

Microsoft®

Zacznij Tu!

**Poznaj
Microsoft® Visual Basic®
2012**

Michael Halvorson

Przekład: Joanna Zatorska

APN Promise, Warszawa 2013

Zacznij Tu! Poznaj Microsoft® Visual Basic® 2012
© 2013 APN PROMISE SA

Authorized Polish translation of the English edition Start Here! Learn Microsoft® Visual Basic® 2012, ISBN 978-0-7356-7298-7
Copyright © 2013 by Michael Halvorson

This translation is published and sold by permission of O'Reilly Media, Inc., which owns or controls all rights to publish and sell the same.

APN PROMISE SA, biuro: ul. Kryniczna 2, 03-934 Warszawa
tel. +48 22 35 51 600, fax +48 22 35 51 699
e-mail: mspress@promise.pl

Wszystkie prawa zastrzeżone. Żadna część niniejszej książki nie może być powielana ani rozpowszechniana w jakiegokolwiek formie i w jakikolwiek sposób (elektroniczny, mechaniczny), włącznie z fotokopiowaniem, nagrywaniem na taśmy lub przy użyciu innych systemów bez pisemnej zgody wydawcy.

Książka ta przedstawia poglądy i opinie autorów. Przykłady firm, produktów, osób i wydarzeń opisane w niniejszej książce są fikcyjne i nie odnoszą się do żadnych konkretnych firm, produktów, osób i wydarzeń, chyba że zostanie jednoznacznie stwierdzone, że jest inaczej. Ewentualne podobieństwo do jakiegokolwiek rzeczywistej firmy, organizacji, produktu, nazwy domeny, adresu poczty elektronicznej, logo, osoby, miejsca lub zdarzenia jest przypadkowe i niezamierzone.

Nazwa Microsoft oraz znaki towarowe wymienione na stronie <http://www.microsoft.com/about/legal/en/us/IntellectualProperty/Trademarks/EN-US.aspx> są zastrzeżonymi znakami towarowymi grupy Microsoft. Wszystkie inne znaki towarowe są własnością ich odnośnych właścicieli.

APN PROMISE SA dołożyła wszelkich starań, aby zapewnić najwyższą jakość tej publikacji. Jednakże nikomu nie udziela się rękojmi ani gwarancji. APN PROMISE SA nie jest w żadnym wypadku odpowiedzialna za jakiegokolwiek szkody będące następstwem korzystania z informacji zawartych w niniejszej publikacji, nawet jeśli APN PROMISE została powiadomiona o możliwości wystąpienia szkód.

ISBN: 978-83-7541-128-7

Przekład: Joanna Zatorska
Redakcja: Marek Włodarz
Korekta: Ewa Swędrowska
Skład i łamanie: MAWart Marek Włodarz

Spis treści

| | |
|--|-----|
| <i>Wstęp</i> | vii |
| 1 Wprowadzenie do Visual Basic 2012 | 1 |
| Możliwości programistów Visual Basic | 2 |
| Nowe platformy programistyczne | 3 |
| Pobieranie, instalacja i uruchamianie pakietu Visual Studio Express | |
| 2012 dla Windows 8 | 5 |
| Pobieranie produktu | 5 |
| Instalacja pakietu Visual Studio Express 2012 dla Windows 8 | 6 |
| Uruchamianie programu Visual Studio Express 2012 | 6 |
| Środowisko programistyczne Visual Studio | 8 |
| Narzędzia pakietu Visual Studio | 10 |
| Okno Designer | 13 |
| Uruchamianie programu Visual Basic | 17 |
| Okno Properties | 20 |
| Zmiana położenia i rozmiaru narzędzi programistycznych | 24 |
| Zmiana położenia i rozmiaru okien narzędziowych | 25 |
| Dokowanie okien narzędziowych | 27 |
| Ukrywanie okien narzędziowych | 28 |
| Przełączanie między otwartymi plikami i przybornikami przy użyciu funkcji IDE Navigator (Nawigator IDE) | 29 |
| Otwieranie przeglądarki sieci Web w programie Visual Studio | 30 |
| Dostosowywanie ustawień środowiska IDE do potrzeb ćwiczeń | 32 |
| Sprawdzanie ustawień projektu i kompilatora | 32 |
| Zamykanie programu Visual Studio | 35 |
| Podsumowanie | 36 |
| 2 Tworzenie pierwszej aplikacji dla Windows 8 | 37 |
| Web List: pierwszy program Visual Basic | 37 |
| Tworzenie interfejsu użytkownika | 39 |
| Ustawianie właściwości | 48 |
| Pisanie kodu Visual Basic | 52 |
| Rzut oka na plik związany z kodem Visual Basic | 57 |
| Uruchamianie aplikacji Visual Basic | 58 |
| Przykładowe projekty na dysku | 61 |
| Tworzenie pliku wykonywalnego i wdrażanie aplikacji | 62 |
| Podsumowanie | 66 |

| | | |
|----------|--|-----|
| 3 | Używanie kontroltek | 69 |
| | Korzystanie z kontroltek <i>Ellipse</i> i <i>TextBlock</i> | 70 |
| | Korzystanie z kontroltek <i>CheckBox</i> i <i>RadioButton</i> | 79 |
| | Korzystanie z kontrolki <i>MediaElement</i> do wstawiania dźwięku i wideo..... | 91 |
| | Kilka słów o plikach multimedialnych..... | 92 |
| | Podsumowanie..... | 101 |
| 4 | Projektowanie aplikacji Windows 8 przy użyciu Blend for Visual Studio | 103 |
| | Blend for Visual Studio..... | 104 |
| | Dlaczego Blend jest użyteczny dla programistów Visual Studio..... | 104 |
| | Uruchamianie programu Blend..... | 105 |
| | Narzędzia projektowe środowiska Blend IDE..... | 108 |
| | Korzystanie z kontroltek XAML w programie Blend..... | 110 |
| | Tworzenie scenorysu w celu utworzenia podstawowych efektów animacji..... | 115 |
| | Pisanie procedur obsługi zdarzeń w programie Visual Studio..... | 122 |
| | Korzystanie ze zdarzenia <i>OnNavigatedTo</i> | 128 |
| | Podsumowanie..... | 131 |
| 5 | Praca z XAML | 133 |
| | Podstawy języka XAML..... | 133 |
| | Co to jest XAML?..... | 134 |
| | XAML jest związany z XML i HTML..... | 135 |
| | Elementy XAML..... | 135 |
| | Zapoznanie z plikami projektu XAML..... | 137 |
| | Tworzenie obiektów XAML..... | 144 |
| | Podsumowanie..... | 155 |
| 6 | Elementy języka Visual Basic | 157 |
| | Instrukcje programu Visual Basic..... | 158 |
| | Korzystanie ze zmiennych do przechowywania informacji..... | 159 |
| | Rezerwacja pamięci dla zmiennych: instrukcja <i>Dim</i> | 159 |
| | Używanie zmiennych w procedurze obsługi zdarzenia..... | 160 |
| | Używanie zmiennych do przechowywania i przetwarzania danych wejściowych..... | 164 |
| | Praca z typami danych..... | 169 |
| | Stałe: zmienne, które się nie zmieniają..... | 177 |
| | Praca z operatorami języka Visual Basic..... | 180 |
| | Podstawowe działania: operatory +, -, *, i /..... | 180 |
| | Operatory zaawansowane: \, Mod, ^ i &..... | 182 |
| | Ustalanie kolejności wykonywania działań..... | 184 |
| | Używanie nawiasów w formułach..... | 185 |
| | Podsumowanie..... | 185 |

| | | |
|-----------|---|-----|
| 7 | Kontrola projektu, układu i przepływu działania programu | 187 |
| | Tworzenie układu kafelkowego aplikacji dla Sklepu Windows | 188 |
| | Projektowanie stron pobierających dane od użytkownika | 189 |
| | Sprawdzanie określonych warunków przy użyciu instrukcji <i>If...Then...Else</i> | 199 |
| | Dni tygodnia w instrukcji <i>If...Then...</i> | 204 |
| | Kontrola przepływu działaniu programu przy użyciu pętli <i>For...Next</i> i <i>For Each...Next</i> | 205 |
| | Pętla <i>For...Next</i> | 205 |
| | Pętla <i>For Each...Next</i> | 208 |
| | Tworzenie procedury obsługi błędów w celu zarządzania warunkami wystąpienia błędów | 213 |
| | Podsumowanie | 217 |
| 8 | Korzystanie z platformy .NET Framework | 219 |
| | Pomysłowe programowanie: używanie bibliotek klas platformy .NET Framework | 220 |
| | Terminologia zorientowana obiektowo | 221 |
| | Korzystanie z okna Object Browser | 222 |
| | Używanie metod <i>System.String</i> | 227 |
| | Używanie metod klasy <i>System.Math</i> | 235 |
| | Praca z liczbami losowymi | 237 |
| | Wstawianie fragmentów gotowego kodu | 239 |
| | Podsumowanie | 244 |
| 9 | Debugowanie aplikacji | 245 |
| | Odnajdywanie i naprawa błędów | 246 |
| | Trzy rodzaje błędów | 246 |
| | Rozpoznawanie błędów logicznych | 247 |
| | ABC debugowania: korzystanie z trybu debugowania | 248 |
| | Śledzenie zmiennych przy użyciu okna Watch | 254 |
| | Wizualizatory: narzędzia do debugowania wyświetlające dane | 257 |
| | Użycie okna Immediate | 259 |
| | Usuwanie punktów przerwania | 261 |
| | Podsumowanie | 261 |
| 10 | Zarządzanie strukturami danych przy użyciu tablic i LINQ | 263 |
| | Używanie tablic do przechowywania danych | 264 |
| | Deklarowanie tablic | 264 |
| | Deklarowanie tablic statycznych | 265 |
| | Korzystanie z tablic | 266 |
| | Przypisywanie wartości początkowych do tablicy | 268 |
| | Używanie metod klasy Array | 273 |
| | Wstęp do LINQ | 278 |
| | Składnia LINQ | 278 |

| | |
|---|------------|
| Praca z dokumentami XML | 287 |
| Używanie dokumentów XML w projekcie Visual Basic | 288 |
| Podsumowanie | 291 |
| 11 Centrum projektu: pięć fantastycznych funkcji dla aplikacji Windows 8 | 293 |
| Tworzenie kafelka aplikacji dla strony startowej systemu Windows | 294 |
| Tworzenie ekranu startowego dla aplikacji | 307 |
| Ustawianie uprawnień i możliwości aplikacji Windows 8 | 312 |
| Używanie szablonu projektu do prezentowania zawartości aplikacji | 315 |
| Optymalizowanie aplikacji pod kątem obsługi dotykiem i gestami | 323 |
| Wbudowana obsługa dotykiem | 324 |
| Naciśnięcie | 325 |
| Przesuwanie | 325 |
| Powiększanie i skalowanie | 326 |
| Projektowanie uwzględniające obsługę dotykiem | 327 |
| Podsumowanie | 327 |
| 12 Dalsze możliwości rozwoju i Sklep Windows | 329 |
| Przygotowanie programu dla Sklepu Windows | 329 |
| Poznawanie funkcji Sklepu | 330 |
| Ceny i sprzedaż | 332 |
| Przygotowanie do certyfikowania i wdrażania | 332 |
| Lista kontrolna Sklepu | 333 |
| Dalsze możliwości i zasoby programistyczne | 336 |
| Witryny sieci Web dotyczące języka Visual Basic i systemu Windows 8 | 337 |
| Zasoby wideo w sieci Web | 338 |
| Książki dotyczące języka Visual Basic i pakietu Visual Studio | 339 |
| Podsumowanie | 341 |
| <i>Indeks</i> | 343 |
| <i>O autorze</i> | 357 |

Wstęp

Microsoft Windows 8 to potężny i przykuwający uwagę system operacyjny, zaprojektowany do radykalnego zwiększenia efektywności klienta oraz oferujący dostęp do szerokiego asortymentu produktów i usług sieciowych. Najważniejszą cechą systemu Windows 8 jest bogactwo doświadczeń użytkownika. Wygląd i sposób postrzegania aplikacji Windows 8 sprawiają, że użytkownik może uzyskać szybki dostęp do muzyki, zdjęć, kontaktów i ustawień w chmurze internetowej, zaś Sklep Windows pozwala konsumentom na szybki dostęp do fascynujących aplikacji. Windows 8 został zaprojektowany do działania na szerokiej gamie urządzeń, począwszy od tabletów wyposażonych w ekrany dotykowe, poprzez laptopy, do tradycyjnych komputerów stacjonarnych. Klienci, którzy zanurzą się w setkach ekscytujących aplikacji Windows 8, będą się mogli całkowicie skupić na swoich zadaniach, nie tracąc czasu na wydawanie zbędnych poleceń czy uruchamianie funkcji systemu operacyjnego.

Windows 8 prezentuje również rewelacyjne możliwości z punktu widzenia programisty; jest to szybki, bezpieczny i solidny system, który będzie zainstalowany na milionach komputerów na świecie, włącznie z tabletami Microsoft Surface. Aplikacje Windows 8 są ekscytujące i łatwe w użyciu, a ponadto oferują konsumentom bogaty w treści interfejs i działają równie dobrze na urządzeniach obsługiwanych dotykiem, jak i na komputerach PC. Najważniejsze jest to, że Sklep Windows pozwala programistom na bezpośrednią sprzedaż aplikacji Windows 8 na rynku globalnym, udostępniając nowe źródło dochodów i ułatwiając procedury instalacji.

Ta książka prezentuje tworzenie przykuwających uwagę aplikacji Windows 8 przy użyciu programu Microsoft Visual Studio 2012, najnowszej wersji bestsellerowego pakietu służącego do tworzenia oprogramowania. Dowiemy się, jak pobrać bezpłatną wersję pakietu Visual Studio 2012 Express (tak jest – *darmow!*) oraz jak stworzyć ciekawe aplikacje Windows 8 z wykorzystaniem kilku narzędzi i technik, dostępnych w programie Visual Studio, włącznie z językiem programowania Visual Basic. Zanim ukończymy niniejszą książkę, nauczymy się tworzyć podstawowe funkcje aplikacji Windows 8; pracować efektywnie ze środowiskiem Visual Studio Integrated Development Environment (IDE); projektować interfejs użytkownika z wykorzystaniem kodu XAML oraz programu Blend for Visual Studio; pisać efektywny kod programu w języku Visual Basic; a także sprzedawać własne aplikacje w Sklepie Windows.

Oczywiście, jedną z najwspanialszych cech tej książki jest to, że każde opisywane tu narzędzie programistyczne jest *bezpłatne!* Firma Microsoft oferuje uzupełniający dostęp do wersji Express pakietu Visual Studio z nadzieją, że nauka programowania przy użyciu tego narzędzia przypadnie nam do gustu i że pewnego dnia staniemy się profesjonalnymi programistami Visual Basic, tworzącymi i sprzedającymi fantastyczne

aplikacje Windows. Wszystko, czego potrzebujemy, to system Windows 8 zainstalowany na kompatybilnym komputerze z dostępem do Internetu oraz chęć tworzenia programów w języku Visual Basic.

W rzeczywistości perspektywy dla profesjonalnych programistów Visual Basic nigdy nie były tak świetlane. Musimy tylko *zacząć!*

Dla kogo jest ta książka

Niniejsza książka to praktyczny podręcznik przeznaczony dla czytelników lubiących poznawać nowe zagadnienia poprzez ich wykonywanie. Założeniem książki *Zacznij Tu! Poznaj Microsoft Visual Basic 2012* jest to, że Czytelnik nie posiada żadnego doświadczenia z programem Visual Studio i językiem Visual Basic. Dlatego książka koncentruje się całkowicie na wstępnych koncepcjach i procedurach programistycznych. Czytelnicy będą zdumieni, jak wiele można osiągnąć w pakiecie Visual Studio będąc początkującym programistą i jak szybko można zacząć tworzenie własnych projektów. Zakładam, że po książkę sięgną inteligentni studenci, pasjonaci lub profesjonalści z branży IT, zainteresowani nauką programowania, którzy nie mieli wcześniej styczności z językiem Visual Basic lub z pakietem Visual Studio.

Niniejsza książka opisuje określone techniki tworzenia kodu w języku Visual Basic oraz zawiera szeroki przegląd strategii programistycznych. Ponadto pozwala poznać możliwości systemu operacyjnego Windows 8 oraz specyficzne wytyczne projektowe firmy Microsoft, tworzące nowe, ekscytujące podejście do budowania aplikacji Windows 8. Zasady projektowe dotyczące interfejsu użytkownika w systemie Windows 8 są niezwykle inspirujące i zachęcają programistów, aby traktowali priorytetowo programy bogate w treści i zorientowane na współpracę z siecią Web. Aplikacje Windows 8 reprezentują nowe sposoby współpracy oraz ekscytujące możliwości pracy z nowymi urządzeniami do zbierania danych, takimi jak wbudowane kamery, panele dotykowe, akcelerometry, żyroskop, kompas, odbiorniki GPS i czujniki natężenia światła.

Ogólnym celem książki *Zacznij Tu! Poznaj Microsoft Visual Basic 2012* jest doprowadzenie Czytelnika do etapu, na którym poczuje się pewnie z narzędziami programistycznymi programu Visual Studio i będzie umiał tworzyć własne proste aplikacje Windows 8. Wówczas będzie już gotów do zakupu bardziej wszechstronnej książki dotyczącej programowania w języku Visual Basic, na przykład książki mojego autorstwa *Microsoft Visual Basic 2012 Krok po kroku* (Microsoft Press, 2013).

Założenia

Ta książka ma na celu nauczyć Czytelników, którzy nie mają doświadczenia programistycznego, programowania w języku Visual Basic. Częścią tego procesu będzie nauka obsługi bezpłatnego pakietu Visual Studio 2012 Express. Pobieranie i instalacja pakietu

Visual Studio 2012 Express w systemie zostały opisane w rozdziale 1., „Wprowadzenie do Visual Basic 2012”.

W książce zakładam, że Czytelnik zakupił i zainstalował system operacyjny Windows 8 i że chciałby się nauczyć tworzenia aplikacji Windows 8. Aplikacje te będą prostymi programami działającymi pod kontrolą systemu Windows 8, spełniającymi podstawowe wytyczne dotyczące interfejsu użytkownika. Będą (lub powinny być) zaprojektowane w taki sposób, aby możliwie najlepiej wykorzystać liczne zasoby i powiązania dostępne w sieci Web. Aplikacje Windows 8 są niezwykle interaktywne i zostały przygotowane do pobierania przez użytkowników ze Sklepu Windows.

Aby wykonać większość zadań programistycznych, Czytelnik powinien umieć wykonywać proste zadania w systemie Windows 8, korzystać z informacji dostępnych w sieci Web, dostosowywać stronę startową i interfejs użytkownika oraz dostosowywać podstawowe ustawienia systemu. Posiadacze tabletów lub paneli dotykowych, działających pod kontrolą systemu Windows 8, są nawet w lepszej sytuacji, ponieważ w systemie Windows 8 kładzie się szczególny nacisk na upowszechnienie obsługi dotykiem i gestami. Aplikacje można zbudować na laptopie lub komputerze stacjonarnym, na którym zainstalowany jest pakiet Visual Studio 2012 i system Windows 8, a następnie przetestować je na tablecie lub urządzeniu wyposażonym w panel dotykowy.

Posiadacze pełnych wersji komercyjnych pakietu Visual Studio 2012 będą mogli utworzyć więcej rodzajów aplikacji, niż opisano w niniejszej książce – Visual Studio Express 2012 dla Windows 8 pozwala na utworzenie jedynie aplikacji Windows 8. W bardziej zaawansowanej książce, na przykład w publikacji *Microsoft Visual Basic 2012 Krok po kroku* dowiemy się, jak tworzyć aplikacje HTML, aplikacje konsolowe, programy dla telefonów Windows Phone oraz aplikacje dla komputerów stacjonarnych (projekty Windows Forms) działających pod kontrolą systemu Windows 8 i Windows 7.

Kto nie powinien czytać tej książki

Zaawansowani programiści lub ci, którzy pragną poznać Visual Basic jako drugi język programowania, z pewnością będą rozczarowani. Przykłady w tej książce są dość podstawowe, a wyjaśnienia – bardzo proste. Rozczarowani będą również Czytelnicy mający znaczne doświadczenie w programowaniu w języku Visual Basic, którzy chcą jedynie poznać nowe funkcje programu Visual Studio 2012. Jednakże jeśli Czytelnik wcześniej nie programował lub miał przerwę w programowaniu, z pewnością doceni gruntowne wprowadzenie do pakietu Visual Studio 2012 i zakres poruszonych zagadnień dotyczących tworzenia programów Windows 8 w języku Visual Basic, zadań, które wymagają użycia wielu narzędzi i nieznanych metod.

Programiści z dużym doświadczeniem zauważą, że opisuję tu sprawy oczywiste – jednak zagadnienia oczywiste dla doświadczonych programistów nie są tak oczywiste

dla osoby, która dopiero zaczyna naukę pisania kodu. Jeśli programowanie jest dla nas nowością, jest to świetny początek.

Organizacja książki

W książce *Zacznij Tu! Poznaj Microsoft Visual Basic 2012* przyjęliśmy metodę nauki poprzez praktykę, w której Czytelnicy będą tworzyć aplikacje Windows 8 od podstaw, krok po kroku. Każdy rozdział wprowadza nowe narzędzie lub technikę, zaś książkę należy czytać rozdział po rozdziale, ponieważ każdy rozdział jest kontynuacją poprzedniego. Co prawda książka skupia się na technikach tworzenia kodu Visual Basic, jednak nauczymy się także korzystać z ciekawych narzędzi i funkcji dostępnych w środowisku Visual Studio IDE, takich jak okno Toolbox, Code Editor, kontrolki XAML, okno Solution Explorer oraz debugger. Nauczymy się również korzystać z programu Blend for Visual Studio 2012, osobnej aplikacji projektowej dystrybuowanej wraz z pakietem Visual Studio.

Dwanaście rozdziałów tej książki zebrane razem tworzą kompletny kurs wprowadzający do programowania, który można ukończyć we własnym tempie. Jeśli chcemy, możemy zrealizować jeden lub dwa rozdziały w ciągu jednego lub kilku dni, następnie przeznaczyć nieco czasu na samodzielne ćwiczenia i dopiero później przejść dalej. Przeczytanie opisu nowych technik, praktyczne wypróbowanie nowych wiadomości, a następnie samodzielne rozszerzenie projektu, to najlepszy sposób nauki wielu nowych umiejętności, włącznie z nauką programowania.

W książce poruszone zostaną następujące zagadnienia:

- **Rozdział 1: Wprowadzenie do Visual Basic 2012** Jakiego rodzaju aplikacji można utworzyć w języku Visual Basic i jak należy się za to zabrać? Ten rozdział odpowiada na te podstawowe pytania i zawiera wprowadzenie do środowiska Visual Studio IDE, elektronicznego warsztatu, w którym można budować aplikacje Visual Basic od samych podstaw. Dowiemy się, jak pobrać program Visual Studio Express 2012 dla Windows 8, jak go uruchomić i jak korzystać z narzędzi programistycznych środowiska Visual Studio.
- **Rozdział 2: Tworzenie pierwszej aplikacji dla Windows 8** W tym rozdziale nauczymy się, jak zbudować pierwszą aplikację Windows 8. Będzie to przeglądarka sieci Web, która pozwoli przeglądać strony WWW i będzie zapisywać adresy odwiedzonych stron. Poszerzymy naszą wiedzę na temat narzędzi programistycznych dostępnych w programie Visual Studio, a następnie dowiemy się, na czym polega testowanie aplikacji i przygotowanie do dystrybucji dla użytkowników.
- **Rozdział 3: Używanie kontrolki** Kontrolki, które służą do pobierania danych, wyświetlania wyników i nawigacji wewnątrz aplikacji są podstawowymi elementami interfejsu użytkownika. W tym rozdziale nauczymy się tworzyć kilka

przydatnych kontroltek XAML, takich jak *Ellipse*, *TextBlock*, *CheckBox*, *RadioButton* i *MediaElement*.

- **Rozdział 4: Projektowanie aplikacji Windows 8 przy użyciu Blend for Visual Studio** Zainstalowany przez nas pakiet Visual Studio 2012 Express zawiera osobny program zwany Blend for Visual Studio, który oferuje łatwe w użyciu narzędzia projektowe służące do tworzenia interfejsu użytkownika aplikacji Windows 8. W tym rozdziale użyjemy programu Blend do skonstruowania interfejsu użytkownika, który będzie wyświetlać zdjęcia cyfrowe i który korzysta ze scenarysów i efektów animacji. Nauczymy się też przełączać z programu Blend do Visual Studio, w którym można dodać kod programu Visual Basic.
- **Rozdział 5: Praca z XAML** Aplikacje Windows 8 wykorzystują język znaczników XAML do zdefiniowania wyglądu elementów interfejsu użytkownika na ekranie oraz sposobu prezentowania informacji. Ten rozdział zgłębia szczegółowo strukturę znaczników XAML i wyjaśnia, jak można dostosować wygląd i sposób obsługi programu poprzez zmianę kodu XAML w oknie Code Editor programu Visual Studio.
- **Rozdział 6: Elementy języka Visual Basic** Visual Basic to zaawansowany język programowania, który pozwala na kontrolę sposobu działania aplikacji Windows. Podczas tworzenia aplikacji Windows 8 używamy kodu Visual Basic do określenia sposobu zarządzania wszystkimi rodzajami informacji w aplikacji, takimi jak dane uzyskane od użytkownika oraz wyniki obliczeń matematycznych. W tym rozdziale nauczymy się składni i formatowania instrukcji programistycznych języka Visual Basic, korzystania ze zmiennych do przechowywania danych, korzystania z podstawowych typów danych i stałych oraz pracy z formułami i operatorami w programie.
- **Rozdział 7: Kontrola projektu, układu i przepływu działania programu** Aplikacje Windows 8 powinny zawierać przykuwające uwagę treści oraz strony zapewniające bogate doświadczenia użytkownikom. Ten rozdział zgłębia zasady projektowe systemu Windows 8 skupiając się na kafelkowym układzie strony oraz interakcji z użytkownikiem poprzez kontrolki *Image* i *ListBox*. Aby ułatwić sobie kontrolę nad wykonaniem i przepływem działania programu, nauczymy się pisać w aplikacji efektywne struktury decyzyjne, pętle i procedury obsługi błędów.
- **Rozdział 8: Korzystanie z platformy .NET Framework** Gdy zaczniemy pisać bardziej zaawansowane programy, przyda się nam wiedza o przetwarzaniu grafiki, wyświetlaniu plików tekstowych, dokonywaniu obliczeń i pobieraniu informacji z sieci Web. Te i wiele innych możliwości oferuje nam platforma .NET Framework, interfejs będący częścią systemu operacyjnego Windows. Ten rozdział wyjaśnia, jak poznawać klasy platformy .NET Framework w oknie Object Browser programu Visual Studio, jak używać metod platformy do przetwarzania napisów i obliczania

formuł oraz jak oszczędzać czas pracy poprzez umieszczanie w programie gotowych fragmentów kodu.

- **Rozdział 9: Debugowanie aplikacji** Złożona natura programowania w systemie Windows wiąże się z możliwością zetknięcia się z błędami syntaktycznymi i innymi błędami logicznymi, które pojawiają się od czasu do czasu podczas tworzenia aplikacji. Ten rozdział zawiera wprowadzenie do narzędzi programistycznych środowiska Visual Studio IDE, które pomagają odszukać i poprawić błędy programistyczne. Dowiemy się też, jak przewidywać błędy wykonania, z którymi mogą się w przyszłości zetknąć użytkownicy programu.
- **Rozdział 10: Zarządzanie strukturami danych przy użyciu tablic i LINQ** Ponieważ na świecie istnieje tak dużo danych – mogą to być dokumenty pracowników w biurze, informacje o cenie i produktach dostępnych w sieci Web, poufne dane o pacjencie w szpitalu – uzasadniony wydaje się fakt, że programiści spędzają mnóstwo czasu na rozważaniach dotyczących zarządzania danymi w ich programach. W pakiecie Microsoft Visual Studio do pobierania i zarządzania danymi służy ważna technologia zwana LINQ (Language Integrated Query), której podstawy poznamy w tym rozdziale. Nauczymy się przechowywać informacje w tymczasowych strukturach zwanych tablicami, tworzenia wyrażeń zapytań LINQ do pobierania danych z tablic i używać danych z dokumentów XML jako źródła dla zapytań LINQ.
- **Rozdział 11: Centrum projektu: pięć fantastycznych funkcji dla aplikacji Windows 8** W niniejszym rozdziale powrócimy do interfejsu użytkownika aplikacji Windows 8 i poszerzymy wiedzę na temat projektowania aplikacji, zgodnie z wytycznymi projektowymi firmy Microsoft dla aplikacji Windows 8. Nauczymy się tworzyć kafelki aplikacji dla strony startowej systemu Windows, tworzyć stronę startową projektu, kontrolować uprawnienia i możliwości aplikacji, używać gotowych szablonów projektów i uwzględniać wsparcie dla obsługi dotykem i gestami.
- **Rozdział 12: Dalsze możliwości rozwoju i Sklep Windows** Ten rozdział zawiera podsumowanie poznanych technik programistycznych języka Visual Basic i prezentuje dalsze możliwości rozwoju dla osób zainteresowanych rozwojem kariery na polu programowania w pakiecie Visual Studio. Rozdział ten prezentuje również szczegółowo sposoby ostatecznego testowania i pakowania aplikacji, z uwzględnieniem opisu Sklepu Windows, nowej, ekscytującej platformy dla aplikacji Windows 8. W rozdziale tym znajdziemy również listę zasobów szkoleniowych dostępnych w sieci Web oraz podręczników pozwalających na kontynuację nauki.

Bezpłatna uzupełniająca książka elektroniczna

Po zakupieniu niniejszej książki, uzyskujemy również bezpłatny dostęp do książki *Start Here! Fundamentals of Microsoft .NET Programming**. Aby uzyskać swój egzemplarz, należy przejść do strony <http://go.microsoft.com/fwlink/?Linkid=230718> i wybrać odpowiedni format (plik PDF, ePub lub Mobi, zgodnie z upodobaniami).

Książka *Fundamentals* zawiera informacje, które można wykorzystać w dowolnym języku programowania, oraz nieco specyficznego materiału przydatnego dla początkujących programistów platformy .NET.

Podczas czytania niniejszej książki natkniemy się niekiedy na odwołania do książki *Fundamentals*, które będą miały następującą postać:

Dodatkowe informacje na temat <temat> można znaleźć w uzupełniającej książce *Start Here! Fundamentals of Microsoft .NET Programming*.

Gdy znajdziemy takie odwołanie, a nie czujemy się wystarczająco obeznani z tematem, powinniśmy przeczytać sugerowany fragment książki *Fundamentals*. Ponadto książka *Fundamentals* zawiera obszerny słownik kluczowych terminów programistycznych.

Konwencje i cechy przyjęte w książce

Informacje zwarte w tej książce są zaprezentowane z wykorzystaniem określonych konwencji ułatwiających korzystanie z informacji:

- Instrukcje krok po kroku pomagające w tworzeniu aplikacji Visual Basic. Każdy zestaw instrukcji mieści się w oddzielnym podrozdziale i dokładnie opisuje, co osiągniemy po ich wykonaniu.
- Zrzuty ekranowe pokazują dokładnie wyniki uzyskane po wykonaniu instrukcji. Zrzuty są wykonane w domyślnej kolorystyce i ustawieniach systemu Windows 8, zaś rozdzielczość ekranu została ustawiona na dość niską, aby poprawić czytelność ilustracji.
- Fragmenty oznaczone ikonami i etykietami w rodzaju „Uwaga” pozwalają zdobyć dodatkowe informacje lub poznać alternatywne metody poprawnego wykonania określonego kroku. Należy zwracać szczególną uwagę na ostrzeżenia, ponieważ zawierają one przydatne informacje, które pozwolą uniknąć problemów i błędów.
- Tekst, który należy wpisać (oprócz bloków kodu), jest **pogrubiony**.
- Znak plus (+) pomiędzy dwoma nazwami klawiszy oznacza, że należy nacisnąć te klawisze jednocześnie. Na przykład zapis „Naciśnij Alt+Tab” oznacza, że najpierw należy nacisnąć i przytrzymać klawisz Alt, a następnie nacisnąć klawisz Tab.

* Pozycja ta jest dostępna tylko w języku angielskim (przyp. red. wydania polskiego).

- Ukośnik pomiędzy co najmniej dwoma elementami menu (na przykład File/Close) oznacza, że należy najpierw wybrać pierwsze menu lub element menu, później kolejny, i tak dalej.

Wymagania dotyczące systemu

Aby wykonać przykłady opisane w książce, konieczne jest spełnienie następujących wymagań:

- System operacyjny Windows 8. W zależności od ustawień systemu Windows, do zainstalowania i skonfigurowania programu Visual Studio 2012 Express może być konieczne uzyskanie uprawnień administratora.
- Do pobrania programu Visual Studio, zapoznania się ze Sklepem Windows i pobrania przykładów dołączonych do książki potrzebne jest połączenie z Internetem.
- Visual Studio 2012 Express dla Windows 8 (dokładny opis instalacji znajdziemy w rozdziale 1.).
- Komputer z procesorem 1.6 GHz lub szybszym.
- 1 GB RAM (system 32-bitowy) lub 2 GB RAM (system 64-bitowy).
- 16 GB wolnego miejsca na dysku twardym (system 32-bitowy) lub 20 GB (system 64-bitowy) dla systemu Windows 8.
- 4 GB wolnego miejsca na dysku dla programu Visual Studio 2012 Express.
- Karta graficzna zgodna z DirectX 9, ze sterownikiem WDDM w wersji 1.0 lub wyższej.
- Rozdzielczość ekranu wynosząca co najmniej 1024 × 768.
- Jeśli zamierzamy wypróbować obsługę dotykową, potrzebny nam będzie laptop, tablet lub ekran wspierający technologię wielodotykową. Windows 8 oferuje wsparcie dla co najmniej pięciu jednoczesnych punktów dotyku, choć nie wszystkie tablety czy ekrany na to pozwalają. Do wykonania ćwiczeń opisanych w niniejszej książce nie musimy posiadać urządzeń wspierających obsługę wielodotykową. Urządzenia takie przydadzą się jednak, jeśli zamierzamy zapoznać się z ich możliwościami z punktu widzenia programisty. Zazwyczaj programiści projektują oprogramowanie na komputerze stacjonarnym lub laptopie, aby następnie testować działanie obsługi wielodotykowej na urządzeniach wspierających tę technikę.

Przykładowy kod

Większość rozdziałów niniejszej książki zawiera ćwiczenia, które pozwalają zastosować w praktyce nowe umiejętności poznane w treści głównej. Wszystkie przykładowe projekty zarówno w postaci początkowej, jak i ukończonej, można pobrać z następującej strony:

<http://go.microsoft.com/fwlink/?Linkid=271576>

Aby pobrać plik *9780735672987-files.zip*, należy wykonać opisane poniżej instrukcje.

Instalowanie przykładowego kodu

Poniższe kroki opisują, jak zainstalować na komputerze przykładowy kod służący do wykonywania ćwiczeń opisanych w książce:

1. Rozpakuj plik *9780735672987-files.zip* pobrany z witryny książki (jeśli to konieczne, wpisz nową nazwę folderu, który chcesz utworzyć).
2. Jeśli na ekranie pokaże się monit o umowie licencyjnej, przejrzyj jej warunki. Aby zaakceptować postanowienia licencyjne, zaznacz opcję akceptacji i kliknij Dalej.

Używanie przykładowego kodu

Po rozpakowaniu pliku .zip z przykładowym kodem, utworzony zostanie folder „Start Here! Programming in Visual Basic” zawierający 11 podfolderów – po jednym dla każdego rozdziału niniejszej książki (oprócz ostatniego rozdziału). Aby odszukać przykłady dotyczące konkretnego rozdziału, należy otworzyć odpowiedni folder. Przykłady dla każdego rozdziału znajdują się w osobnych podfolderach. Nazwy podfolderów odpowiadają nazwom projektów w książce. Przykładowo, pliki dla projektu zwanego „Web List” znajdziemy w folderze *My Documents\Start Here! Programming in Visual Basic\Chapter 02* na twardym dysku. Jeśli system został skonfigurowany do wyświetlania rozszerzeń plików właściwych projektom języka Visual Basic, należy odszukać pliki z rozszerzeniem *.sln*. W zależności od konfiguracji komputera, zamiast folderu „My Documents”, możemy mieć do czynienia z folderem „Documents”.

Podziękowania

Pomysł napisania niniejszej książki pojawił się o wiele wcześniej niż wydano system Windows 8 i pierwsze testowe wydania pakietu Visual Studio 2012. Podczas wczesnych rozmów z wydawnictwami Microsoft Press i O'Reilly Media zdaliśmy sobie sprawę, że Windows 8 i Visual Studio 2012 prezentują prawdziwie rewolucyjne możliwości dla programistów języka Visual Basic. Pojawiło się pytanie: jak należy przygotować odpowiednie materiały szkoleniowe dla nowych i dotychczasowych programistów, aby możliwie jak najszybciej zaznajomili się z nowym materiałem i zaczęli tworzyć wspaniałe aplikacje Windows?

Zdecydowaliśmy się na napisanie dwóch oryginalnych książek. Pierwsza dotyczy nowego wydania programu Visual Studio – jest to moja książka *Zaczynij Tu! Poznaj Microsoft Visual Basic 2012*, którą Czytelnik trzyma w ręce. Drugą jest książka, również mojego autorstwa, przeznaczona dla bardziej doświadczonych programistów, zatytułowana *Microsoft Visual Basic 2012 Krok po kroku*. Te dwie książki uzupełniają się wzajemnie, tworząc wszechstronny kurs programowania dla systemu Windows 8 w języku Visual Basic 2012.

Mimo że podczas mojej kariery jako autora i programisty napisałem już ponad tuzin książek dotyczących programowania w języku Visual Basic, obecne doświadczenie okazało się najbardziej satysfakcjonujące i ekscytujące, ponieważ dwie kolejne książki wymagały znacznej koordynacji pomiędzy członkami zespołów wydawniczych zarówno w wydawnictwie Microsoft Press, jak i O'Reilly Media. Mam nadzieję, że książki przypadną Czytelnikom do gustu i że skorzystają z nich, aby dogłębnie poznać nowe fantastyczne produkty. Już niebawem będziemy się uczyć programowania w języku Visual Basic 2012 i tworzenia aplikacji przeznaczonych do dystrybucji w Sklepie Windows.

Chciałbym podziękować Devonowi Musgrave z wydawnictwa Microsoft Press, za jego wczesny entuzjazm dla tych książek oraz za skontaktowanie mnie z zespołem grupy produktu Visual Studio. W wydawnictwie O'Reilly Media szczególne podziękowania należą się głównie Russellowi Jonesowi, za nasze liczne rozmowy na temat programowania w języku Visual Basic i za podtrzymywanie nadziei, że te książki staną się przewodnikiem po sekretach języka Visual Basic 2012 dla nowych i doświadczonych programistów. Tim Patrick, utalentowany autor i programista Visual Basic, dokonał szczegółowej recenzji rękopisu *Zaczynij Tu!* i odpowiedział na wiele moich praktycznych pytań dotyczących programu Visual Studio. W obrębie grupy wydawniczej chciałbym podziękować Holly Bauer za recenzję wydawniczą i odpowiedzi na pytania dotyczące treści oraz Damonowi Larsonowi za jego biegłą adiację i rozwiązanie problemu ze stylem, który się nieoczekiwanie pojawił. Chciałbym również podziękować Kristin Borg i Rebecce Demarest z wydawnictwa O'Reilly Media oraz Lindzie Weidemann, Kimowi Burton-Weisman'owi i Kevinowi Broccoli z grupy Zyq Group, za istotne uwagi dotyczące aspektów redakcyjnych, technicznych i graficznych.

Jestem również niezmiernie wdzięczny zespołowi programistycznemu pakietu Microsoft Visual Studio 2012 za udostępnienie wersji beta i wersji RC, z którymi mogłem pracować. Ponadto chciałbym podziękować za wsparcie zespołowi systemu Microsoft Windows 8. Specjalne podziękowania kieruję do wielu uczestników forum MSDN, którzy odpowiadali na moje pytania dotyczące programowania w języku Visual Basic i systemu Windows 8.

Jak zawsze, najgłębsza wdzięczność i podziękowania należą się mojej rodzinie, za ich nieustające wsparcie podczas moich projektów pisarskich i akademickich. W szczególności pragnę podziękować Henry'emu Halvorsonowi, który utworzył rewe-lacyjny klip muzyki elektronicznej, grafikę cyfrową i plik wideo na potrzeby rozdziałów 3., 7. i 11. Synu, dziękuję Ci bardzo za wysiłek.

Errata i wsparcie dla książki

Dodażyliśmy wszelkich starań, by zapewnić wysoką jakość niniejszej książki i materiałów pomocniczych. Wszelkie błędy zgłoszone już po opublikowaniu książki są wymienione w witrynie oreilly.com w sekcji poświęconej Microsoft Press, pod adresem:

<http://go.microsoft.com/fwlink/?Linkid=271575>

Za pośrednictwem tej strony można również zgłaszać nowe błędy, zauważone przez Czytelnika. Jeśli potrzebne jest dodatkowe wsparcie, można wysłać wiadomość e-mail do zespołu Microsoft Press Book Support, na adres mspinput@microsoft.com.

Należy zwrócić uwagę, że na powyższych stronach nie znajdziemy wsparcia dla oprogramowania firmy Microsoft.

Jesteśmy ciekawi opinii Czytelników

Najwyższym priorytetem w wydawnictwie Microsoft Press jest zadowolenie klienta, zaś jego opinie są naszym najcenniejszym atutem. Poniższa strona pozwala na wyrażenie swojej opinii na temat niniejszej książki:

<http://www.microsoft.com/learning/booksurvey>

Ankieta jest krótka, zaś w wydawnictwie czytane są wszystkie przesłane opinie i pomysły. Dziękujemy z góry za wkład!

Pozostańmy w kontakcie

Niech nasza konwersacja się rozwija! Można nas znaleźć w portalu Twitter: <http://twitter.com/MicrosoftPress>.

Czytelnik może również odwiedzić stronę <http://michaelhalvorsonbooks.com>, aby dowiedzieć się więcej na temat książek i pomysłów Michaela Halvorson'a.

ROZDZIAŁ 1

Wprowadzenie do Visual Basic 2012

Po przeczytaniu tego rozdziału Czytelnik będzie umiał:

- Opisać różnorodne możliwości projektowe, jakie daje programistom język Visual Basic.
- Pobrać program Visual Studio Express 2012 i zainstalować go w systemie Windows 8.
- Uruchomić program Visual Studio Express i zapoznać się ze środowiskiem Visual Studio IDE.
- Otwierać i uruchamiać programy Visual Basic.
- Korzystać z narzędzi programistycznych i okien dostępnych w Visual Studio.
- Dostosowywać środowisko Visual Studio IDE.
- Zapisywać zmiany i zamykać program Visual Studio.

Niniejszy rozdział wprowadzi nas do programowania w języku Microsoft Visual Basic i pozwoli zdobyć umiejętności niezbędne do rozpoczęcia pracy w zintegrowanym środowisku programowania (ang. Integrated Development Environment (IDE)) Microsoft Visual Studio Express 2012 dla Windows 8. Visual Studio IDE to program służący do tworzenia i uruchamiania programów Visual Basic. Jest to środowisko obfitujące w różnorodne opcje menu, przyciski, okna narzędziowe, edytory kodu i okna wyniku. Czytelnicy mający już pewne doświadczenie z aplikacjami systemu Windows będą mieć znacznie ułatwione zadanie i nauka korzystania z IDE nie powinna im sprawić trudności. Wszyscy zorientujemy się również, że pewne narzędzia i funkcje mają większe znaczenie niż pozostałe. Postawieni wobec ogromu możliwości IDE powinniśmy uświadomić sobie, że nie ma konieczności nauki wszystkich funkcji od razu.

W niniejszym rozdziale omówione zostaną również rodzaje programów, inaczej *aplikacji*, które można utworzyć w programie Visual Basic 2012. Dawniej nie było żadnych wątpliwości, ponieważ Visual Basic umożliwiał tworzenie fantastycznych aplikacji, ale jedynie dla systemu Windows zainstalowanego na komputerach stacjonarnych. Natomiast najnowsza wersja Visual Studio pozwala programistom Visual Basic na tworzenie wielu rodzajów aplikacji o różnorodnym zastosowaniu. Niniejsza książka koncentruje się na tworzeniu aplikacji dla systemu Windows 8, jednak warto zapoznać się z pełnymi możliwościami programu Visual Studio, a szczególnie z dowolną z pełnych wersji komercyjnych.

Możliwości programistów Visual Basic

Visual Basic to język programowania zorientowany obiektowo, którego początki sięgają wcześniejszych narzędzi programistycznych, takich jak BASIC i QuickBASIC – logicznych i praktycznych (choć nieco dziwacznych) języków programowania z lat 60-tych, 70-tych i 80-tych XX wieku.

W 1991 roku Microsoft udostępnił Visual Basic 1.0, w którym w innowacyjny sposób połączono zaawansowany kompilator języka Visual Basic ze środowiskiem IDE. Pozwoliło to programistom na tworzenie aplikacji Windows poprzez rozmieszczanie kontrolki na formularzu w trybie graficznym, a następnie dostosowanie ich właściwości oraz kodu Visual Basic. W przeciągu ponad dwudziestu lat Visual Basic stał się niezwykle potężnym narzędziem programistycznym, umożliwiającym tworzenie szybkich i wydajnych aplikacji Windows, które można uruchamiać na różnorodnych platformach sprzętowych.

Termin *Visual Basic* nabrał przez ostatnie dziesięć lat dwojakiego znaczenia. W węższym znaczeniu *Visual Basic* to nazwa języka programowania, o określonej składni i procedurach logicznych, których należy używać podczas tworzenia kodu kontrolującego pewne aspekty działania aplikacji. Natomiast drugie, szersze znaczenie terminu *Visual Basic* związane jest z produktem, będącym zestawem narzędzi i technik stosowanych przez programistów podczas tworzenia aplikacji dla systemu Windows przy użyciu określonego pakietu oprogramowania. Dawniej programiści mogli nabyć osobną wersję programu Visual Basic, taką jak Microsoft Visual Basic .NET 2003 Professional. Jednak obecnie Visual Basic jest dostępny jako komponent pakietu Visual Studio, który zawiera również Microsoft Visual C#, Microsoft Visual C++, Microsoft Visual Web Developer oraz inne narzędzia projektowe.



DODATKOWE INFORMACJE Więcej informacji na temat programowania zorientowanego obiektowo można znaleźć w rozdziale 10 „Object-oriented programming” w darmowej książce uzupełniającej *Start Here! Fundamentals of .NET Programming* (Microsoft Press, 2011).

Pakiet Visual Studio 2012 jest dostępny w kilku różnych wersjach, takich jak Test Professional, Ultimate, Premium, Professional, Express dla Windows 8 oraz Express dla Web. Express dla Windows 8 oraz Express dla Web to bezpłatne wersje pakietu, które mogą służyć do testowania oprogramowania. (W niniejszej książce korzystamy z pakietu Express dla Windows 8). Pełne komercyjne wersje pakietu Visual Studio 2012 różnią się pod względem ceny oraz dostępnych funkcji. Najpełniejszą (oraz najdroższą) wersją pakietu jest wersja Ultimate. Różnice między wersjami zostały szczegółowo opisane w witrynie poświęconej Visual Studio (<http://www.microsoft.com/visualstudio/plk/>).

Czytelnicy zakupili niniejszą książkę, ponieważ pragną nauczyć się języka Visual Basic. To świetny wybór. Na całym świecie ponad 3 miliony programistów Visual Basic

tworzy innowacyjne rozwiązania, prowadzi blogi oraz kupuje dodatki i materiały szkoleniowe.

Wybór języka programowania, którego chcemy się nauczyć, zazwyczaj podyktowany jest dotychczasowym doświadczeniem oraz wymaganiami naszej firmy. Ponieważ firmy poświęciły już mnóstwo czasu oraz pieniędzy na utworzenie podstaw kodu, to mają teraz określone wymagania odnośnie języka i oprogramowania, którymi muszą posługiwać się ich podwładni. Można się z nimi zetknąć przeglądając oferty pracy dla programistów. Zwykle firmy wymagają znajomości kilku języków programowania oraz określonych umiejętności związanych z bazami danych lub z tworzeniem rozwiązań dla sieci Web.

Microsoft dokonał próby zadowolenia szerokiej rzeszy programistów i zebrał kilka technologii rozwoju oprogramowania w jeden pakiet Visual Studio, który umożliwia programowanie w językach Visual Basic, Visual C#, Visual Web Developer i JavaScript. Visual Studio zawiera również pewne podstawowe narzędzia używane przez wszystkich programistów dowolnego języka. Do narzędzi tych należą różnorodne kontrolki dostępne w przybornikach, okna Project (Projekt) i Properties (Właściwości), edytor kodu, debugger, Blend Designer, różnorodne narzędzia zarządzania oraz platforma .NET Framework – biblioteka gotowych rozwiązań przygotowana z myślą o aplikacjach działających w systemie Windows.

Nowe platformy programistyczne

Jakie są więc możliwości języka Visual Basic i pakietu Visual Studio?

Na początku XXI wieku programiści Visual Basic skupiali się przede wszystkim na aplikacjach dla systemu Windows, które usprawniały zarządzanie bazami danych w firmach. Na korzyść języka Visual Basic przemawiała możliwość przedstawiania informacji w trybie graficznym oraz udostępnienia użytkownikom dostępu do danych poprzez kontrolki interfejsu użytkownika. W ciągu ostatniej dekady język Visual Basic był wykorzystywany głównie w aplikacjach obsługujących bazy danych, systemach zarządzania zapasami, aplikacjach i narzędziach sieciowych, narzędziach zakupowych, programach typu CAD, aplikacjach naukowych oraz w grach.

Jednak niesłyszany rozwój Internetu i handlu internetowego po 2010 roku sprawiły, że świat programistów uległ kolosalnej zmianie. W przeszłości większość aplikacji Windows działała na serwerach lub na komputerach stacjonarnych. Dziś królują laptopy, tablety i smartfony. Bardzo często jedna osoba posiada urządzenia wszystkich trzech typów. Konsumenci oczekują możliwości bezproblemowego przenoszenia aplikacji i danych pomiędzy urządzeniami. Natomiast programiści potrzebują narzędzi, które pozwolą im tworzyć aplikacje działające na wielu platformach lub choćby takie, które można z łatwością zaimplementować na innym urządzeniu.

Zespół projektowy odpowiedzialny za Visual Studio 2012 podjął wyzwanie utworzenia środowiska projektowego dla różnorodnych platform tworząc pakiet, który

pozwala programistom zmniejszyć nakłady pracy, a jednocześnie umożliwia im tworzenie szerokiej gamy aplikacji różnego typu. Poniższa lista przedstawia najważniejsze możliwości i platformy rozwoju oprogramowania dostępne dla programistów języka Visual Basic (niektóre z nich są wspierane jedynie przez pełne wersje komercyjnego pakietu Visual Studio 2012):

- **Windows 8** Programiści Visual Basic mogą tworzyć aplikacje dla systemu Windows 8 oraz tradycyjne aplikacje na komputery stacjonarne, które można uruchamiać na szerokiej gamie urządzeń działających pod kontrolą systemu Windows 8, takich jak tablety Microsoft Surface, oraz sprzedawać je w Sklepie Windows.
- **Windows 7 i wcześniejsze wersje** Programiści Visual Basic mogą tworzyć aplikacje przeznaczone dla wcześniejszych wersji systemu Windows i rozpowszechniać je na szereg sposobów. Techniki programowania w językach Visual Basic i XAML (ang. Extensible Application Markup Language), które zostaną omówione w niniejszej książce, są najbardziej zbliżone do metod stosowanych przy tworzeniu programów w tych językach dla platformy Windows Presentation Foundation (WPF).
- **Windows Phone** Visual Studio wraz z pakietem Windows Phone SDK pozwalają programistom języka Visual Basic na tworzenie aplikacji przeznaczonych dla urządzenia Windows Phone, wykorzystujących pełnię jego unikatowych możliwości.
- **Aplikacje sieciowe** Programiści mogą wykorzystać Visual Basic, HTML5, CSS3, lub JavaScript przy tworzeniu aplikacji dla sieci Web wyglądających świetnie w różnorodnych przeglądarkach. Technologia znana pod nazwą ASP.NET pozwala programistom Visual Basic na szybkie tworzenie stron internetowych, aplikacji i usług sieciowych bez konieczności poznania wszystkich szczegółów dotyczących przechowywania danych w sieci.
- **Sterowniki urządzeń i aplikacje konsolowe** Programiści języka Visual Basic mogą tworzyć aplikacje przeznaczone głównie do obsługi wewnętrznych komponentów systemu operacyjnego lub działające w trybie wiersza poleceń (powłoka MS-DOS).
- **Pakiet Office** Programiści języka Visual Basic mogą tworzyć makra i inne narzędzia wzbogacające możliwości aplikacji pakietu Microsoft Office, takich jak Excel, Word, Access i PowerPoint.
- **Xbox 360** Wykorzystując Visual Studio i Microsoft XNA Game Studio programiści języka Visual Basic mogą pisać gry przeznaczone dla urządzeń Xbox.
- **Aplikacje Windows Azure przeznaczone dla serwerów sieci Web i usług działających w chmurze** Visual Basic jest wystarczająco potężny do tworzenia aplikacji używanych w zaawansowanych serwerach sieciowych, rozproszonych centrach danych i w wersji systemu Windows zaprojektowanej do działania w chmurze, znanej pod nazwą Windows Azure.

To doprawdy niezwykła lista różnorodnych typów aplikacji! Nawet jeśli na pierwszy rzut oka jest ona zniechęcająca, pocieszający niech będzie fakt, iż podstawy języka Visual Basic pozostają niezmiennie, niezależnie od platform. Ponadto szereg narzędzi i technik znacznie ułatwia przenoszenie kodu pomiędzy różnymi platformami. Niniejsza książka pozwala zdobyć solidne podstawy wielu niezbędnych umiejętności. Będą one punktem wyjścia do nauki bardziej wyspecjalizowanych technik programistycznych związanych z określonymi platformami.

Pobieranie, instalacja i uruchamianie pakietu Visual Studio Express 2012 dla Windows 8

Przed rozpoczęciem programowania w języku Visual Basic należy zainstalować pakiet Visual Studio. Posiadacze systemu Windows 8 i jednej z komercyjnych wersji pakietu Visual Studio 2012 mają wszystkie niezbędne narzędzia do nauki technik omówionych w niniejszej książce. Ci, którzy nie mają jeszcze pakietu Visual Studio, mogą pobrać darmową wersję Visual Studio Express 2012 dla Windows 8 bezpośrednio z serwisu internetowego firmy Microsoft. Po zainstalowaniu oprogramowania można już zacząć korzystać z przykładów zawartych w książce do utworzenia własnych aplikacji dla systemu Windows 8.

UWAGA Do wykonania ćwiczeń opisanych w niniejszej książce niezbędny jest system Windows 8 wraz z jedną z wersji pakietu Visual Studio 2012. Visual Studio Express 2012 dla systemu Windows 8 można pobrać za darmo, jednak do utworzenia aplikacji konieczna jest również działająca wersja systemu Windows 8.



Pobieranie produktu

Firma Microsoft udostępnia szereg produktów w wersji Express, które można pobrać z witryny <http://www.microsoft.com/express/Downloads/>, jednak na potrzeby niniejszej książki należy pobrać tylko następujący pakiet:

- **Visual Studio Express 2012 dla Windows 8** Pakiet ten zawiera środowisko projektowe Visual Studio oraz narzędzia służące do tworzenia aplikacji dla systemu Windows 8 w języku Visual Basic, Visual C#, Visual C++, lub JavaScript. System Windows 8 nie wchodzi w skład pakietu Visual Studio Express 2012, dlatego należy go wcześniej zainstalować.

W celu instalacji pakietu konieczne jest połączenie z Internetem. Pliki instalacyjne pakietu Visual Studio Express 2012 można zainstalować bezpośrednio z sieci lub zapisać na dysku w celu późniejszej instalacji.

Instalacja pakietu Visual Studio Express 2012 dla Windows 8

Poniżej opisano kolejne kroki pobierania pakietu Visual Studio Express 2012 dla Windows 8:

1. Uruchom przeglądarkę WWW (Internet Explorer lub inną) i wyświetl następującą stronę: <http://www.microsoft.com/express/Downloads>.
2. Kliknij etykietę Visual Studio 2012, aby wyświetlić listę produktów w wersji Express dostępnych dla pakietu Visual Studio 2012.



UWAGA Przed instalacją pakietu Visual Studio Express 2012 dla Windows 8 należy zainstalować na komputerze system Windows 8.

3. Kliknij Express dla Windows 8 i postępuj zgodnie z instrukcją, aby pobrać i zainstalować pakiet Visual Studio Express 2012 dla Windows 8.

Do wyboru mamy bezpośrednią instalację z sieci lub pobranie pakietu i instalację z pliku zapisanego na dysku. Można również wybrać język instalacji pakietu Visual Studio. (Na potrzeby niniejszej książki zalecany jest język angielski). Po ukończeniu instalacji można rozpocząć pracę przy użyciu pakietu Visual Studio!

Uruchamianie programu Visual Studio Express 2012

Poniższe kroki opisują, jak uruchomić program Visual Studio Express i rozpocząć pracę w środowisku programistycznym.

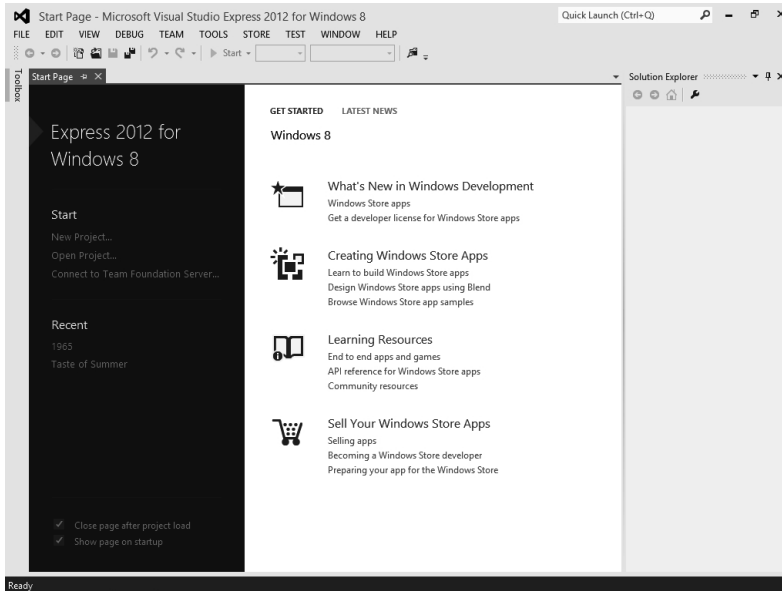
→ Uruchamianie programu Visual Studio Express 2012

1. Kliknij VS Express for Windows 8 na ekranie startowym Windows. Podczas pierwszego uruchomienia Visual Studio będziemy musieli nieco poczekać, aż program skonfiguruje środowisko. Jeśli konieczne będzie podanie licencji programisty dla Windows 8, należy utworzyć konto Windows Live lub wpisać dane dostępowe do istniejącego konta. Podczas testów beta pakietu Visual Studio 2012 licencje programistyczne były darmowe i ważne przez miesiąc. Po upływie tego czasu należało je odnowić. Prawdopodobnie będziemy musieli przejść podobną procedurę rejestracyjną.
2. Jeśli na ekranie pojawi się monit o podanie preferencji programowania, wybierz opcję Visual Basic Development Settings (Ustawienia programowania Visual Basic). Po uruchomieniu programu Visual Studio na ekranie pojawi się środowisko IDE zawierające szereg pasków menu, narzędzi i okien komponentów (okna te są czasem

zwane oknami narzędziowymi). Powinniśmy zauważyć również stronę startową (Start page) zawierającą zestaw zakładek z linkami, materiałami szkoleniowymi, wiadomościami i opcjami projektu. Strona startowa to kompleksowe źródło informacji o projekcie oraz o zasobach dostępnych dla społeczności programistów Visual Basic.

Na poniższym zrzucie ekranowym widoczny jest typowy ekran startowy programu Visual Studio. Zrzut ten został wykonany w rozdzielczości 1024x768, prawdopodobnie mniejszej niż na komputerze Czytelnika, jednak dzięki temu obraz jest wystarczająco czytelny. (Wyższa rozdzielczość bywa bardziej przydatna podczas pracy na odpowiednich monitorach, jednak nie sprawdza się w druku).

Na potrzeby tworzenia zrzutów ekranowych został wybrany motyw kolorystyczny Light (Jasny). Podczas pierwszego uruchomienia programu Visual Studio ustawiony będzie motyw kolorystyczny Dark (Ciemny), w którym biały tekst umieszczony jest na ciemnym tle. Mimo że motyw kolorystyczny Dark wygląda kojąco i dobrze wyróżnia kod oraz elementy interfejsu użytkownika w tworzonym programie, nie wygląda on zbyt dobrze w druku. Motyw kolorystyczny Dark można przełączyć na motyw Light wybierając polecenie Options (Opcje) z menu Tools (Narzędzia). Następnie należy kliknąć General (Ogólne) w kategorii Environment (Środowisko), wybrać Light (Jasny) z listy Color Theme (Motyw kolorów) i kliknąć OK. Na poniższym zrzucie ekranowym widoczny jest motyw kolorystyczny Light.



Po uruchomieniu programu Visual Studio można rozpocząć poznawanie środowiska Visual Studio IDE.



UWAGA W kolejnym podrozdziale opisano, w jaki sposób otworzyć i uruchomić program utworzony w języku Visual Basic w środowisku IDE. Jeśli dotychczas nie pobrałeś kodu dla ćwiczeń opisanych w książce, powinieneś zrobić to teraz, ponieważ niebawem będziesz musiał otworzyć jeden z programów zapisanych na twardym dysku. (Instrukcje instalacji przykładowego kodu zostały zawarte w podrozdziale „Pliki ćwiczeniowe” we Wstępie do książki). Po zainstalowaniu przykładowego kodu możesz kontynuować pracę.

Środowisko programistyczne Visual Studio

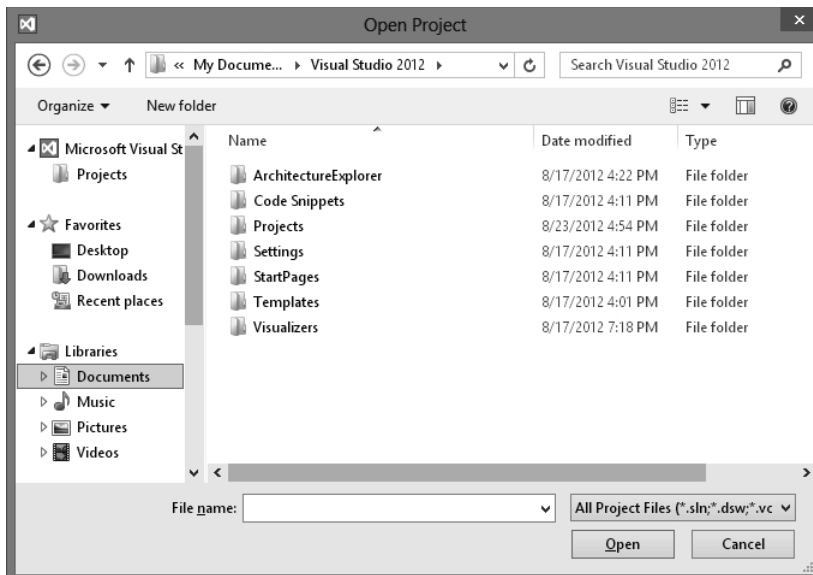
W środowisku Visual Studio IDE możemy utworzyć nowy lub istniejący projekt Visual Studio, a także przejrzeć wiele zasobów związanych z programowaniem w języku Visual Basic dostępnych online.

Otwórzmy więc, przygotowany na potrzeby tego rozdziału, projekt Visual Studio o nazwie World Capitals, który wyświetla w polu tekstowym nazwę stolicy Peru.

→ Otwieranie projektu Visual Basic

1. Kliknij łącze Open Project (Otwieranie projektu) znajdujące się z lewej strony ekranu startowego.

Na ekranie pojawi się okno dialogowe Open Project (Otwieranie projektu) pokazane na poniższym rysunku. (Okno to można wyświetlić wybierając polecenie Open Project z menu File (Plik) lub naciskając kombinację klawiszy Ctrl+Shift+O).



Nawet użytkownicy, którzy nie używali jeszcze pakietu Visual Studio, powinni sobie poradzić z obsługą okna dialogowego Open Project, ponieważ jest ono podobne do okna dialogowego Otwórz znanego z wielu innych aplikacji Windows.

WSKAZÓWKA Z lewej strony okna dialogowego Open Project widocznych jest kilka lokalizacji. Na szczególną uwagę zasługuje folder Projects znajdujący się w folderze Microsoft Visual Studio Express 2012 for Windows 8. Folder ten jest domyślną lokalizacją, w której program Visual Studio zapisuje projekty tworząc dla nich osobne podfoldery. Jednak podczas pracy z niniejszą książką będziemy zapisywać projekty w innym folderze, co zostanie niebawem opisane. Okno dialogowe pozwala również na wybór innych folderów, takich jak Ulubione czy Biblioteki, co zależy w dużym stopniu od konfiguracji komputera i systemu operacyjnego.



2. Przejdź do folderu *My Documents\Start Here! Programming in VB 2012* znajdującego się na dysku twardym.

Jest to domyślna lokalizacja obszernego zestawu przykładowego kodu przygotowanego na potrzeby niniejszej książki. Pliki będą się tam znajdować, o ile wykonane zostały instrukcje opisane w podrozdziale „Pliki ćwiczeniowe” we Wstępie do książki. W przeciwnym razie należy zamknąć okno dialogowe i skopiować pliki.

3. Otwórz folder *Chapter 01\World Capitals* i kliknij dwukrotnie plik rozwiązania *WorldCapitals* (jeśli w systemie wyświetlane są rozszerzenia nazw plików, plik będzie miał rozszerzenie *.sln*).

Visual Studio wczyta stronę *WorldCapitals*, właściwości oraz kod programu rozwiązania. W oknie Solution Explorer (Eksplorator rozwiązania), znajdującym się z prawej strony ekranu, pojawi się lista niektórych plików projektu.

W programie Visual Studio dostępna jest specjalna opcja o nazwie Always Show Solution (Zawsze pokazuj rozwiązanie), która pozwala na kontrolę kilku opcji dotyczących rozwiązania w środowisku IDE. Pole wyboru tej opcji znajduje się w zakładce Projects And Solutions/General (Projekty i rozwiązania/Ogólne) okna dialogowego Options (Opcje), które można otworzyć wybierając polecenie Options (Opcje) z menu Tools (Narzędzia). Jeśli pole opcji jest zaznaczone (ustawienie domyślne), każde rozwiązanie umieszczane jest w nowym podfolderze, zaś związany z nim projekt i pliki zostaną zapisane w osobnych podfolderach podrzędnych. Ponadto, jeśli nie zmienimy domyślnego zaznaczenia opcji Always Show Solution, w środowisku IDE pojawi się kilka dodatkowych opcji związanych z projektem, takich jak polecenia w menu File oraz element reprezentujący rozwiązanie w oknie Solution Explorer. Jeśli pomysł tworzenia oddzielnych folderów dla każdego projektu przypadnie Czytelnikowi do gustu, należy pozostawić domyślne zaznaczenie pola wspomnianej opcji. Więcej informacji na temat tych opcji można znaleźć na końcu rozdziału.

Projekty i rozwiązania

Programy tworzone przy użyciu pakietu Visual Studio nazywane są zwykle projektami (ang. project) lub rozwiązaniami (ang. solution), ponieważ nie składają się z pojedynczego pliku, lecz z wielu osobnych komponentów. Programy utworzone w Visual Basic 2012 składają się z pliku projektu (*.vbproj*), pliku rozwiązania (*.sln*), co najmniej jednego pliku znaczników (*.xaml*) oraz z kilku dodatkowych plików znajdujących się w różnych podfolderach.

Folder *project* zawiera pliki i inne informacje specyficzne dla każdego projektu. Folder *solution* zawiera wszystkie informacje dla jednego lub więcej projektu. Rozwiązania to bardzo przydatny mechanizm zarządzania kilkoma powiązаныmi projektami. Przykłady opisane w niniejszej książce zazwyczaj zawierają jeden projekt w każdym rozwiązaniu, dlatego otwarcie pliku projektu (*.vbproj*) będzie miało taki sam efekt jak otwarcie pliku rozwiązania (*.sln*). Jednak w przypadku rozwiązania składającego się z kilku projektów należy otwierać plik rozwiązania.

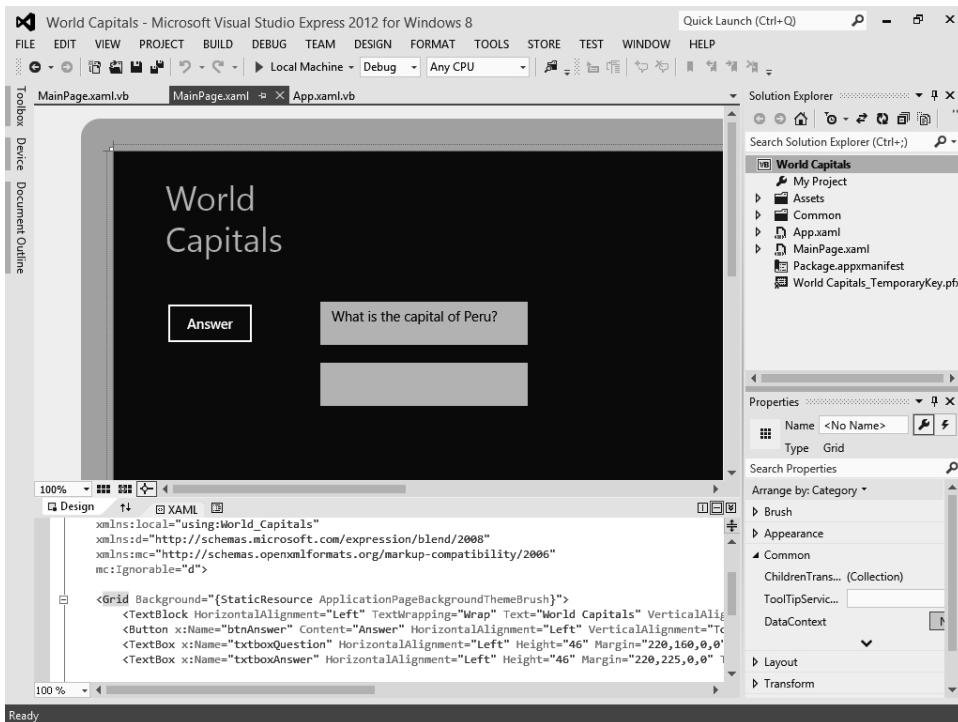
Narzędzia pakietu Visual Studio

Na tym etapie warto poświęcić kilka chwil na zapoznanie się ze środowiskiem Visual Studio IDE i odszukać kilka narzędzi programistycznych i okien, z których będziemy korzystać podczas pracy z książką. Programiści mający doświadczenie z językiem Visual Basic z pewnością rozpoznają wiele (choć prawdopodobnie nie wszystkie) z narzędzi programistycznych. Wszystkie te narzędzia służą do tworzenia, organizacji i testowania programów Visual Basic. Kilka narzędzi programistycznych ułatwia poznawanie zasobów systemowych, takich jak bazy danych czy połączenia sieci Web dostępne dla programisty. Do dyspozycji mamy również szereg potężnych narzędzi systemu pomocy.

Pasek menu pozwala na dostęp do większości poleceń kontrolujących środowisko programistyczne. Paski menu i polecenia działają tak samo jak we wszystkich programach systemu Windows i można je uruchomić klikając myszą lub używając klawiatury. Poniżej paska menu znajduje się pasek narzędzi Standard (Standardowy), czyli zestaw przycisków pełniących rolę skrótów poleceń i kontrolujących środowisko Visual Studio IDE. Zakładam, że Czytelnicy korzystali z programów Word, Excel lub innych na tyle często, by polecenia Open (Otwórz), Save (Zapisz), Cut (Wytnij) i Paste (Wklej) nie były im obce. Jednak wszyscy prawdopodobnie będą pod wrażeniem liczby i różnorodności pasków narzędziowych programu Visual Studio służących do wykonywania zadań programistycznych. W niniejszej książce opisano sposoby korzystania z niektórych pasków narzędzi. Pełną ich listę można wyświetlić w dowolnym momencie, klikając prawym przyciskiem myszy dowolny pasek w środowisku IDE.

Wzdłuż dolnej krawędzi ekranu może być widoczny pasek zadań systemu Windows. Pasek ten służy do przełączania między różnymi komponentami pakietu Visual Studio oraz do aktywacji innych programów uruchomionych w systemie Windows. Na pasku tym mogą się znajdować również ikony programu Windows Internet Explorer, narzędzi antywirusowych oraz innych programów zainstalowanych w systemie. Na większości zrzutów ekranowych znajdujących się w tej książce pasek zadań jest ukryty, co umożliwi pokazanie większej części środowiska IDE.

Poniższa ilustracja przedstawia niektóre narzędzia i okna dialogowe środowiska Visual Studio IDE. Obraz ten może się różnić od widoku własnego środowiska programistycznego, czym nie należy się przejmować. Więcej informacji na temat tych elementów (oraz sposobów dostosowywania widoku) można znaleźć w dalszej części rozdziału.



Najważniejsze narzędzia widoczne w środowisku Visual Studio IDE to Designer (Projektant), Solution Explorer, okno Properties (Właściwości) oraz zakładka XAML okna Code Editor (Edytor kodu). Należy zapoznać się z położeniem tych narzędzi oraz zapamiętać ich nazwy, ponieważ będziemy z nich często korzystać. Na ekranie mogą znajdować się również inne narzędzia, takie jak Toolbox (Przybornik), okno Document Outline (Konspekt dokumentu), okno Device (Urządzenie), Server Explorer (Eksplorator serwera) oraz Object Browser (Przeglądarka obiektów); narzędzia te mogą

być dostępne również w postaci zakładek w środowisku IDE. Ze względu na różnorodność preferencji programistów trudno przewidzieć wygląd pakietu Visual Studio, który był już wcześniej uruchamiany (zrzuty ekranowe pokazane w książce przedstawiają program tuż po zainstalowaniu, z oknem Designer wyświetlającym interfejs użytkownika projektu World Capitals, zapisany w pliku *MainPage.xaml*).

Jeśli potrzebne narzędzie nie jest widoczne, należy kliknąć menu View (Widok), a następnie wybrać właściwe narzędzie. Ponieważ menu View stopniowo się rozrastało w ciągu ostatnich lat, Microsoft przeniósł część rzadziej używanych narzędzi do podmenu Other Windows (Inne okna). Jeśli nie możemy odszukać pewnego narzędzia, powinniśmy sprawdzić wspomniane podmenu.

Powodem stwierdzenia, iż każde środowisko IDE będzie się najprawdopodobniej różnić od powyższego zrzutu, jest zależność dokładnego rozmiaru i kształtu narzędzi oraz okien od konfiguracji danego środowiska projektowego. W pakiecie Visual Studio można wyrównywać, dołączać, czy też *dokować* okna, aby widoczne były tylko potrzebne elementy. Narzędzia można również częściowo ukrywać w postaci zakładek dokumentów wzdłuż krawędzi środowiska programistycznego, a następnie szybko przełączać pomiędzy nimi. Przykładowo, po kliknięciu etykiety Toolbox z lewej strony ekranu rozwinię się panel Toolbox gotowy do użycia. Po kliknięciu innego narzędzia lub okna w środowisku IDE panel Toolbox powróci do ukrytej postaci.

Próba określenia, które narzędzia będą potrzebne natychmiast, a które nieco później, jest prawdziwym wyzwaniem dla początkującego użytkownika programu Visual Studio, dopiero zapoznającego się z jego bogatym interfejsem. Środowisko programistyczne będzie prawdopodobnie wyglądać najlepiej, gdy tak ustawimy właściwości monitora i pulpitu systemu Windows, aby zmaksymalizować ilość miejsca na ekranie. Jednak nawet wtedy ekran może się wydawać zbyt zatłoczony (w rzeczywistości niektórzy doświadczeni programiści korzystający z programu Visual Studio używają w pracy dwóch monitorów, aby wyświetlać różne widoki programu).

Celem tej mnogości narzędzi jest wzbogacenie środowiska IDE w wiele nowych i użytecznych funkcji, a jednocześnie udostępnienie inteligentnych mechanizmów zarządzania nimi. Do mechanizmów tych należy dokowanie, automatyczne ukrywanie, pływanie oraz kilka innych stanów okien, które zostaną opisane później. Visual Studio 2012 ukrywa również rzadko używane funkcje środowiska IDE, aż do chwili, gdy zostaną użyte. Pomaga to w uporządkowaniu obszaru roboczego środowiska IDE.

Początkujący użytkownicy pakietu Visual Studio powinni sobie poradzić z nadmiarem funkcji ukrywając narzędzia, które nie będą często używane, aby udostępnić więcej miejsca dla ważniejszych narzędzi. Kluczowe narzędzia dla początkujących programistów Visual Basic – te, których zaczniemy używać już niedługo – to okno Designer, Properties, Solution Explorer oraz Toolbox. Na razie nie będzie nam potrzebne okno Document Outline, Server Explorer, Class View (Widok klas), Object Browser, Device, oraz okno Debug (Debugowanie), dlatego możemy je teraz ukryć klikając przycisk Zamknij, znajdujący się na pasku tytułowym każdego okna, które należy zamknąć.

W kolejnych ćwiczeniach zaczniemy eksperymentować z najważniejszymi narzędziami środowiska Visual Studio IDE. Nauczymy się również, jak wyświetlać przeglądarkę sieci Web w programie Visual Studio oraz dowiemy się więcej o ukrywaniu niepotrzebnych narzędzi.

Okno Designer

Jeśli poprzednie ćwiczenie („Otwieranie projektu Visual Basic”) zostało ukończone, projekt World Capitals jest już wczytany do środowiska projektowego Visual Studio. Jednak interfejs użytkownika lub strona projektu może być nadal niewidoczna w programie Visual Studio (bardziej złożone projekty mogą się składać z kilku stron, jednak pierwszy przykładowy program zawiera tylko jedną). Aby wyświetlić stronę projektu World Capitals w środowisku IDE, należy posłużyć się oknem Solution Explorer.

UWAGA Jeśli projekt World Capitals nie został jeszcze otwarty, należy wrócić do podrozdziału „Otwieranie projektu Visual Basic” znajdującego się w niniejszym rozdziale i wykonać opisane w nim ćwiczenie.



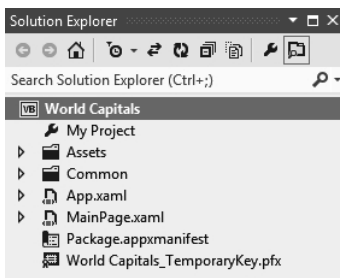
→ Wyświetlanie okna Designer

1. Odszukaj okno Solution Explorer znajdujące się w okolicach prawego górnego rogu środowiska programistycznego Visual Studio. Jeśli nie możesz znaleźć okna (lub jeśli jest ono ukryte w postaci zakładki, której nie możesz odnaleźć lub która jest niewidoczna), kliknij menu View i wybierz opcję Solution Explorer, aby wyświetlić okno.

UWAGA Począwszy od tego miejsca, przy opisie poleceń z paska menu będziemy niekiedy stosować krótszą notację. Przykładowo, zapis „Wydaj polecenie View/Solution Explorer” oznacza „Kliknij menu View i wybierz polecenie Solution Explorer”

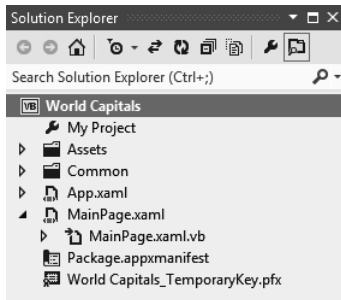


Po wczytaniu projektu World Capitals okno Solution Explorer wygląda następująco:



Podobnie jak w większości prostych aplikacji systemu Windows 8 w tym rozwiązaniu Visual Basic znajduje się plik *App.xaml*, zawierający globalne ustawienia oraz zasoby projektu; folder *Assets* (Zasoby) zawierający ekran powitalny oraz pliki logo projektu; folder *Common* (Wspólne) zawierający wspólne klasy oraz style XAML, które upraszczają zadania programistyczne; plik *manifestu wdrażania pakietu*, określający ustawienia budowania oraz dystrybucji dla pliku; oraz co najmniej jedno okno interfejsu użytkownika, zwane inaczej *stroną* (ang. page), które można z łatwością zidentyfikować poprzez rozszerzenie *.xaml*.

2. Kliknij trójkątną ikonę z lewej strony pliku *MainPage.xaml* w oknie Solution Explorer. Po rozwinięciu zawartości pliku *MainPage.xaml* okno Solution Explorer wygląda jak poniżej:



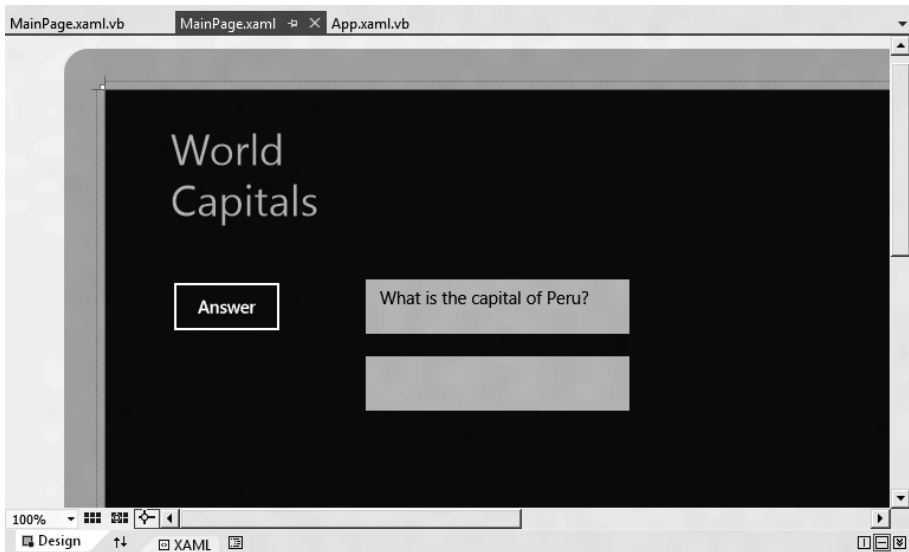
W tym projekcie układ głównej strony programu World Capitals jest zdefiniowany w pliku *MainPage.xaml* (*MainPage.xaml* jest domyślną nazwą strony głównej dla nowych aplikacji tworzonych bez specyficznego szablonu).

Plik *MainPage.xaml* możemy wyświetlić w widoku Design (Projekt), co pozwoli nam zapoznać się z interfejsem użytkownika lub zmodyfikować go korzystając z narzędzi graficznych. Można również otworzyć ten plik w oknie Code Editor, a następnie zmodyfikować interfejs użytkownika przy użyciu składni XAML, specjalnego języka opisującego interfejs użytkownika, zaprojektowanego na potrzeby aplikacji dla Windows oraz innych programów.

Poniżej pliku *MainPage.xaml* znajduje się drugi plik o nazwie *MainPage.xaml.vb*. Również ten plik jest związany z interfejsem użytkownika projektu World Capitals. *MainPage.xaml.vb* jest nazywany *plikiem związanym z kodem*, ponieważ zawiera kod programu Visual Basic obsługujący zachowanie elementów interfejsu użytkownika zdefiniowanych w pliku *MainPage.xaml*. W miarę postępu nauki programowania w języku Visual Basic nabierzemy biegłości w dostosowywaniu tego pliku.

Solution Explorer to brama prowadząca do różnorodnych plików wchodzących w skład projektu – jest to niezbędne narzędzie pracy. Plik kliknięty dwukrotnie w oknie Solution Explorer otworzy się w odpowiednim edytorze, o ile możliwa jest jego bezpośrednia edycja.

3. Jeśli interfejs użytkownika projektu nie jest widoczny w oknie Designer, kliknij dwukrotnie plik w oknie Solution Explorer, aby go wyświetlić. Jeśli to konieczne, użyj poziomego paska przewijania, aby dopasować widok interfejsu użytkownika. W oknie Designer zostanie wyświetlona strona World Capitals, tak jak na poniższej ilustracji:

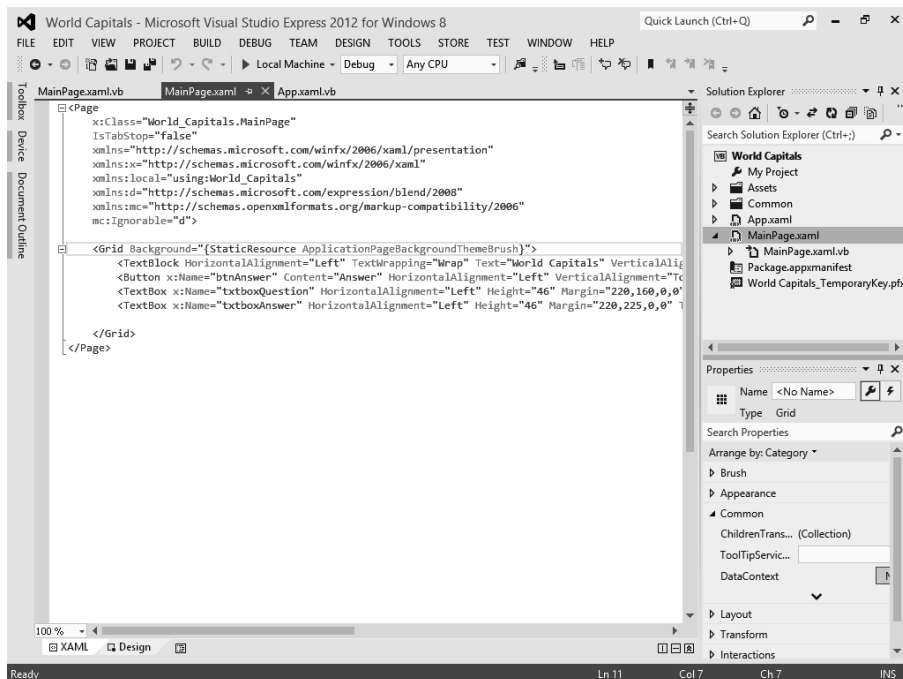


Zwróćmy uwagę na zakładkę z nazwą pliku *MainPage.xaml* widoczną w górnej części okna Designer, obok dodatkowych zakładek. W dowolnym momencie można kliknąć jedną z zakładek, aby wyświetlić zawartość odpowiedniego pliku, nawet jeśli jego okno jest zakryte.

Jak wcześniej wspomniano, plik *MainPage.xaml* jest wizualną reprezentacją interfejsu użytkownika programu. Jednakże można też przejrzeć kod XAML służący do opisu interfejsu użytkownika. W tym celu należy kliknąć dwukrotnie zakładkę XAML okna Code Editor w dolnej części okna Designer. Jeśli zakładka XAML jest już otwarta w oknie Code Editor, możemy przejrzeć kod XAML opisujący interfejs użytkownika i używając pasków przewijania wyświetlić dowolny fragmentu kodu, który nie jest widoczny.

4. Kliknij dwukrotnie zakładkę XAML, aby wyświetlić kod XAML dla strony w oknie Code Editor. Przewiń zawartość okna w górę, aby przejrzeć cały dokument.

Ekran powinien wyglądać następująco:



W oknie Code Editor wyświetlony jest kod XAML pliku `MainPage.xaml`, który zawiera ustrukturyzowane informacje kontrolujące sposób wyświetlania interfejsu użytkownika i grafiki aplikacji w Visual Studio oraz w systemie Windows. Dla programistów znających choć trochę język HTML ten plik powinien wyglądać dość znajomo. Składnię XAML tworzą *znaczniki* – instrukcje, których głównym zadaniem jest przekazanie programowi informacji o sposobie wyświetlania elementów na ekranie. Informacje XAML pokazane na ilustracji znajdują się pomiędzy znacznikami `<Page>` i `</Page>`. Ich struktura została skomponowana w taki sposób, aby zapewnić czytelność informacji. Siedem pierwszych wierszy poniżej znacznika `<Page>` definiuje zasoby użyte do utworzenia interfejsu użytkownika. Znajdująca się poniżej sekcja umieszczona pomiędzy znacznikami `<Grid>` i `</Grid>` definiuje obiekty tworzące interfejs użytkownika. W tym przypadku plik XAML zawiera definicję jednego przycisku, dwóch pól tekstowych oraz jednego bloku tekstowego. Wygląd tych elementów można sprawdzić na ilustracji przedstawiającej okno Designer. Plik pozwala również na sprawdzenie wartości określonych właściwości obiektów przypisanych poprzez nazwy właściwości (takie jak `HorizontalAlignment` (Wyrównanie poziome)) oraz wartości (takie jak `Left` (Do lewej)).

Więcej informacji na temat języka znaczników XAML można znaleźć w kolejnych rozdziałach. Na razie wystarczy zapamiętać, że w oknie Designer można wyświetlić *podgląd* interfejsu użytkownika oraz zawartość pliku XAML definiującego

specyficzne cechy obiektów widocznych na stronie podglądu. Programiści korzystający z programu Visual Studio zwykle pracują z obydwoma widokami ułożonymi obok siebie. Osoby, które mają doświadczenie w tworzeniu aplikacji HTML, z pewnością zauważą podobieństwa, ponieważ wiele narzędzi służących projektowaniu stron WWW również wyświetla układ strony w górnej części ekranu, zaś kod HTML w dolnej.

WSKAZÓWKA W dolnej części okna Designer oraz Code Editor znajduje się seria przydatnych przycisków, które pozwalają na sporą kontrolę sposobu podziału ekranu. W lewej dolnej części okna Designer znajdują się zakładki XAML i Design oraz przydatny przycisk Document Outline, służący do otwarcia osobnego okna, w którym wyświetlane są obiekty interfejsu użytkownika ułożone według typu. W prawej dolnej części okna Designer znajdują się przyciski Vertical Split (Podział pionowy), Horizontal Split (Podział poziomy), oraz Expand Pane/Collapse Pane (Rozwiń okienka/Zwiń okienka), które określają wzajemne ułożenie okien Designer oraz Code Editor. Szczególnie przydatny jest przycisk Expand Pane/Collapse; jest to przełącznik pozwalający na wyświetlenie tylko jednego okna lub dwóch okien obok siebie.



5. Kliknij zakładkę Design, aby ponownie wyświetlić główną stronę projektu w oknie Designer.
6. Kliknij przycisk Expand Pane, aby wyświetlić kod znaczników XAML, odpowiedzialny za wyświetlanie strony w oknie poniżej okna Designer.

Następnie spróbujemy uruchomić ten prosty program w Visual Studio.

Uruchamianie programu Visual Basic

World Capitals to prosty program Visual Basic, zaprojektowany w celu zaznajomienia czytelnika z narzędziami programistycznymi Visual Studio. Na stronie widocznej na ekranie umieszczone zostały cztery obiekty (jeden przycisk, blok tekstowy oraz dwa pola tekstowe). W pliku związanym z kodem umieszczono jeden wiersz kodu, odpowiadający za zadawanie użytkownikowi pytania oraz za wyświetlanie właściwej odpowiedzi. Więcej informacji na temat tworzenia tego typu obiektów oraz na temat dodawania do nich kodu Visual Basic można znaleźć w rozdziale 2. „Tworzenie pierwszej aplikacji dla Windows 8”. Na razie spróbujemy uruchomić program w środowisku Visual Studio IDE.

→ Uruchamianie programu World Capitals

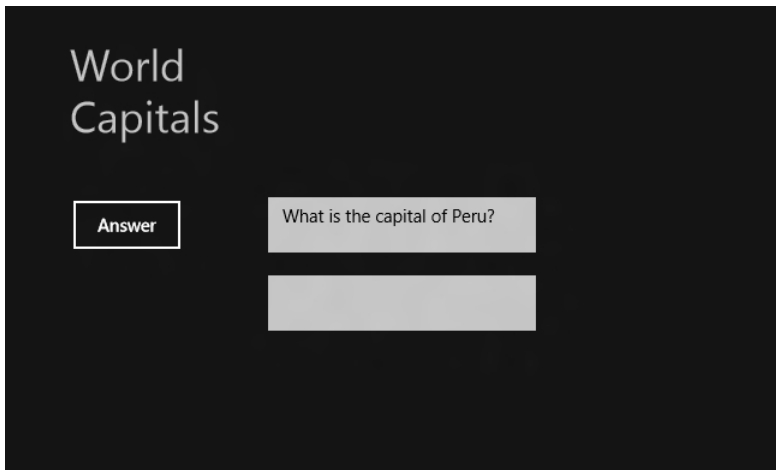
1. Kliknij przycisk Start (strzałka w prawo obok napisu *Local Machine*) na pasku narzędzi Standard, aby uruchomić program World Capitals w Visual Studio.



Wskazówka Aby uruchomić program w środowisku Visual Studio IDE, można również nacisnąć klawisz F5 lub wybrać polecenie Start Debugging z menu Debug.

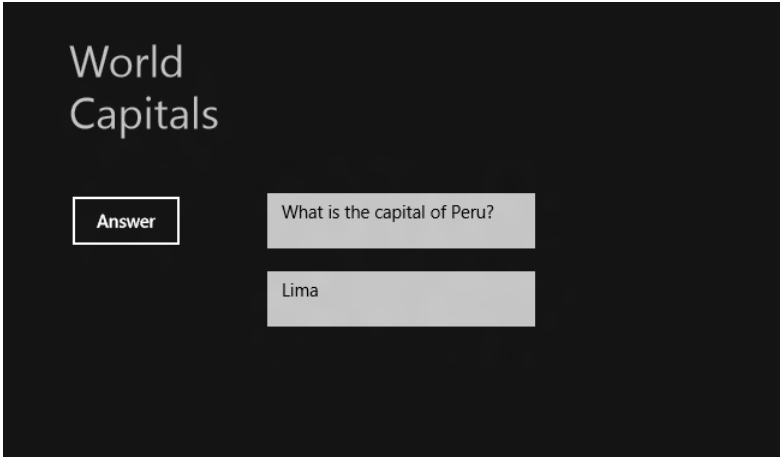
Visual Studio wczyta i skompiluje projekt do postaci *podzespołu*, pliku EXE, zawierającego dane i kod w postaci gotowej do użycia przez komputer. Ten konkretny plik zawiera również informacje przydatne podczas testowania lub *debugowania*, które jest fundamentalną częścią procesu rozwoju oprogramowania. Jeśli kompilacja się powiedzie, Visual Studio uruchomi program w środowisku IDE (ten sposób zwany jest uruchamianiem programu na *maszynie lokalnej*, w odróżnieniu od uruchamiania na komputerze zdalnym znajdującym się w sieci lub na emulatorze dowolnego rodzaju sprzętu).

W trakcie działania programu, na pasku zadań system Windows pojawia się jego ikona. Po chwili zobaczymy interfejs użytkownika programu World Capitals działający podobnie jak każda aplikacja w systemie Windows 8. Wygląd programu jest zgodny z podglądem wyświetlonym w oknie Designer programu Visual Studio:



Program World Capitals zadaje nam pytanie: What is the capital of Peru?

2. Kliknij przycisk Answer, aby wyświetlić odpowiedź na pytanie. Program powinien wyświetlić odpowiedź (Lima) poniżej pytania.



3. Zamknij aplikację przeciągając pasek tytułowy (lub górną część ekranu) w dolną część ekranu (lub w dowolny sposób, w jaki zamykasz aplikacje Windows 8).

Gdy przesuniemy kursor myszy nad górną krawędź ekranu, ikona kursora przybiera postać dłoni, co potwierdza czynność przeciągania paska tytułowego w dolną część ekranu w celu zamknięcia programu. Po zamknięciu aplikacji można nacisnąć klawisz Windows lub kliknąć ikonę programu Visual Studio znajdującą się na pulpicie, aby powrócić do środowiska IDE.

Aplikacja World Capitals może jeszcze przez chwilę działać, podczas gdy środowisko Visual Studio IDE przetwarza żądanie zakończenia programu (na pasku tytułowym programu Visual Studio może być nadal widoczny napis *Running*, oznaczający, że program wczytany do środowiska IDE jest nadal uruchomiony). Klikając przycisk Stop Debugging (Zatrzymaj debugowanie) możemy natychmiast zatrzymać każdą aplikację uruchomioną w środowisku Visual Studio IDE.

Po zatrzymaniu programu w środowisku IDE zajdzie kilka zmian. W dolnej części środowiska IDE pojawi się okno Output (Dane wyjściowe) zawierające informacje o przebiegu kompilacji i uruchomienia podzespołów aplikacji. Tego właśnie oczekujemy od programu Visual Studio po skompilowaniu i uruchomieniu aplikacji. Okno Output zawiera dość szczegółowy opis przebiegu kompilacji, kilkietapowego procesu, obejmującego wczytywanie szeregu plików i zasobów zwanych *bibliotekami*. Ten zapis przebiegu procesu jest szczególnie przydatny w przypadku niepowodzenia kompilacji z powodu niedostrzeżonej pomyłki programisty lub innego błędu (z pewnością zetknie się z tym każdy programista, chociaż jeszcze nie teraz!).

4. Po zapoznaniu się z zawartością okna Output zamknij je, klikając jego przycisk Zamknij.

W początkowych rozdziałach tej książki nie będziemy przywiązywać dużej wagi do okna Output, jednak w przypadku wystąpienia niezamierzonego błędu podczas pisania własnych programów narzędzie to okaże się bardzo przydatne. Jednak zazwyczaj można po prostu zamknąć to okno, aby uzyskać więcej miejsca na pracę z kodem.

Nadszedł czas, aby zapoznać się z kolejnym przydatnym narzędziem.

Okno Properties

W celu zmiany cech lub ustawień właściwości jednego lub więcej elementów interfejsu użytkownika na stronie można skorzystać z okna Properties dostępnego w środowisku IDE. *Ustawienie właściwości* to pewna cecha każdego z obiektów wchodzących w skład programu, taka jak położenie na ekranie, rozmiar, wyświetlany w nim tekst i inne. Przykładowo, możemy odpowiednio dostosować ustawienie właściwości, aby zmodyfikować wygląd pytania o stolice, tak aby napis był wyświetlany z zastosowaniem innego kroju lub wielkości czcionki (Visual Studio, podobnie jak program Excel i Word, pozwala na wyświetlenie napisu w dowolnej czcionce zainstalowanej w systemie).

Okno Properties zawiera listę właściwości obiektu zaznaczonego w oknie Designer. Przykładowo, jeśli w oknie Designer zaznaczony jest przycisk, w oknie Properties wyświetlone będą właściwości obiektu przycisku. Pierwsza właściwość znajdująca się w górnej części okna Properties to *Name* (Nazwa), której będziemy używać do nadawania nazw obiektom, jeśli zamierzamy dostosowywać je do własnych potrzeb przy użyciu kodu Visual Basic (domyślnie nowe obiekty XAML nie są nazywane). Każdy obiekt na stronie ma wiele właściwości, jednak Visual Studio przypisuje większości z nich domyślne wartości, które można z łatwością odszukać zmieniając ich kolejność przy użyciu listy Arrange By (Rozmieść według) w górnej części okna Properties.

Ustawienie właściwości można zmienić w oknie Properties podczas pracy na stronie w trybie graficznym, edytując kod XAML dla strony lub dodając kod Visual Basic do pliku związanego z kodem, aby poinstruować Windows, w jaki sposób należy zmienić jedno lub więcej ustawień właściwości w trakcie działania programu.

Możemy również dostosować procedury obsługi zdarzeń dla obiektów na stronie, korzystając z przycisku Event Handlers (Programy obsługi zdarzeń) (który wygląda jak niebieski piorun), znajdujący się w górnej części okna Properties. Wrócimy do tego w dalszej części książki. *Programy obsługi zdarzeń* (ang. *Event handlers*) to niestandardowe procedury Visual Basic, które są uruchamiane w momencie interakcji użytkownika z obiektami na stronie poprzez klikanie, pukanie, przeciąganie i inne.

Zdobądźmy nieco doświadczenia w zmianie ustawień i zajmijmy się edycją tekstu obiektu przycisku. Zmienimy też czcionkę bloku tekstowego na pogrubioną i pochyloną.