B( _  
    ByVal sender As System.Object, _  
    ByVal e As System.EventArgs) _  
    Handles btnB.Click  
  
    MessageBox.Show( _  
        "Światłość w sierpniu.", _  
        "William Faulkner", _  
        MessageBoxButtons.OK, _  
        MessageBoxIcon.Information)  
End Sub
```

Metoda `B` jest dołączona do zdarzenia `Click` formantu `btnB`, o czym świadczy instrukcja `Handles btnB.Click` kończąca nagłówek tej metody.

Kod umieszczony wewnątrz metody `B` korzysta z metody `Show` klasy `MessageBox` do wyświetlenia okna komunikatu z tekstem, tytułem, przyciskiem oraz ikoną.

3. Aby zapisać zmiany, z menu *File* wybierz pozycję *Save All*.

Formant C

Teraz do formularza *Zdarzenia* dodamy trzeci formant *Button*, pisząc kod wewnątrz klasy *Zdarzenia*, a następnie do jego zdarzenia *Click* dołączymy metodę *C*, korzystając z instrukcji *AddHandler*.

Dodanie formantu

1. Wyświetl edytor kodu formularza *Zdarzenia* — w tym celu w oknie *Solution Explorer* zaznacz nazwę formularza *Zdarzenia* i z menu *View* wybierz pozycję *Code*.
2. Wewnątrz klasy *Zdarzenia* poniżej innych metod dodaj kod przedstawiony na wydruku 3.6.

Wydruk 3.6. Kod tworzący formant *btnC*

```
Private btnC As Button

Private Sub UtwórzFormantC()
    btnC = New Button()

    btnC.Cursor = Cursors.Cross
    btnC.Text = "C"
    btnC.Name = "btnC"
    btnC.Left = 75
    btnC.Top = 80
    btnC.Width = 75
    btnC.Height = 23

    Me.Controls.Add(btnC)

    AddHandler btnC.Click, AddressOf C
End Sub
```

Jeżeli za pomocą kodu chcesz utworzyć formant *Button*, by potem do jego zdarzenia dołączyć metodę korzystając z instrukcji *AddHandler*, najpierw musisz zadeklarować zmienną (typ danych *Button*) bez słowa kluczowego *WithEvents* (formant zostaje utworzony dopiero wewnątrz metody *UtwórzFormantC*):

```
Private btnC As Button
```

Metoda *UtwórzFormantC* zawiera: wiersz kodu, który tworzy instancję klasy *Button*, czyli formant *Button*:

```
    btnC = New Button()
```

siedem wierszy kodu ustawiających właściwości formantu *Button*:

```
    btnC.Cursor = Cursors.Cross
    btnC.Text = "C"
    btnC.Name = "btnC"
    btnC.Left = 75
    btnC.Top = 80
    btnC.Width = 75
    btnC.Height = 23
```

wiersz kodu dodający formant `Button` do zbioru formantów formularza:

```
Me.Controls.Add(btnC)
```

oraz wiersz kodu, który za pomocą instrukcji `AddHandler` dołącza metodę `C` do zdarzenia `Click` formantu `btnC`:

```
AddHandler btnC.Click, AddressOf C
```

Instrukcja `AddHandler` pozwala dołączyć metodę do zdarzenia formularza lub formantu.

Składnia instrukcji `AddHandler`:

```
AddHandler NazwaObiektu.NazwaZdarzenia, AddressOf NazwaMetody
```

Instrukcja `RemoveHandler` pozwala odłączyć metodę od zdarzenia formularza lub formantu.

Składnia instrukcji `RemoveHandler`:

```
RemoveHandler NazwaObiektu.NazwaZdarzenia, AddressOf NazwaMetody
```

3. Aby zapisać zmiany, z menu *File* wybierz pozycję *Save All*.

Metoda dołączona do zdarzenia

1. Wyświetl edytor kodu formularza *Zdarzenia* — w tym celu w oknie *Solution Explorer* zaznacz nazwę formularza *Zdarzenia* i z menu *View* wybierz pozycję *Code*.
2. Wewnątrz klasy *Zdarzenia* poniżej innych metod dodaj kod metody `C` przedstawiony na wydruku 3.7.

Wydruk 3.7. Kod metody `C` dołączonej do zdarzenia `Click` formantu `btnC`

```
Private Sub C( _  
    ByVal sender As System.Object, _  
    ByVal e As System.EventArgs)  
  
    MessageBox.Show( _  
        "Faraon.", _  
        "Bolesław Prus", _  
        MessageBoxButtons.OK, _  
        MessageBoxIcon.Information)  
End Sub
```

Kod umieszczony wewnątrz metody `C` korzysta z metody `Show` klasy `MessageBox` do wyświetlenia okna komunikatu z tekstem, tytułem, przyciskiem oraz ikoną.

3. Aby zapisać zmiany, z menu *File* wybierz pozycję *Save All*.

Kod potrzebny do uruchomienia metod UtwórzFormantB oraz UtwórzFormantC tworzących formanty

Zdarzenie Load formularza zachodzi, kiedy formularz jest wczytywany do pamięci operacyjnej. Do tego zdarzenia można dołączyć metodę, która ma zostać uruchomiona zanim użytkownik zobaczy formularz.

1. Wewnątrz klasy Zdarzenia poniżej innych metod dodaj kod metody UtwórzFormanty przedstawiony na wydruku 3.8.

Wydruk 3.8. Kod metody UtwórzFormanty

```
Private Sub UtwórzFormanty( _  
    ByVal sender As System.Object, _  
    ByVal e As System.EventArgs) _  
    Handles MyBase.Load  
  
    UtwórzFormantB()  
    UtwórzFormantC()  
End Sub
```

Metoda UtwórzFormanty jest dołączona do zdarzenia Load formularza, o czym świadczy instrukcja `Handles MyBase.Load` kończąca nagłówek tej metody.

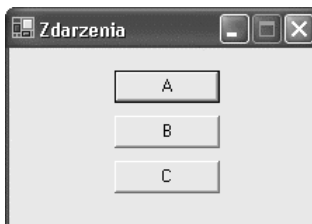
Kod umieszczony wewnątrz metody UtwórzFormanty uruchamia metodę UtwórzFormantB oraz metodę UtwórzFormantC.

2. Aby zapisać zmiany, z menu *File* wybierz pozycję *Save All*.

Program Zdarzenia w działaniu

Aby skompilować i uruchomić program Zdarzenia, z menu *Debug* wybierz pozycję *Start*. Okno programu Zdarzenia (formularz *Zdarzenia*) z trzema formantami Button pojawi się na środku ekranu (rysunek 3.1).

Rysunek 3.1.
*Program Zdarzenia
w działaniu*



Jeżeli klikniesz formant A, zostanie wygenerowane jego zdarzenie `Click`, w efekcie zostanie wykonana metoda A dołączona do tego zdarzenia i pojawi się okno komunikatu (rysunek 3.2).

Rysunek 3.2.

Efekt działania metody A



Jeżeli klikniesz formant B, zostanie wygenerowane jego zdarzenie Click, w efekcie zostanie wykonana metoda B dołączona do tego zdarzenia i pojawi się okno komunikatu (rysunek 3.3).

Rysunek 3.3.

Efekt działania metody B



Jeżeli klikniesz formant C, zostanie wygenerowane jego zdarzenie Click, w efekcie zostanie wykonana metoda C dołączona do tego zdarzenia i pojawi się okno komunikatu (rysunek 3.4).

Rysunek 3.4.

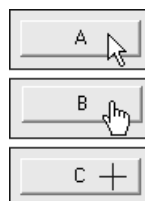
Efekt działania metody C



Jeżeli umieścisz wskaźnik myszy nad formantem A, wskaźnik przyjmie postać standardową. Jeżeli umieścisz wskaźnik myszy nad formantem B, wskaźnik przyjmie postać dłoni. Jeżeli umieścisz wskaźnik myszy nad formantem C, wskaźnik przyjmie postać krzyżyka (rysunek 3.5).

Rysunek 3.5.

Wygląd wskaźnika myszy nad formantem zależy od ustawienia właściwości Cursor tego formantu



Sygnatury metod dołączanych do zdarzeń formularzy i formantów

Edytor kodu pomaga utworzyć szablon metody z odpowiednią sygnaturą dla danego zdarzenia.

Aby na przykład utworzyć szablon metody, którą można dołączyć do zdarzenia Click formantu btnA, z listy rozwijanej Class Name (lewy górny róg edytora kodu) wybierz element btnA, a z listy rozwijanej Method Name (prawy górny róg edytora kodu) wybierz element Click. Aby natomiast utworzyć szablon metody, którą można dołączyć do zdarzenia Load formularza, z listy rozwijanej Class Name wybierz element (Base Class Events) (w przypadku wersji 2002 pakietu Visual Basic .NET) lub element ([Nazwa klasy formularza] Events) (w przypadku nowszych wersji), a z listy rozwijanej Method Name wybierz element Load.