

Spis treści

Wstęp	xiii
Część I Wprowadzenie do języka Microsoft Visual C# oraz Microsoft Visual Studio 2010	
1 Wprowadzenie do języka C#	3
Rozpoczynamy programowanie przy użyciu środowiska Visual Studio 2010	4
Piszemy pierwszy program	9
Przestrzenie nazw	15
Tworzenie aplikacji graficznej	19
Krótkie podsumowanie rozdziału 1	30
2 Zmienne, operatory i wyrażenia	31
Instrukcje	31
Identyfikatory	32
Słowa kluczowe	33
Zmienne	34
Nazywanie zmiennych	34
Deklarowanie zmiennych	35
Podstawowe typy danych	36
Zmienne lokalne bez przypisanej wartości	36
Wyświetlanie wartości podstawowych typów danych	37
Posługiwanie się operatorami arytmetycznymi	42
Operatory i typy danych	42
Poznajemy operatory arytmetyczne	44
Kontrolowanie pierwszeństwa	47
Stosowanie zasad łączności przy wyznaczaniu wartości wyrażen	48
Zasady łączności a operator przypisania	48
Inkrementacja i dekrementacja wartości zmiennych	50
Formy przyrostkowe i przedrostkowe	50
Deklarowanie zmiennych lokalnych o niejawnie określonym typie danych	51
Krótkie podsumowanie rozdziału 2	53
3 Tworzenie metod i stosowanie zasięgów zmiennych	55
Tworzenie metod	55
Deklarowanie metody	56
Zwracanie danych przez metodę	57
Wywoływanie metod	59
Określanie składni wywołania metody	59

Stosowanie zasięgu	62
Definiowanie zasięgu lokalnego	62
Definiowanie zasięgu klasy	63
Przeciążanie metod	63
Tworzenie metod	64
Stosowanie parametrów opcjonalnych oraz nazwanych argumentów	73
Definiowanie parametrów opcjonalnych	74
Przekazywanie nazwanych argumentów	75
Rozwiązywanie niejednoznaczności związanych z parametrami opcjonalnymi i argumentami nazwanymi	76
Krótkie podsumowanie rozdziału 3	81
4 Instrukcje wyboru	83
Deklarowanie zmiennych logicznych	83
Stosowanie operatorów logicznych	84
Operatory równościowe oraz operatory relacji	84
Warunkowe operatory logiczne	85
Skracanie działania	86
Podsumowanie informacji o pierwszeństwie oraz łączności operatorów	87
Podejmowanie decyzji przy użyciu instrukcji <i>if</i>	88
Składnia instrukcji <i>if</i>	88
Grupowanie instrukcji w bloki	89
Kaskadowe łączenie instrukcji <i>if</i>	90
Stosowanie instrukcji <i>switch</i>	96
Składnia instrukcji <i>switch</i>	96
Reguły stosowania instrukcji <i>switch</i>	97
Krótkie podsumowanie rozdziału 4	101
5 Złożone instrukcje przypisania oraz instrukcje iteracji	103
Złożone operatory przypisania	103
Instrukcja <i>while</i>	105
Instrukcja <i>for</i>	110
Zasięg instrukcji <i>for</i>	111
Instrukcja <i>do</i>	112
Krótkie podsumowanie rozdziału 5	122
6 Obsługa błędów i wyjątków	123
Zmaganie się z błędami	124
Wypróbowywanie kodu i przechwytywanie wyjątków	124
Nieobsłużone wyjątki	126
Stosowanie kilku bloków obsługi pułapki	127
Przechwytywanie wielu wyjątków	128
Wykonywanie operacji arytmetycznych z kontrolą lub bez kontroli przepełnienia	133
Pisanie instrukcji objętych kontrolą przepełnienia	134
Pisanie wyrażeń objętych kontrolą przepełnienia	135
Zgłaszanie wyjątków	137

Stosowanie bloku <i>finally</i>	141
Krótkie podsumowanie rozdziału 6	143

Część II Język C#

7 Tworzenie i zarządzanie klasami oraz obiektami	147
Omówienie klasyfikacji	148
Cele hermetyzacji	148
Definiowanie i używanie klas	149
Kontrolowanie dostępności	150
Konstruktory	152
Przeciążanie konstruktorów	153
Metody i dane statyczne	161
Tworzenie pól współdzielonych	162
Tworzenie pól statycznych przy użyciu słowa kluczowego <i>const</i>	163
Klasy statyczne	163
Klasy anonimowe	167
Krótkie podsumowanie rozdziału 7	169
8 Wartości i referencje	171
Kopiowanie klas oraz zmiennych typu wartościowego	171
Wartości Null oraz typy danych dopuszczające stosowanie wartości Null	177
Typy danych dopuszczające stosowanie wartości Null	178
Właściwości typów danych dopuszczających stosowanie wartości Null	179
Używanie parametrów typu <i>ref</i> i <i>out</i>	180
Tworzenie parametrów typu <i>ref</i>	181
Tworzenie parametrów typu <i>out</i>	182
Sposób organizacji pamięci komputera	184
Korzystanie ze stosu oraz ze sterty	185
Klasa <i>System.Object</i>	186
Opakowywanie typów danych wewnątrz obiektów	187
Rozpakowywanie typów danych, opakowanych wewnątrz obiektów	188
Bezpieczne rzutowanie danych	190
Operator <i>is</i>	190
Operator <i>as</i>	191
Krótkie podsumowanie rozdziału 8	195
9 Tworzenie typów wartości przy użyciu wyliczeń oraz struktur	197
Wyliczeniowe typy danych	197
Deklarowanie wyliczeniowego typu danych	198
Stosowanie wyliczeniowych typów danych	198
Wybór wartości literałów wyliczeniowych	199
Wybór typu danych używanego do wewnętrznego reprezentowania wartości wyliczeniowych	200

Struktury	203
Deklarowanie struktury	204
Omówienie różnicy pomiędzy strukturami i klasami	205
Deklarowanie zmiennych strukturalnych	207
Omówienie inicjalizacji struktur	208
Kopiowanie zmiennych strukturalnych	212
Krótkie podsumowanie rozdziału 9	216
10 Tablice i kolekcje	217
Czym jest tablica?	217
Deklarowanie zmiennych tablicowych	218
Tworzenie instancji tabeli	218
Inicjalizowanie zmiennych tablicowych	219
Tworzenie tablic o niejawnie określonym typie elementów	220
Korzystanie z indywidualnych elementów tablicy	221
Wykonywanie iteracji poprzez elementy tablicy	222
Kopiowanie tablic	224
Tablice wielowymiarowe	225
Wykorzystanie tablic do gry w karty	226
Co to są klasy kolekcji?	233
Klasa kolekcji <i>ArrayList</i>	235
Klasa kolekcji <i>Queue</i>	237
Klasa kolekcji <i>Stack</i>	238
Klasa kolekcji <i>Hashtable</i>	239
Klasa kolekcji <i>SortedList</i>	240
Inicjalizowanie kolekcji	241
Porównanie tablic i kolekcji	242
Wykorzystanie klas kolekcji do gry w karty	242
Krótkie podsumowanie rozdziału 10	246
11 Tablice parametrów	247
Używanie argumentów będących tablicami	248
Deklarowanie tablicy parametrów typu <i>params</i>	249
Używanie parametru typu <i>params object[]</i>	251
Stosowanie tablicy parametrów typu <i>params</i>	253
Porównanie tablic parametrów z parametrami opcjonalnymi	256
Krótkie podsumowanie rozdziału 11	258
12 Dziedziczenie	259
Czym jest dziedziczenie?	259
Korzystanie z mechanizmów dziedziczenia	260
Wywoływanie konstruktora klasy bazowej	262
Przypisywanie klas	263
Deklarowanie metod z użyciem słowa kluczowego <i>new</i>	265
Deklarowanie metod wirtualnych	267
Deklarowanie metod z użyciem słowa kluczowego <i>override</i>	268
Omówienie dostępu chronionego	271
Metody rozszerzające	277

Krótkie podsumowanie rozdziału 12	281
13 Tworzenie interfejsów oraz definiowanie klas abstrakcyjnych	283
Interfejsy	283
Definiowanie interfejsu	285
Implementowanie interfejsu	285
Odwoływanie się do klasy za pomocą jej interfejsu	286
Praca z wieloma interfejsami	287
Jawne implementowanie interfejsu	287
Ograniczenia interfejsu	290
Definiowanie i używanie interfejsów	290
Klasy abstrakcyjne	300
Metody abstrakcyjne	301
Klasy zamknięte	302
Metody zamknięte	302
Implementowanie i używanie klas abstrakcyjnych	303
Krótkie podsumowanie rozdziału 13	308
14 Proces oczyszczania pamięci i zarządzanie zasobami	309
Żywot obiektu	309
Tworzenie destruktorów	311
Dlaczego istnieje proces oczyszczania pamięci?	312
Sposób działania procesu oczyszczania pamięci	314
Zalecenia	315
Zarządzanie zasobami	315
Metody sprzątające	316
Sprzątanie w sposób odporny na występowanie wyjątków	316
Instrukcja <i>using</i>	317
Wywoływanie metody <i>Dispose</i> z poziomu destruktora	319
Implementacja metody sprzątającej w sposób odporny na występowanie wyjątków	321
Krótkie podsumowanie rozdziału 14	324

Część III Tworzenie komponentów

15 Implementacja właściwości w celu dostępu do pól	327
Implementacja kapsułkowania za pomocą metod	328
Co to są właściwości?	330
Używanie właściwości	331
Właściwości tylko do odczytu	332
Właściwości tylko do zapisu	332
Dostępność właściwości	333
Ograniczenia właściwości	334
Deklaracja właściwości interfejsu	336
Używanie właściwości w aplikacji Windows	337
Generowanie automatycznych właściwości	339

Inicjalizacja obiektów przez użycie właściwości	340
Krótkie podsumowanie rozdziału 15	345
16 Indeksatory	347
Co to jest indeksator?	347
Przykład bez użycia indeksatorów	348
Ten sam przykład z wykorzystaniem indeksatorów	349
Akcesory indeksatora	351
Porównanie indeksatorów i tablic	352
Indeksatory w interfejsach	354
Korzystanie z indeksatorów w aplikacji Windows	355
Krótkie podsumowanie rozdziału 16	361
17 Przerwanie działania programu oraz obsługa zdarzeń	363
Deklarowanie i używanie delegatów	364
Przykład zautomatyzowanej fabryki	364
Implementacja fabryki bez używania delegatów	365
Implementacja fabryki z wykorzystaniem delegata	365
Korzystanie z delegatów	368
Wyrażenia lambda a delegaty	373
Tworzenie adaptera metody	373
Korzystanie z wyrażenia lambda jako adaptera	374
Postać wyrażen lambda	374
Włączanie powiadomień za pomocą zdarzeń	376
Deklarowanie zdarzenia	377
Subskrypcja zdarzenia	377
Odwołanie subskrypcji zdarzenia	378
Wywoływanie zdarzenia	378
Zdarzenia interfejsu użytkownika WPF	379
Używanie zdarzeń	380
Krótkie podsumowanie rozdziału 17	384
18 Typy ogólne	387
Problem z typem <i>object</i>	387
Rozwiązanie za pomocą typów ogólnych	389
Typy ogólne a klasy uogólnione	391
Typy ogólne a ograniczenia	392
Tworzenie klasy ogólnej	392
Teoria drzew binarnych	392
Budowa klasy drzewa binarnego przy użyciu typów ogólnych	395
Tworzenie metody ogólnej	404
Definiowanie metody ogólnej do budowy drzewa binarnego	405
Wariancja i interfejsy ogólne	407
Interfejsy kowariantne	409
Interfejsy kontrawariantne	410
Krótkie podsumowanie rozdziału 18	413

19 Wyliczanie kolekcji	415
Wyliczanie elementów kolekcji	415
Ręczna implementacja modułu wyliczającego	417
Implementacja interfejsu <i>IEnumerable</i>	421
Implementacja modułu wyliczeniowego z wykorzystaniem iteratora ..	423
Prosty iterator	423
Definiowanie modułu wyliczającego dla klasy <i>Tree<TItem></i> z wykorzystaniem iteratora	425
Krótkie podsumowanie rozdziału 19	428
20 Odpytywanie danych znajdujących się w pamięci przy użyciu wyrażeń w języku zapytań	429
Co to jest Language Integrated Query?	429
Używanie LINQ w aplikacji C#	430
Wybieranie danych	432
Filtrowanie danych	434
Porządkowanie, grupowanie i agregowanie danych	435
Łączenie danych	437
Korzystanie z operatorów zapytań	439
Odpytywanie danych w obiektach <i>Tree<TItem></i>	441
LINQ i odroczone ewaluacja	446
Krótkie podsumowanie rozdziału 20	449
21 Przeciążanie operatorów	451
Pojęcie operatorów	451
Ograniczenia operatorów	452
Operatory przeciążone	452
Tworzenie operatorów symetrycznych	454
Złożone przypisanie z obliczaniem	456
Deklarowanie operatorów zwiększających i zmniejszających	457
Porównanie operatorów w strukturach i klasach	458
Definiowanie par operatorów	459
Implementacja operatorów	460
Operatory konwersji	466
Świadczenie wbudowanej konwersji	466
Implementacja definiowanych przez użytkownika operatorów konwersji	467
Tworzenie operatorów symetrycznych – uzupełnienie	468
Pisanie operatorów konwersji	469
Krótkie podsumowanie rozdziału 21	472

Część IV Budowanie aplikacji Windows Presentation Foundation

22 Omówienie platformy Windows Presentation Foundation	475
Tworzenie aplikacji WPF	476
Budowanie aplikacji WPF	476

Dodawanie kontrolek do formularza	490
Korzystanie z kontrolek WPF	490
Dynamiczne zmienianie właściwości	498
Obsługa zdarzeń w formularzu WPF	502
Przetwarzanie zdarzeń w formularzach Windows	502
Krótkie podsumowanie rozdziału 22	507
23 Interakcja z użytkownikiem	509
Wskazówki i style menu	510
Menu i zdarzenia menu	510
Tworzenie menu	511
Obsługiwanie zdarzeń menu	517
Menu podręczne	522
Tworzenie menu podręcznych	523
Typowe okna dialogowe systemu Windows	526
Korzystanie z klasy <i>SaveFileDialog</i>	527
Zwiększanie interakcyjności w aplikacji WPF	530
Krótkie podsumowanie rozdziału 23	539
24 Wykonywanie walidacji	541
Walidacja danych	541
Strategie sprawdzania danych użytkownika	542
Przykład – Zamawianie biletów na imprezy	542
Walidacja przy użyciu wiązania danych	543
Zmiana momentu przeprowadzania walidacji	559
Krótkie podsumowanie rozdziału 24	563

Część V Zarządzanie danymi

25 Zapytania do bazy danych	567
Wprowadzanie zapytań do bazy danych za pomocą ADO.NET	568
Baza danych Northwind	568
Tworzenie bazy danych	569
Korzystanie z ADO.NET do tworzenia zapytań dotyczących informacji o zamówieniach	570
Zapytania do bazy danych za pomocą LINQ to SQL	581
Definiowanie klasy encji	581
Tworzenie i uruchamianie zapytania LINQ to SQL	583
Opóźnione i natychmiastowe pobieranie danych	585
Łączenie tabel i tworzenie relacji	586
Odroczone i natychmiastowe pozyskiwanie danych, uzupełnienie	590
Definiowanie własnej klasy <i>DataContext</i>	590
Korzystanie z LINQ to SQL do odczytywania informacji o zamówieniach	591
Krótkie podsumowanie rozdziału 25	596

26 Wyświetlanie i edycja danych przy użyciu Entity Framework i wiązania danych	597
Stosowanie wiązania danych z Entity Framework	598
Używanie wiązania danych do modyfikacji danych	615
Aktualizacja istniejących danych	615
Obsługa konfliktów aktualizacji	616
Dodawanie i usuwanie danych	619
Krótkie podsumowanie rozdziału 26	628
Część VI Tworzenie profesjonalnych rozwiązań przy użyciu Visual Studio 2010	
27 Biblioteka równoległego realizowania zadań	631
Po co stosować wielozadaniowość przy użyciu przetwarzania równoległego?	632
Era procesorów wielordzeniowych	634
Implementowanie wielozadaniowości w aplikacjach desktopowych	635
Zadania, wątki i klasa <i>ThreadPool</i>	636
Tworzenie, uruchamianie i kontrolowanie zadań	638
Implementowanie równoległego działania przy użyciu klasy <i>Task</i>	642
Tworzenie abstrakcji zadań przy użyciu klasy <i>Parallel</i>	651
Zwracanie wartości z poziomu zadania	659
Używanie zadań w połączeniu z interfejsem użytkownika	664
Anulowanie zadań i obsługiwane wyjątków	668
Mechanizm kooperatywnego anulowania	668
Obsługiwanie wyjątków zadania za pomocą klasy <i>AggregateException</i>	678
Używanie kontynuacji dla anulowanych lub nieudanych zadań	681
Krótkie podsumowanie rozdziału 27	682
28 Równoległy dostęp do danych	685
Używanie technologii PLINQ do zrównoleglania deklaratywnego dostępu do danych	686
Używanie technologii PLINQ do zwiększenia wydajności iteracji poprzez elementy kolekcji	687
Określanie opcji dla zapytania PLINQ	692
Anulowanie zapytania PLINQ	693
Synchronizowanie jednoczesnego dostępu do danych	693
Blokowanie danych	697
Podstawowe typy danych służące do synchronizacji oferowane przez bibliotekę TPL	698
Mechanizmy anulowania a podstawowe typy danych służące do synchronizacji	706
Klasy kolekcji przystosowane do środowiska wielowątkowego	706

Wykorzystanie wielowątkowych klas kolekcji oraz blokad do zaimplementowania bezpiecznego dostępu do danych w środowisku wielowątkowym	709
Krótkie podsumowanie rozdziału 28	720
29 Tworzenie i wykorzystywanie usług webowych	723
Czym jest usługa webowa?	724
Rola platformy Windows Communication Foundation	724
Architektura usługi webowej	725
Usługi webowe typu SOAP	725
Usługi webowe typu REST	729
Tworzenie usług webowych	730
Tworzenie usługi webowej typu SOAP <i>ProductInformation</i>	730
Usługi webowe typu SOAP, klienci i pośrednicy	739
Korzystanie z usługi webowej typu SOAP: <i>ProductInformation</i>	741
Tworzenie usługi webowej typu REST: <i>ProductDetails</i>	747
Korzystanie z usługi webowej typu REST: <i>ProductDetails</i>	755
Krótkie podsumowanie rozdziału 29	760
Dodatek Współdziałanie z dynamicznymi językami programowania	761
Czym jest dynamiczne środowisko uruchomieniowe?	762
Słowo kluczowe <i>dynamic</i>	764
Przykład: IronPython	764
Przykład: IronRuby	767
Podsumowanie	770
Indeks	771