

# Wprowadzenie

Pisanie programów komputerowych nie jest rzeczą trudną. Oczywiście tworzenie dużych systemów realizujących skomplikowane zadania wymaga dobrej wiedzy informatycznej i doświadczenia. Jednak proste programy można pisać już po kilku lekcjach. Visual Basic jest takim językiem programowania, który umożliwia pisanie całkiem pożytecznych i atrakcyjnych programów bez konieczności zdobywania głębokiej wiedzy o programowaniu.

Program komputerowy to zbiór instrukcji, które mówią komputerowi jak wykonać określone zadanie. Visual Basic umożliwia napisanie takiego zbioru instrukcji, a jednocześnie pozwala programiście zaprojektować atrakcyjny graficzny interfejs złożony z gotowych elementów znanych z profesjonalnych programów środowiska Windows bez konieczności programowania. Również rozbudowany system pomocy i podpowiedzi sprawia, że Visual Basic jest środowiskiem przyjaznym dla użytkownika. Visual Basic jest kolejną odmianą popularnego języka BASIC, znanego z najwcześniejszych komputerów osobistych, ale w odróżnieniu od swoich poprzedników pozwala tworzyć aplikacje dla środowiska Windows. W tej książce do przykładów i ćwiczeń wykorzystano wersję 6.0 tego języka.

Książka jest przeznaczona dla początkujących programistów, którzy chcieliby w szybki sposób poznać podstawy języka Visual Basic i nabyć umiejętności pisania prostych programów w środowisku Windows. Jej celem nie jest przedstawienie wszystkich możliwości języka Visual Basic, gdyż wymagałoby to kilkakrotnego przekroczenia jej objętości.

Książki przedstawiające języki programowania liczą zwykle kilkaset stron, dlatego też ograniczono się jedynie do najbardziej popularnych konstrukcji, które dają jednak możliwość tworzenia nie tylko trywialnych programów. Aby zbyt nie obciążać tekstu technicznymi szczegółami zrezygnowano z podawania dokładnej składni wszystkich omawianych instrukcji. Czytelnik zainteresowany pełną składnią poszczególnych instrukcji powinien zajrzeć do dokumentacji technicznej. W zamian za to zostały przedstawione liczne przykłady zastosowania omawianych elementów języka oraz środowiska Visual Basic, a także ćwiczenia podające praktyczny sposób ich wykorzystania.

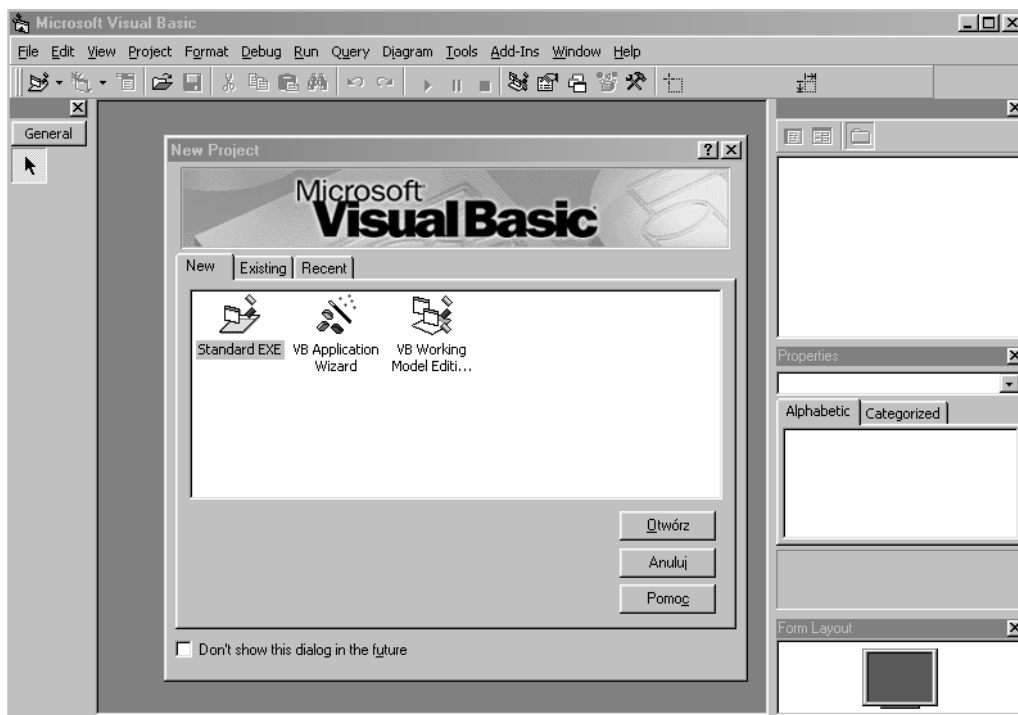
Każde ćwiczenie stanowi pełny projekt Visual Basica, który może zostać uruchomiony na komputerze. Podano krok po kroku wszystkie czynności jakie należy wykonać, aby zbudować odpowiedni formularz, a następnie szczegółowo omówiono tworzenie kodu programu. Prawie każde ćwiczenie stanowi odrębną całość i może być wykonane niezależnie od innych ćwiczeń. Pełne rozwiązania zamieszczonych w książce ćwiczeń znajdują się w dodatku oraz na załączonej dyskietce. W dodatku zapre-

zentowano również schematy blokowe algorytmów zbudowanych na potrzeby większości ćwiczeń. Na końcu kilku rozdziałów zamieszczono zadania do samodzielnego wykonania.

Mamy nadzieję, że zamieszczone przykłady, ćwiczenia oraz zadania do samodzielnego wykonania spełnią swoje cele i dadzą Czytelnikom umiejętność tworzenia prostych programów w Visual Basicu, a jednocześnie zachęcą do poszukiwania w ramach tego języka sposobów rozwiązania bardziej skomplikowanych zadań z jakimi mogą się spotkać w nauce czy pracy zawodowej.

# 1. Środowisko Visual Basic 6.0

Po zainstalowaniu Visual Basic 6.0, w menu Start utworzony został folder *Microsoft Visual Basic*, a w nim wstawione zostało polecenie Microsoft Visual Basic 6.0, które pozwala uruchomić środowisko programistyczne Visual Basic. Po uruchomieniu tego polecenia na ekranie pojawia się okno (rys. 1.1), które daje dostęp do wszystkich elementów tego środowiska.



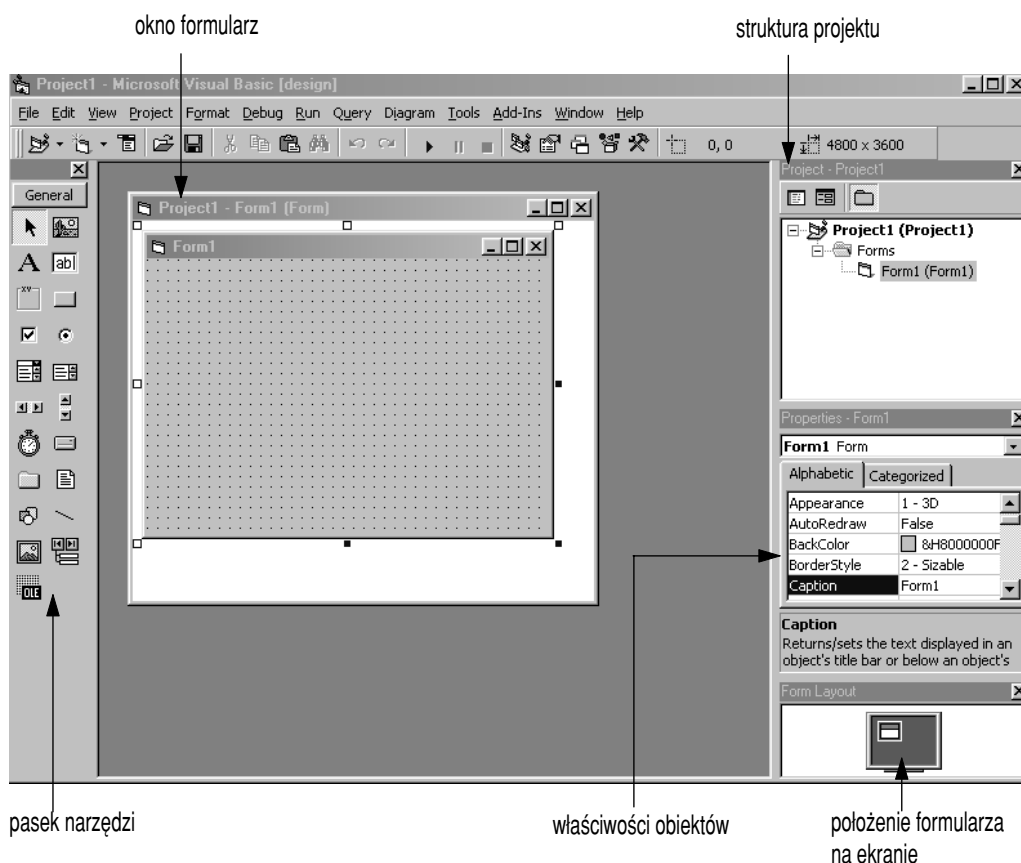
Rysunek 1.1. Początkowe okno Visual Basica po uruchomieniu

To pierwsze okno dialogowe, **New Project**, daje możliwość wyboru między trzema formami pracy:

- standardowym trybem tworzenia programów (*Standard EXE*);
- wspomaganym trybem tworzenia aplikacji (*VB Application Wizard*);
- edycją modelu (*VB Working Model Editing*).

Jednocześnie poszczególne zakładki dają dostęp albo do ostatnio opracowywanych projektów (**Recent**), albo do wszystkich projektów znajdujących się na dysku (**Existing**), albo otwierają zupełnie nowy projekt (**New**).

Otwarcie nowego projektu w standardowym trybie pracy spowoduje pojawienie się następującego ekranu (rys. 1.2).



Rysunek 1.2. Okno nowego projektu

Ekran Visual Basica w trakcie tworzenia projektu podzielony jest na kilka obszarów pełniących różne funkcje (rys. 1.2).

Z lewej strony znajduje się pasek narzędzi zawierający kilkanaście ikon. Odpowiadają one elementom zwanym kontrolkami, które programista wykorzystuje w swoim programie. Kontrolki takie jak: przyciski, pola tekstowe, listy, rysunki, napisy, itp., są obiektami. Rozmieszcza się je w innym obiekcie zwanym formularzem, który stanowi interfejs użytkownika, a więc pole komunikacji między opracowanym programem a wykorzystującym go użytkownikiem.

Na środku ekranu znajduje się okno **Project1 – Form1 (Form)** zawierające początkową postać interfejsu użytkownika – formularz. Formularz ten nosi nazwę: Form1 oraz wyposażony jest standardowo w górnym prawym rogu w trzy przyciski pozwalające zwinąć go do paska, rozwinąć do maksymalnych rozmiarów oraz zamknąć.

Po prawej stronie ekranu znajdują się trzy okna zawierające kolejno:

- strukturę projektu, nad którym aktualnie pracujemy;
- właściwości wszystkich obiektów znajdujących się w aktualnym projekcie, w tym w szczególności właściwości formularza;
- położenie formularza na ekranie w momencie uruchomienia tworzonej aplikacji.

W momencie uruchomienia nowego projektu jego struktura zawiera aktualnie tylko jeden formularz. Jednak bardziej złożone aplikacje mogą zawierać kilka lub kilkanaście różnych obiektów. W oknie właściwości wyświetlane są wszystkie cechy aktualnie aktywnego obiektu. W szczególności dla formularza znajdują tam się takie cechy, jak nazwa – **Name**, tytuł jego okna – **Caption**, kolor jego tła – **Background Color**, jego rozmiary itp. Niektóre właściwości (cechy) aktualnie aktywnego obiektu można zmieniać bezpośrednio manipulując obiektem, np. jego rozmiary, zaś inne są dostępne jedynie w oknie właściwości.

### Ćwiczenie 1.1.

*Zmienić niektóre właściwości formularza.*

- ◆ W oknie właściwości odszukaj właściwość **Caption** i zmień ją z Form1 na Mój pierwszy projekt.

Warto zauważyć zmianę, która zaszła w aktualnym formularzu w oknie **Project1**. Należy też zauważyć, że początkowo formularz posiadał dwie właściwości, które miały wartość Form1. Była to właściwość **Name** oraz **Caption**. Należy dobrze odróżniać te dwie właściwości. Właściwość **Name** definiuje nazwę danej kontrolki i pozwala na odwołanie się do niej w kodzie programu. Natomiast właściwość **Caption** to napis jaki się pojawia na niektórych kontrolkach np. właśnie na pasku formularza.

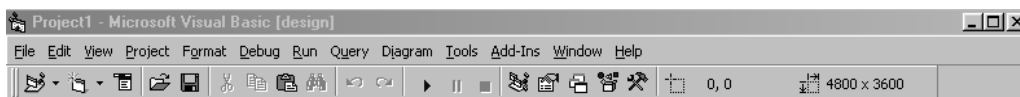
- ◆ Przewiń okno właściwości w taki sposób, aby widoczna była właściwość **Width** (szerokość).
- ◆ Uaktywnij formularz w oknie **Project1** i zmień jego szerokość przeciągając za pomocą myszy czarny kwadracik po prawej stronie formularza.
- ◆ Zaobserwuj zmiany zachodzące w oknie właściwości w polu **Width**.



Oprócz formularzy i zawartych w nich obiektów aktualny projekt zawiera też okno, w którym znajduje się kod programu. Znajdują się w nim teksty procedur, za pomocą których można wykonywać niestandardowe działania zaprojektowane przez programistę. Aby wywołać to okno należy z menu **View** wybrać opcję **Code**.




Innym sposobem przejścia do okna kodu źródłowego jest dwukrotnie kliknięcie danego obiektu, w tym przypadku formularza. Działanie to spowoduje przejście do okna kodu źródłowego programu. Z reguły dwukrotnie klikając dany obiekt przechodzimy do miejsca w kodzie źródłowym, które obsługuje najczęściej występujące zdarzenie związane z danym obiektem. Na przykład w wyniku dwukrotnego kliknięcia formularza, Visual Basic przejdzie do procedury `Form_Load`, która odpowiada za umieszczenie formularza w pamięci komputera.

Oprócz omówionych części składowych środowisko Visual Basica, jak każda aplikacja Windows, zawiera także pasek menu i pasek narzędzi (rys. 1.3).



Rysunek 1.3. Pasek menu i pasek narzędzi Visual Basic

Pasek menu zawiera wszystkie polecenia jakie jest w stanie wykonać Visual Basic, natomiast w pasku narzędzi znajdują się ikony-skróty do najczęściej stosowanych poleceń. Większość poleceń posiada też skróty klawiszowe pozwalające na uruchomienie danego polecenia bezpośrednio z klawiatury. Na przykład zapisanie projektu na dysku można zrealizować wybierając z menu **File** polecenie **Save**, lub klikając ikonę **Save** na pasku narzędzi, ewentualnie naciskając z klawiatury **Ctrl+S**.



Aby zobaczyć działanie tworzonych projektu należy z menu **Run** wybrać opcję **Start** lub z paska narzędzi wybrać odpowiadającą jej ikonę . Zakończenie działania programu uzyskuje się przez wybranie opcji **End** z menu **Run** lub odpowiadającej jej ikony . Można też zamknąć formularz za pomocą umieszczonego w jego oknie przycisku  lub też w inny sposób przewidziany przez programistę.

### Ćwiczenie 1.2.

*Uruchomić projekt zawierający pusty formularz umieszczając go początkowo na środku ekranu.*

- ◆ W oknie właściwości odszukaj właściwość **Caption** i zmień ją na Pusty formularz.
- ◆ W oknie położenie formularza (**Form layout**) przesuń formularz na środek ekranu za pomocą myszki.
- ◆ Uruchom projekt wybierając polecenie **Start** z menu **Run**.

Zauważ, że pojawił się na środku ekranu pusty formularz, nie zawierający żadnych elementów oprócz przycisków na pasku tytułowym.

- ◆ Aby zakończyć ten projekt i zamknąć formularz wybierz polecenie **End** z menu **Run**, albo wybierz przycisk  znajdujący się na pasku narzędzi Visual Basica lub też naciśnij przycisk  na pasku tytułowym formularza. Każde z tych działań spowoduje powrót do środowiska Visual Basic.



## 2. Tworzenie projektu

Rozwiązanie postawionego zadania za pomocą programu w języku Visual Basic wymaga wykonania trzech kroków:

- zaprojektowania i utworzenia interfejsu użytkownika, za pomocą którego program będzie komunikował się ze światem zewnętrznym;
- opracowania algorytmu rozwiązania postawionego problemu oraz napisania na tej podstawie kodu źródłowego programu;
- uruchomienia napisanego programu i zaprojektowanego interfejsu użytkownika oraz poprawienia ewentualnych błędów i niedociągnięć, a następnie skompilowania projektu do postaci programu wykonawczego, który do działania nie będzie wymagał obecności Visual Basic w pamięci.

Każdy z tych kroków zawiera szereg drobnych zadań i problemów składowych, które należy rozwiązać.

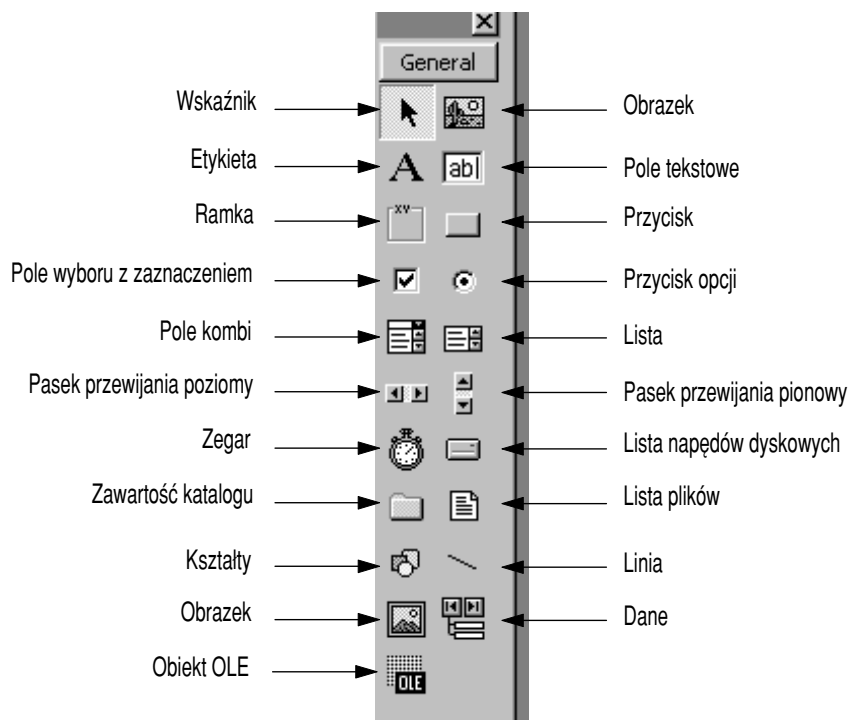
### 2.1. Komunikacja z użytkownikiem

Program napisany w języku Visual Basic komunikuje się z użytkownikiem poprzez interfejs. Często interfejs użytkownika stanowi odpowiednio zaprojektowany formularz. W formularzu umieszczone mogą być kontrolki takie jak przyciski, pola tekstowe, menu które pozwalają użytkownikowi na manipulowanie nimi w celu wykonania odpowiednich poleceń czy spowodowania odpowiednich zachowań.

Środowisko Visual Basic pozwala w łatwy sposób utworzyć formularz i wypełnić go potrzebnymi kontrolkami. Standardowe kontrolki jakimi dysponuje Visual Basic pokazane są na rysunku 2.1.

Aby umieścić wybraną kontrolkę na formularzu należy po prostu ją dwukrotnie kliknąć. Po pojawieniu się kontrolki na formularzu możemy ją dowolnie przesuwać oraz zmieniać jej wielkość.





Rysunek 2.1. Standardowe kontrolki dostępne w oknie Visual Basic

### **Ćwiczenie 2.1.**

Zbudować formularz, który będzie zawierał kilka kontrolki. Po uruchomieniu go, żadna z kontrolki nie będzie zdolna wykonać jakiegokolwiek zadania (do tego potrzebne jest odpowiednie oprogramowanie zdarzeń związanych z kontrolkami). Wyjście z formularza (i z programu) będzie możliwe dzięki przyciskowi zamknięcia okna formularza w jego prawym górnym rogu.

### **Wskazówki do rozwiązania**

#### **1. Tworzenie formularza**

- 1.1. Utwórz nowy projekt typu Standard EXE wybierając z menu File polecenie NewProject.
- 1.2. W oknie właściwości zmień nazwę (Name) utworzonego formularza z *Frame1* na *Pierwszy* oraz tytuł (Caption) formularza z *Frame1* na *Mój pierwszy projekt*.
- 1.3. Umieść na formularzu kontrolkę etykieta (Label) dwukrotnie ją klikając i w oknie właściwości zmień jej napis (Caption) na *Próbujemy różne kontrolki*.

- Zmień czcionkę (Font) na Times New Roman 14, pogrubiona oraz wyśrodkuj napis (właściwość położenia (Alignment) ustaw na Center).
- 1.4. Następnie umieść na formularzu kontrolkę pole tekstowe (TextBox), a nad nią etykietę (Label), której w oknie właściwości zmień napis (Caption) na *Pole tekstowe*.
  - 1.5. Umieść dalej na formularzu kontrolkę przycisk polecenia (CommandButton), zmień jej rozmiar za pomocą myszki oraz zmień w oknie właściwości jej nazwę (Name) na *cmdOK*, napis (Caption) na niej na *OK* oraz czcionkę (Font) na Times New Roman 12, pogrubiona kursywa.
  - 1.6. W podobny sposób dodaj przycisk (CommandButton) o nazwie *cmdAnuluj* i napisie *Anuluj*.
  - 1.7. Umieść na formularzu kontrolkę ramka (Frame) i zmień jej napis (Caption) na *Zaznacz*, a następnie umieść wewnątrz tej ramki trzy kontrolki pola wyboru z zaznaczeniem (CheckBox) i nazwij je odpowiednio: *Pole zaznaczenia nr 1*, *Pole zaznaczenia nr 2*, *Pole zaznaczenia nr 3*.
  - 1.8. Następnie umieść drugą kontrolkę typu ramka (Frame), zmiań jej napis (Caption) na *Wybierz* oraz umieść w niej dwa przyciski opcji (OptionButton), z napisami *Pierwsza opcja* i *Druga opcja*.




Rysunek 2.2. Formularz z rozmieszczonymi na nim różnymi kontrolkami

- 1.9. Następnie umieść na formularzu paski przewijania: poziomy (HScrollBar) oraz pionowy (VScrollBar).
- 1.10. Następną kontrolką, którą umieścisz na formularzu jest pole kombi (ComboBox), nad którym umieścisz etykietę (Label) z napisem *Lista kombi*.
- 1.11. Z lewej strony formularza w jego dolnej części umieść kontrolkę lista napędów dyskowych (DriveListBox), a pod nią kontrolkę lista plików (FileListBox).

Porównaj utworzony formularz z formularzem pokazanym na rysunku 2.2. Jeżeli wszystkie kroki wykonane zostały poprawnie to Twój formularz będzie identyczny z pokazanym na rysunku.



Zauważmy, że po uruchomieniu formularza za pomocą opcji **Start** z menu **Run** (lub kliknięciu przycisku  na pasku narzędzi) pewne kontrolki potrafią same wykonać pewne zadania, bez żadnego dodatkowego oprogramowania. W polu tekstowym można wpisać i korygować dowolny tekst, który następnie może być wykorzystany w programie. Paski przewijania poziomego i pionowego potrafią przesuwają się, a pola wyboru zaznaczają się niewykluczająco natomiast przyciski opcji można zaznaczyć tylko wykluczająco (tzn. albo pierwszy albo drugi, a nie mogą być zaznaczone obydwa). Kontrolki napędów dyskowych oraz lista plików dają dostęp do zasobów systemu.

Oczywiście brak dodatkowego oprogramowania nie pozwala na powiązanie pomiędzy wybieranym napędem dyskowym a listą nazw plików wyświetlanych w kontrolce lista plików. Podobny problem dotyczy przycisków poleceń **OK** oraz **Anuluj**, których kliknięcie nie powoduje żadnych działań. Aby uzyskać oczekiwany od tych kontrolki efekt, trzeba napisać procedury związane ze zdarzeniem kliknięcia tych przycisków.

## 2.2. Przypisanie kodu programu do kontrolki

Kontrolki umieszczane na formularzu powinny odpowiednio reagować na zdarzenia wykonywane przez użytkownika, takie jak np. kliknięcie myszką. Aby tak się działo każde takie zdarzenie musi zostać oprogramowane. Niektóre zdarzenia są już standardowo oprogramowane w środowisku Visual Basic. Na przykład kliknięcie przycisku, który ma właściwość przycisku **Cancel** spowoduje wycofanie się z formularza, na którym przycisk ten został umieszczony. Programista może wszakże zmienić działanie takich predefiniowanych zachowań tworząc własne procedury obsługi zdarzeń.

### Ćwiczenie 2.2.

*Zdefiniujmy zachowanie się programu na skutek kliknięcia przez użytkownika przycisków OK oraz Anuluj na formularzu zbudowanym w poprzednim przykładzie.*

### **Wskazówki do rozwiązania**

#### **2. Pisanie kodu programu**

- 2.1. Dwukrotne kliknięcie przycisku *cmdOK* podczas projektowania formularza przeniesie cię do okna kodu, do procedury obsługi tego zdarzenia, czyli do procedury `Private Sub cmdOK_Click()`.
- 2.2. Wewnątrz tej procedury umieść wywołanie okna komunikatu `MsgBox` z komunikatem *Wcisnąłeś przycisk OK*. Procedura obsługi tego zdarzenia powinna wyglądać tak:


```
Private Sub cmdOK_Click()  
    MsgBox ("Wcisnąłeś przycisk OK.")  
End Sub
```

- 2.3. Po ponownym przejściu do projektowanego formularza dwukrotnie kliknij przycisk polecenia *cmdAnuluj*, co spowoduje przejście do procedury *cmdAnuluj\_Click*
- 2.4. Wewnątrz tej procedury umieść wywołanie procedury zakończenia pracy z formularzem i usunięcia go z pamięci, tzn. `Unload Me`. Ostatecznie procedura ta powinna wyglądać tak:

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

- 2.5. Zapisz projekt na dysku nadając mu nazwę *Pierwszy*. Zwróć uwagę na nadanie nazwy całemu projektowi (*Pierwszy.vbp*) oraz formularzowi (*Pierwszy.frm*). Obejrzyj następnie działanie tego projektu uruchamiając go opcją **Start** z menu **Run**.



Zauważmy, że tym razem kliknięcie przycisku **OK** powoduje wyświetlenie odpowiedniego komunikatu, natomiast kliknięcie przycisku **Anuluj** spowoduje zakończenie pracy programu, podobnie jak wybranie przycisku , **Zamknij formularz**.

#### **2.3. Wprowadzanie danych i wyprowadzanie wyników**

Aby program mógł wykonywać pewne czynności w zależności od różnych dostarczanych mu danych oraz prezentować wykonane obliczenia potrzebne są proste narzędzia do wprowadzania i wyprowadzania liczb i tekstów. Najprostszymi z nich jest umieszczenie pola tekstowego (**Text Box**) na formularzu w celu pobrania danych, oraz wydruk bezpośrednio na formularzu za pomocą instrukcji **print**.

Innym sposobem jest wprowadzanie danych za pomocą specjalnego formularza (tzw. okna dialogowego) **InputBox**, natomiast wyprowadzenie wyników w formie

komunikatu za pomocą osobnego małego formularza (również okna dialogowego) **Message Box** (jakiego użyliśmy już w Ćwiczeniu 2.2).

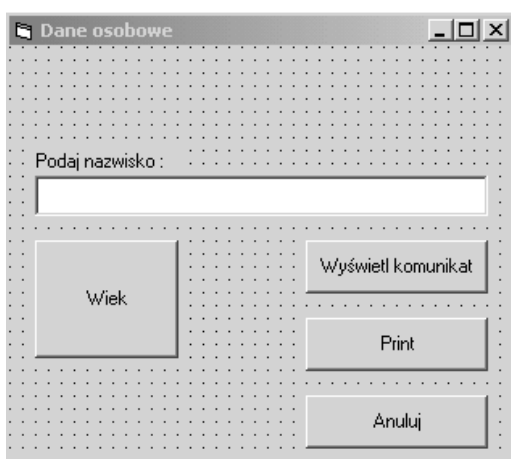
### Ćwiczenie 2.3.

*Podamy dwie dane: nazwisko oraz wiek, dwoma metodami, a następnie wyświetlimy komunikat o wieku danej osoby dwiema metodami: bezpośrednio na formularzu oraz w osobnym oknie dialogowym Message Box.*

### **Wskazówki do rozwiązania**

#### **1. Tworzenie formularza**

- 1.1. Utwórz nowy projekt typu Standard EXE wybierając opcję **New Project** z menu **File**. Zmień nazwę (Name) utworzonego formularza na *Osobowe* oraz tytuł (Caption) formularza na *Dane osobowe*.
- 1.2. Na formularzu umieść kontrolkę pole tekstowe (TextBox) podwójnie ją klikając na pasku narzędzi, w oknie właściwości zmień jej nazwę (Name) na *txtNazwisko* oraz usuń domyślną wartość *Text1* z właściwości tekst (Text).
- 1.3. Nad tą kontrolką formularza etykietę (Label) z napisem (Caption) *Podaj nazwisko: .*
- 1.4. Pod polem tekstowym umieść przycisk polecenia (CommandButton), zmień jego nazwę (Name) na *cmdWiek* oraz zmień napis (Caption) na nim na *Wiek*.
- 1.5. Następnie umieść trzy przyciski poleceń (CommandButton) o nazwach (Name) *cmdMsg*, *cmdPrint* oraz *cmdAnuluj* i napisach na nich (Caption) odpowiednio: *Wyświetl komunikat*, *Print* oraz *Anuluj*. Zaprojektowany formularz przedstawiony jest na rysunku 2.3.



Rysunek 2.3. Zaprojektowany formularz

- 1.6. Zapisz na dysku swoją pracę nadając nazwę *Osobowe* zarówno projektowi, jak i formularzowi.

## 2. Pisanie kodu programu

Aby formularz spełnił swoje zadanie należy oprogramować odpowiednie zdarzenia.

- 2.1. Dwukrotnie klikając kontrolkę pole tekstowe *txtNazwisko* przejdź do tworzenia procedury `Private Sub txtNazwisko_Change()` w oknie kodu. Pobierz tekst wpisany do tej kontrolki poprzez odczytanie właściwości `Text` obiektu *txtNazwisko* i podstaw go pod zmienną *Nazwisko*. Procedura ta powinna wyglądać następująco:

```
Private Sub txtNazwisko_Change()  
    Nazwisko = txtNazwisko.Text  
End Sub
```

- 2.2. Wróć ponownie do okna obiektu. Dwukrotne kliknięcie przycisku polecenia *Wiek* przeniesie cię ponownie do okna kodu, do procedury obsługującej zdarzenie kliknięcia przycisku, czyli do procedury `cmdWiek_Click()`. Dane o wieku wprowadź za pomocą formularza *InputBox* wywołując ten formularz z komunikatem *Podaj wiek*. Procedura ta powinna wyglądać następująco:

```
Private Sub cmdWiek_Click()  
    Wiek = InputBox("Podaj wiek")  
End Sub
```

- 2.3. W dalszej kolejności zaprogramuj zachowanie się programu w przypadku kliknięcia przycisków *Wyświetl komunikat*, *Print* oraz *Anuluj*. Naciśnięcie pierwszego przycisku powinno spowodować wyświetlenie za pomocą formularza *MsgBox* komunikatu o wprowadzonych danych. Procedura ta wygląda tak:

```
Private Sub cmdMsg_Click()  
    MsgBox Nazwisko & " ma "& Wiek & " lat."  
End Sub
```

Natomiast wybranie drugiego przycisku powinno spowodować wypisanie tego komunikatu bezpośrednio na formularzu. Procedura obsługi tego zdarzenia ma postać:

```
Private Sub cmdMsg_Click()  
    Print Nazwisko & " ma "& Wiek & " lat."  
End Sub
```

- 2.4. I ostatecznie przycisk **Anuluj** kończy działanie programu. W procedurze obsługującej kliknięcie tego przycisku umieść wywołanie funkcji powodująco-

jącej usunięcie formularza z pamięci komputera i zakończenie działania programu, czyli:

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

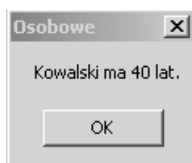
- 2.5. Należy zwrócić uwagę, że do przechowywania wprowadzonych danych użyliśmy dwóch zmiennych *Nazwisko* i *Wiek*. Aby zmienne te mogły być widoczne przez wszystkie procedury, trzeba je zadeklarować w początkowej części kodu (więcej na temat zmiennych, ich typów oraz ich deklarowania powiemy w następnym rozdziale). Umieść zatem na początku programu przed tekstem pierwszej procedury następującą deklarację:

```
Dim Wiek, Nazwisko As String
```

- 2.6. Zapisz projekt na dysku, a następnie spróbuj uruchomić zaprojektowany formularz i program za pomocą opcji **Start** z menu **Run**. Zaobserwuj działanie formularzy *InputBox* (patrz rysunek 2.4) oraz *MsgBox* (rys. 2.5), a także wykonanie polecenia *Print* (rys. 2.6).



Rysunek 2.4. Wprowadzanie danych za pomocą formularza *InputBox*



Rysunek 2.5. Wyprowadzanie komunikatów i wyników za pomocą formularza *MsgBox*