

John Sharp

Microsoft Visual C# 2015

Krok po kroku

Wydanie ósme

Przekład: Natalia Chounlamany, Janusz Machowski, Krzysztof Szkudlarek,
Marek Włodarz

APN Promise, Warszawa 2016

Spis treści

<i>Wstęp</i>	xiii
Część I: Wprowadzenie do języka Microsoft Visual C# oraz programu Microsoft Visual Studio 2015	
1 Wprowadzenie do języka C#	3
Rozpoczynamy programowanie przy użyciu środowiska Visual Studio 2015	3
Piszemy pierwszy program	9
Przestrzenie nazw	16
Tworzenie aplikacji graficznej	20
Analiza aplikacji Sklepu Windows	31
Dodawanie kodu do aplikacji graficznej	36
Podsumowanie	38
Krótki przegląd rozdziału 1	39
2 Zmienne, operatory i wyrażenia	41
Instrukcje	41
Identyfikatory	42
Słowa kluczowe	43
Zmienne	44
Nazywanie zmiennych	44
Deklarowanie zmiennych	45
Podstawowe typy danych	46
Zmienne lokalne bez przypisanej wartości	46
Wyświetlanie wartości podstawowych typów danych	47
Posługiwanie się operatorami arytmetycznymi	54
Operatory i typy danych	55
Poznajemy operatory arytmetyczne	57
Kontrolowanie pierwszeństwa	63
Stosowanie zasad łączności przy wyznaczaniu wartości wyrażień	64
Zasady łączności a operator przypisania	65
Inkrementacja i dekrementacja wartości zmiennych	66
Formy przyrostkowe i przedrostkowe	66
Deklarowanie zmiennych lokalnych o niejawnie określonym typie danych	67
Podsumowanie	69
Krótki przegląd rozdziału 2	69

3	Tworzenie metod i stosowanie zasięgów zmiennych	71
	Tworzenie metod	71
	Deklarowanie metody	72
	Zwracanie danych przez metodę	73
	Stosowanie metod wcielających wyrażenie	74
	Wywoływanie metod	76
	Stosowanie zasięgu	79
	Definiowanie zasięgu lokalnego	80
	Definiowanie zasięgu klasy	80
	Przeciążanie metod	81
	Tworzenie metod	82
	Stosowanie parametrów opcjonalnych oraz nazwanych argumentów	93
	Definiowanie parametrów opcjonalnych	95
	Przekazywanie nazwanych argumentów	95
	Rozwiązywanie niejednoznaczności związanych z parametrami opcjonalnymi i argumentami nazwanymi	96
	Podsumowanie	102
	Krótki przegląd rozdziału 3	103
4	Instrukcje wyboru	105
	Deklarowanie zmiennych logicznych	105
	Stosowanie operatorów logicznych	106
	Operatory równościowe oraz operatory relacji	106
	Warunkowe operatory logiczne	107
	Skracanie działania	108
	Podsumowanie informacji o pierwszeństwie oraz łączności operatorów	109
	Podejmowanie decyzji przy użyciu instrukcji if	110
	Składnia instrukcji if	110
	Grupowanie instrukcji w bloki	112
	Kaskadowe łączenie instrukcji if	113
	Stosowanie instrukcji switch	119
	Składnia instrukcji switch	120
	Reguły stosowania instrukcji switch	121
	Podsumowanie	126
	Krótki przegląd rozdziału 4	126
5	Złożone instrukcje przypisania oraz instrukcje iteracji	129
	Złożone operatory przypisania	129
	Instrukcja while	131
	Instrukcja for	137
	Zasięg instrukcji for	139
	Instrukcja do	139
	Podsumowanie	149

Krótki przegląd rozdziału 5	150
6 Obsługa błędów i wyjątków	151
Zmaganie się z błędami	152
Wypróbowywanie kodu i przechwytywanie wyjątków	152
Nieobsłużone wyjątki	154
Stosowanie kilku bloków obsługi pułapki	155
Przechwytywanie wielu wyjątków	156
Propagowanie wyjątków	163
Wykonywanie operacji arytmetycznych z kontrolą lub bez kontroli przepełnienia	165
Pisanie instrukcji objętych kontrolą przepełnienia	166
Pisanie wyrażeń objętych kontrolą przepełnienia	167
Zgłaszanie wyjątków	171
Stosowanie bloku <i>finally</i>	177
Podsumowanie	179
Krótki przegląd rozdziału 6	179

Część II: Omówienie modelu obiektowego języka C#

7 Tworzenie i zarządzanie klasami oraz obiektami	183
Omówienie klasyfikacji	183
Cele hermetyzacji	184
Definiowanie i używanie klas	184
Kontrolowanie dostępności	186
Konstruktory	188
Przeciążanie konstruktorów	190
Metody i dane statyczne	199
Tworzenie pól współdzielonych	201
Tworzenie pól statycznych przy użyciu słowa kluczowego <i>const</i>	201
Klasy statyczne	202
Statyczne instrukcje <i>using</i>	203
Klasy anonimowe	206
Podsumowanie	207
Krótki przegląd rozdziału 7	208
8 Wartości i referencje	211
Kopiowanie klas oraz zmiennych typu wartościowego	211
Wartości null oraz typy danych dopuszczające stosowanie wartości null	218
Typy danych dopuszczające stosowanie wartości <i>null</i>	221
Właściwości typów danych dopuszczających stosowanie wartości <i>null</i>	222
Używanie parametrów typu <i>ref</i> i <i>out</i>	223
Tworzenie parametrów typu <i>ref</i>	224

Tworzenie parametrów typu <i>out</i>	225
Sposób organizacji pamięci komputera	227
Korzystanie ze stosu oraz ze serty	229
Klasa System.Object	230
Opakowywanie typów danych wewnątrz obiektów	231
Rozpakowywanie typów danych, opakowanych wewnątrz obiektów	232
Bezpieczne rzutowanie danych	234
Operator <i>is</i>	235
Operator <i>as</i>	235
Podsumowanie	238
Krótki przegląd rozdziału 8	239
9 Tworzenie typów wartości przy użyciu wyliczeń oraz struktur	241
Wyliczeniowe typy danych	241
Deklarowanie wyliczeniowego typu danych	242
Stosowanie wyliczeniowych typów danych	242
Wybór wartości literałów wyliczeniowych	243
Wybór typu danych używanego do wewnętrznego reprezentowania wartości wyliczeniowych	244
Struktury	247
Deklarowanie struktury	249
Omówienie różnic pomiędzy strukturami i klasami	250
Deklarowanie zmiennych strukturalnych	252
Omówienie inicjalizacji struktur	253
Kopiowanie zmiennych strukturalnych	258
Podsumowanie	263
Krótki przegląd rozdziału 9	263
10 Tablice	265
Deklarowanie i tworzenie tablicy	265
Deklarowanie zmiennych tablicowych	266
Tworzenie instancji tablicy	266
Wypełnianie tablic danymi i ich używanie	268
Tworzenie tablic o niejawnie określonym typie elementów	269
Korzystanie z indywidualnych elementów tablicy	270
Wykonywanie iteracji poprzez elementy tablicy	271
Przekazywanie tablic jako parametrów i zwracanie ich jako wartości metod	272
Kopiowanie tablic	274
Tablice wielowymiarowe	276
Tworzenie tablic nieregularnych	277
Podsumowanie	289
Krótki przegląd rozdziału 10	289

11	Tablice parametrów	291
	Przeciążanie: krótkie przypomnienie faktów	291
	Używanie argumentów będących tablicami	292
	Deklarowanie tablicy parametrów typu <i>params</i>	294
	Używanie parametru typu <i>params object[]</i>	296
	Stosowanie tablicy parametrów typu <i>params</i>	298
	Porównanie tablic parametrów z parametrami opcjonalnymi	301
	Podsumowanie	304
	Krótki przegląd rozdziału 11	304
12	Dziedziczenie	305
	Czym jest dziedziczenie?	305
	Korzystanie z mechanizmów dziedziczenia	306
	Powtórka informacji na temat klasy <i>System.Object</i>	308
	Wywoływanie konstruktora klasy bazowej	309
	Przypisywanie klas	310
	Deklarowanie metod z użyciem słowa kluczowego <i>new</i>	312
	Deklarowanie metod wirtualnych	314
	Deklarowanie metod z użyciem słowa kluczowego <i>override</i>	315
	Omówienie dostępu chronionego	318
	Metody rozszerzające	325
	Podsumowanie	330
	Krótki przegląd rozdziału 12	330
13	Tworzenie interfejsów oraz definiowanie klas abstrakcyjnych	333
	Interfejsy	333
	Definiowanie interfejsu	335
	Implementowanie interfejsu	335
	Odwoływanie się do klasy za pomocą jej interfejsu	337
	Praca z wieloma interfejsami	338
	Jawne implementowanie interfejsu	339
	Ograniczenia interfejsów	341
	Definiowanie i używanie interfejsów	342
	Klasy abstrakcyjne	352
	Metody abstrakcyjne	354
	Klasy zamknięte	354
	Metody zamknięte	355
	Implementowanie i używanie klas abstrakcyjnych	355
	Podsumowanie	362
	Krótki przegląd rozdziału 13	363
14	Proces oczyszczania pamięci i zarządzanie zasobami	365
	Żywot obiektów	365

Tworzenie destruktorów	367
Dlaczego istnieje proces oczyszczania pamięci?	369
Działanie procesu oczyszczania pamięci?	371
Zalecenia	372
Zarządzanie zasobami	372
Metody sprzątające	373
Sprzątanie w sposób odporny na występowanie wyjątków	373
Instrukcja <i>using</i> oraz interfejs <i>IDisposable</i>	374
Wywoływanie metody <i>Dispose</i> z poziomu destruktora	376
Implementacja metody sprzątającej w sposób odporny na występowanie wyjątków	379
Podsumowanie	389
Krótki przegląd rozdziału 14	389

Część III: Tworzenie rozszerzalnych typów danych w języku C#

15 Implementacja właściwości zapewniających dostęp do pól	393
Implementacja kapsułkowania przy użyciu metod	393
Co to są właściwości?	395
Używanie właściwości	398
Właściwości tylko do odczytu	399
Właściwości tylko do zapisu	399
Dostępność właściwości	400
Ograniczenia właściwości	401
Deklarowanie właściwości interfejsu	402
Zastępowanie metod właściwościami	404
Generowanie automatycznych właściwości	408
Inicjalizowanie obiektów przy użyciu właściwości	411
Podsumowanie	415
Krótki przegląd rozdziału 15	416
16 Indeksatory	419
Co to jest indeksator?	419
Przykład bez użycia indeksatorów	419
Ten sam przykład z wykorzystaniem indeksatorów	422
Aksesory indeksatora	424
Porównanie indeksatorów i tablic	425
Indeksatory w interfejsach	427
Stosowanie indeksatorów w aplikacjach Windows	428
Podsumowanie	434
Krótki przegląd rozdziału 16	435

17	Typy ogólne	437
	Problem z typem object	437
	Rozwiązanie z użyciem typów ogólnych.....	441
	Typy ogólne a klasy uogólnione.....	443
	Typy ogólne i nakładanie ograniczeń.....	444
	Tworzenie klasy ogólnej.....	444
	Teoria drzew binarnych.....	444
	Budowanie klasy drzewa binarnego przy użyciu typu ogólnego.....	448
	Tworzenie metody ogólnej.....	458
	Definiowanie metody ogólnej do budowy drzewa binarnego.....	459
	Interfejsy ogólne i niezgodność typów.....	461
	Interfejsy kowariantne.....	463
	Interfejsy kontrawariantne.....	465
	Podsumowanie.....	467
	Krótki przegląd rozdziału 17.....	468
18	Kolekcje	469
	Co to są klasy kolekcji?.....	469
	Klasa kolekcji List<T>	471
	Klasa kolekcji LinkedList<T>	474
	Klasa kolekcji Queue<T>	475
	Klasa kolekcji Stack<T>	476
	Klasa kolekcji Dictionary<TKey, TValue>	478
	Klasa kolekcji SortedList<TKey, TValue>	479
	Klasa kolekcji HashSet<T>	480
	Inicjalizowanie kolekcji.....	482
	Metody Find , predykaty i wyrażenia lambda.....	483
	Różne formy wyrażeń lambda.....	485
	Porównanie tablic i kolekcji.....	487
	Wykorzystanie klas kolekcji do gry w karty.....	487
	Podsumowanie.....	492
	Krótki przegląd rozdziału 18.....	493
19	Wyliczanie kolekcji	495
	Wyliczanie elementów kolekcji.....	495
	Ręczna implementacja modułu wyliczającego.....	496
	Implementowanie interfejsu IEnumerable	501
	Implementowanie modułu wyliczającego przy użyciu iteratora.....	504
	Prosty iterator.....	504
	Definiowanie modułu wyliczającego dla klasy Tree<TItem> przy użyciu iteratora.....	506
	Podsumowanie.....	509
	Krótki przegląd rozdziału 19.....	509

20	Wydzielanie logiki aplikacji i obsługa zdarzeń	511
	Co to są delegaty	512
	Przykłady delegatów w bibliotece klas .NET Framework	513
	Przykład zautomatyzowanej fabryki	514
	Implementowanie systemu sterowania fabryką bez użycia delegatów	516
	Implementowanie sterowania fabryką przy użyciu delegata	516
	Deklarowanie i używanie delegatów	519
	Delegaty i wyrażenia lambda	528
	Tworzenie adaptera metody	528
	Włączanie powiadomień za pomocą zdarzeń	529
	Deklarowanie zdarzenia	529
	Subskrypcja zdarzenia	530
	Anulowanie subskrypcji zdarzenia	531
	Zgłaszanie zdarzenia	531
	Zdarzenia interfejsu użytkownika	532
	Używanie zdarzeń	534
	Podsumowanie	540
	Krótki przegląd rozdziału 20	540
21	Odpytywanie danych w pamięci przy użyciu wyrażeń w języku zapytań	543
	Co to jest LINQ?	543
	Używanie LINQ w aplikacjach C#	544
	Wybieranie danych	546
	Filtrowanie danych	549
	Porządkowanie, grupowanie i agregowanie danych	549
	Łączenie danych	552
	Operatory zapytań	553
	Odpytywanie danych w obiektach <i>Tree<TItem></i>	555
	LINQ i opóźnione przetwarzanie	562
	Podsumowanie	566
	Krótki przegląd rozdziału 21	566
22	Przeciążanie operatorów	569
	Czym są operatory	569
	Ograniczenia operatorów	570
	Operatory przeciążone	570
	Tworzenie operatorów symetrycznych	572
	Przetwarzanie złożonej instrukcji przypisania	574
	Deklarowanie operatorów zwiększających i zmniejszających	575
	Operatory porównań w strukturach i klasach	576
	Definiowanie par operatorów	577
	Implementowanie operatorów	578

Operatory konwersji	585
Wbudowane metody konwersji	585
Implementowanie własnych operatorów konwersji	586
Tworzenie operatorów symetrycznych – uzupełnienie	587
Zapisywanie operatorów konwersji	588
Podsumowanie	590
Krótki przegląd rozdziału 22	591
Część IV: Tworzenie aplikacji Universal Windows Platform w języku C#	
23 Przyspieszanie działania za pomocą zadań	595
Po co stosować wielozadaniowość przy użyciu przetwarzania równoległego? ..	595
Narodziny procesora wielordzeniowego	596
Implementowanie wielozadaniowości w .NET Framework	597
Zadania, wątki i pula wątków	598
Tworzenie, uruchamianie i kontrolowanie zadań	600
Używanie klasy <i>Task</i> do implementacji równoległości	603
Tworzenie abstrakcji zadań za pomocą klasy <i>Parallel</i>	616
Kiedy nie używać klasy <i>Parallel</i>	621
Anulowanie zadań i obsługa wyjątków	623
Mechanizm anulowania kooperatywnego	624
Kontynuowanie w przypadku zadań anulowanych lub przerwanych z powodu wyjątku	639
Podsumowanie	639
Krótki przegląd rozdziału 23	640
24 Skracanie czasu reakcji za pomocą działań asynchronicznych	643
Implementowanie metod asynchronicznych	644
Definiowanie metod asynchronicznych: problem	645
Definiowanie metod asynchronicznych: rozwiązanie	648
Definiowanie metod asynchronicznych zwracających wartości	654
Wskazówki dotyczące metod asynchronicznych	656
Metody asynchroniczne i interfejsy API środowiska Windows Runtime	657
Zrównoleglanie deklaratywnego dostępu do danych za pomocą PLINQ	661
Wykorzystanie PLINQ do poprawy wydajności podczas wykonywania iteracji po elementach kolekcji	662
Anulowanie zapytania PLINQ	667
Synchronizowanie współbieżnych operacji dostępu do danych	668
Blokowanie danych	671
Elementarne narzędzia synchronizacji umożliwiające koordynowanie zadań ..	671
Anulowanie synchronizacji	674
Współbieżne klasy kolekcji	675

Wykorzystanie kolekcji współbieżnej i blokady do implementacji dostępu do danych przystosowanego do trybu wielowątkowego	676
Podsumowanie	687
Krótki przegląd rozdziału 24	687
25 Implementowanie interfejsu użytkownika aplikacji Universal Windows Platform	691
Funkcje aplikacji Universal Windows Platform	692
Budowa aplikacji UWP przy użyciu szablonu Blank App	696
Implementowanie skalowalnego interfejsu użytkownika	698
Stosowanie stylów do interfejsu użytkownika	732
Podsumowanie	743
Krótki przegląd rozdziału 25	743
26 Wyświetlanie i wyszukiwanie danych w aplikacjach Universal Windows Platform	745
Implementowanie wzorca projektowego Model-View-ViewModel	745
Wyświetlanie danych przy użyciu mechanizmu wiązania danych	747
Modyfikowanie danych przy użyciu wiązania danych	753
Stosowanie wiązania danych do kontrolki ComboBox	758
Tworzenie składnika ViewModel	760
Dodawanie poleceń do składnika ViewModel	764
Wyszukiwanie danych przy użyciu Cortany	775
Dostarczanie odpowiedzi głosowej na polecenia głosowe	788
Podsumowanie	792
Krótki przegląd rozdziału 26	793
27 Dostęp do zdalnej bazy danych z poziomu aplikacji Universal Windows Platform	795
Pobieranie informacji z bazy danych	796
Tworzenie modelu encji	803
Tworzenie i korzystanie z usługi web typu REST	813
Wstawianie, aktualizacja i usuwanie danych za pośrednictwem usługi web typu REST	830
Raportowanie błędów i aktualizacja interfejsu użytkownika	842
Podsumowanie	851
Krótki przegląd rozdziału 27	852
<i>O autorze</i>	854
<i>Indeks</i>	855