

John Sharp

Microsoft Visual C# 2017

Krok po kroku

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

Spis treści

<i>Podziękowania</i>	xv
<i>O autorze</i>	xvii
<i>Wstęp</i>	xix
Część V Wprowadzenie do języka Visual C# i Microsoft Visual Studio 2017	
1 Wprowadzenie do języka C#	3
Rozpoczynamy programowanie przy użyciu środowiska Visual Studio 2017	3
Piszemy pierwszy program	9
Przestrzenie nazw	16
Tworzenie aplikacji graficznej	20
Analiza aplikacji Universal Windows Platform	31
Dodawanie kodu do aplikacji graficznej	35
Inne typy aplikacji graficznych	38
Podsumowanie	39
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
Specyfikowanie wartości numerycznych	46
Podstawowe typy danych	47
Zmienne lokalne bez przypisanej wartości	47
Wyświetlanie wartości podstawowych typów danych	48
Posługiwanie się operatorami arytmetycznymi	55
Operatory i typy danych	55
Poznajemy operatory arytmetyczne	58
Kontrolowanie pierwszeństwa	64
Stosowanie zasad łączności przy wyznaczaniu wartości wyrażień	64
Zasady łączności a operator przypisania	65
Inkrementacja i dekrementacja wartości zmiennych	66
Formy przyrostkowe i przedrostkowe	67
Deklarowanie zmiennych lokalnych o niejawnie określonym typie danych	68

Podsumowanie	69
Krótki przegląd rozdziału 2	70
3 Tworzenie metod i stosowanie zakresów zmiennych	71
Tworzenie metod	71
Deklarowanie metody	72
Zwracanie danych przez metodę	73
Stosowanie metod wcielających wyrażenie	74
Wywoływanie metod	76
Zwracanie wielu wartości z metody	79
Stosowanie zakresu	82
Definiowanie zakresu lokalnego	83
Definiowanie zakresu klasy	83
Przeciążanie metod	84
Tworzenie metod	85
Zagnieżdżanie metod	95
Stosowanie parametrów opcjonalnych oraz nazwanych argumentów	98
Definiowanie parametrów opcjonalnych	100
Przekazywanie nazwanych argumentów	101
Rozwiązywanie niejednoznaczności związanych z parametrami opcjonalnymi i argumentami nazwanymi	102
Podsumowanie	108
Krótki przegląd rozdziału 3	108
4 Instrukcje wyboru	111
Deklarowanie zmiennych logicznych	111
Stosowanie operatorów logicznych	112
Operatory równościowe oraz operatory relacji	112
Warunkowe operatory logiczne	113
Skracanie działania	114
Podsumowanie informacji o pierwszeństwie oraz łączności operatorów	115
Podejmowanie decyzji przy użyciu instrukcji <i>if</i>	116
Składnia instrukcji <i>if</i>	116
Grupowanie instrukcji w bloki	118
Kaskadowe łączenie instrukcji <i>if</i>	119
Stosowanie instrukcji <i>switch</i>	125
Składnia instrukcji <i>switch</i>	125
Reguły stosowania instrukcji <i>switch</i>	127
Podsumowanie	131
Krótki przegląd rozdziału 4	131
5 Złożone instrukcje przypisania oraz instrukcje iteracji	133
Złożone operatory przypisania	133
Instrukcja <i>while</i>	135

Instrukcja <i>for</i>	140
Zakres instrukcji <i>for</i>	142
Instrukcja <i>do</i>	143
Podsumowanie	152
Krótki przegląd rozdziału 5	152
6 Obsługa błędów i wyjątków	153
Zmaganie się z błędami	154
Wypróbowywanie kodu i przechwytywanie wyjątków	154
Nieobsłużone wyjątki	156
Stosowanie kilku bloków obsługi	157
Przechwytywanie wielu wyjątków	158
Filtry wyjątków	160
Propagowanie wyjątków	165
Wykonywanie operacji arytmetycznych z kontrolą lub bez kontroli przepełnienia	168
Pisanie instrukcji z kontrolą przepełnienia	169
Pisanie wyrażeń z kontrolą przepełnienia	170
Zgłaszanie wyjątków	172
Używanie wyjątków <i>throw</i>	178
Stosowanie bloku <i>finally</i>	179
Podsumowanie	181
Krótki przegląd rozdziału 6	181

Część VI Model obiektowy języka C#

7 Tworzenie i zarządzanie klasami oraz obiektami	185
Omówienie klasyfikacji	185
Cele hermetyzacji	186
Definiowanie i używanie klas	186
Kontrolowanie dostępności	188
Konstruktory	190
Przeciążanie konstruktorów	192
Dekonstrukcja obiektu	201
Metody i dane statyczne	203
Tworzenie pól współdzielonych	204
Tworzenie pól statycznych przy użyciu słowa kluczowego <i>const</i>	205
Klasy statyczne	205
Statyczne instrukcje <i>using</i>	206
Klasy anonimowe	209
Podsumowanie	210
Krótki przegląd rozdziału 7	211

8	Wartości i referencje	213
	Kopiowanie klas oraz zmiennych typu wartościowego	213
	Wartości null oraz typy danych dopuszczające stosowanie wartości null	220
	Operatory warunkowe wartości <i>null</i>	222
	Typy danych dopuszczające stosowanie wartości <i>null</i>	223
	Właściwości typów nullable	224
	Używanie parametrów typu <i>ref</i> i <i>out</i>	225
	Tworzenie parametrów typu <i>ref</i>	226
	Tworzenie parametrów typu <i>out</i>	227
	Organizacja pamięci komputera	229
	Korzystanie ze stosu i ze sterty	231
	Klasa <i>System.Object</i>	232
	Opakowywanie typów danych	233
	Rozpakowywanie typów danych	234
	Bezpieczne rzutowanie danych	236
	Operator <i>is</i>	236
	Operator <i>as</i>	237
	Ponowne odwiedziny instrukcji <i>switch</i>	237
	Podsumowanie	241
	Krótki przegląd rozdziału 8	241
9	Tworzenie typów wartościowych przy użyciu wyliczeń oraz struktur	243
	Wyliczeniowe typy danych	243
	Deklarowanie wyliczeniowego typu danych	244
	Stosowanie wyliczeniowych typów danych	244
	Wybór wartości literałów wyliczeniowych	245
	Wybór typu danych używanego do wewnętrznego reprezentowania wartości wyliczeniowych	246
	Struktury	249
	Deklarowanie struktury	251
	Różnice pomiędzy strukturami i klasami	252
	Deklarowanie zmiennych strukturalnych	254
	Omówienie inicjowania struktur	254
	Kopiowanie zmiennych strukturalnych	259
	Podsumowanie	264
	Krótki przegląd rozdziału 9	264
10	Tablice	267
	Deklarowanie i tworzenie tablicy	267
	Deklarowanie zmiennych tablicowych	268
	Tworzenie instancji tablicy	268
	Wypełnianie tablic danymi i ich używanie	270
	Tworzenie tablic o niejawnie określonym typie elementów	271
	Korzystanie z indywidualnych elementów tablicy	272

Wykonywanie iteracji poprzez elementy tablicy.	272
Przekazywanie tablic jako parametrów i zwracanie ich jako wartości metod.	274
Kopiowanie tablic	276
Tablice wielowymiarowe.	278
Tworzenie tablic nieregularnych	279
Dostęp do tablic zawierających typy wartościowe.	290
Podsumowanie.	294
Krótki przegląd rozdziału 10	294
11 Tablice parametrów	297
Przeciążanie: krótkie przypomnienie faktów.	297
Używanie argumentów będących tablicami	298
Deklarowanie tablicy parametrów typu <i>params</i>	300
Używanie parametru typu <i>params object[]</i>	302
Stosowanie tablicy parametrów typu <i>params</i>	304
Porównanie tablic parametrów z parametrami opcjonalnymi.	307
Podsumowanie.	310
Krótki przegląd rozdziału 11	310
12 Dziedziczenie.	311
Czym jest dziedziczenie?	311
Korzystanie z mechanizmów dziedziczenia.	312
Powtórka informacji na temat klasy <i>System.Object</i>	314
Wywoływanie konstruktora klasy bazowej	315
Przypisywanie klas	316
Deklarowanie metod z użyciem słowa kluczowego <i>new</i>	318
Deklarowanie metod wirtualnych	319
Deklarowanie metod z użyciem słowa kluczowego <i>override</i>	321
Dostęp chroniony.	324
Metody rozszerzające	330
Podsumowanie.	335
Krótki przegląd rozdziału 12	335
13 Tworzenie interfejsów oraz definiowanie klas abstrakcyjnych	337
Interfejsy	337
Definiowanie interfejsu	339
Implementowanie interfejsu.	339
Odwoływanie się do klasy za pomocą jej interfejsu.	341
Praca z wieloma interfejsami	342
Jawne implementowanie interfejsu.	343
Ograniczenia interfejsów.	345
Definiowanie i używanie interfejsów.	346
Klasy abstrakcyjne	356
Metody abstrakcyjne	357

Klasy zamknięte	358
Metody zamknięte	358
Implementowanie i używanie klas abstrakcyjnych	359
Podsumowanie	366
Krótki przegląd rozdziału 13	367
14 Proces oczyszczania pamięci i zarządzanie zasobami	369
Żywot obiektów	369
Tworzenie destruktorów	371
Dlaczego trzeba używać kolektorów śmieci?	373
Działanie procesu oczyszczania pamięci?	375
Zalecenia	376
Zarządzanie zasobami	377
Metody sprzątające	377
Sprzątanie w sposób odporny na występowanie wyjątków	378
Instrukcja <i>using</i> oraz interfejs <i>IDisposable</i>	379
Wywoływanie metody <i>Dispose</i> z poziomu destruktora	380
Implementacja metody sprzątającej odpornej na występowanie wyjątków	383
Podsumowanie	393
Krótki przegląd rozdziału 14	393
 Część VII Rozszerzalne typy danych w języku C#	
15 Implementacja właściwości zapewniających dostęp do pól	397
Implementacja hermetyzacji przy użyciu metod	397
Co to są właściwości?	399
Używanie właściwości	402
Właściwości tylko do odczytu	403
Właściwości tylko do zapisu	403
Dostępność właściwości	404
Ograniczenia właściwości	405
Deklarowanie właściwości interfejsu	407
Zastępowanie metod właściwościami	408
Generowanie automatycznych właściwości	412
Inicjowanie obiektów przy użyciu właściwości	415
Podsumowanie	419
Krótki przegląd rozdziału 15	419
16 Indeksatory i obsługa danych binarnych	423
Co to jest indeksator?	423
Przechowywanie danych dwójkowych	424
Wyświetlanie wartości dwójkowych	425
Manipulowanie wartościami dwójkowymi	425
Ten sam przykład z wykorzystaniem indeksatorów	427

Akcesory indeksatora	429
Porównanie indeksatorów i tablic	430
Indeksatory w interfejsach	432
Stosowanie indeksatorów w aplikacjach Windows	433
Podsumowanie	440
Krótki przegląd rozdziału 16	440
17 Typy ogólne	443
Problem niewłaściwego użycia typu <i>object</i>	443
Rozwiązanie z użyciem typów ogólnych	447
Typy ogólne a klasy uogólnione	449
Typy ogólne i nakładanie ograniczeń	450
Tworzenie klasy ogólnej	450
Teoria drzew binarnych	450
Budowanie klasy drzewa binarnego przy użyciu typu ogólnego	454
Tworzenie metody ogólnej	463
Definiowanie metody ogólnej do budowy drzewa binarnego	464
Interfejsy ogólne i niezgodność typów	467
Interfejsy kowariantne	468
Interfejsy kontrawariantne	470
Podsumowanie	473
Krótki przegląd rozdziału 17	473
18 Kolekcje	475
Co to są klasy kolekcji?	475
Klasa kolekcji <i>List<T></i>	477
Klasa kolekcji <i>LinkedList<T></i>	480
Klasa kolekcji <i>Queue<T></i>	481
Klasa kolekcji <i>Stack<T></i>	482
Klasa kolekcji <i>Dictionary<TKey, TValue></i>	484
Klasa kolekcji <i>SortedList<TKey, TValue></i>	485
Klasa kolekcji <i>HashSet<T></i>	486
Inicjowanie kolekcji	488
Metody <i>Find</i> , predykaty i wyrażenia lambda	489
Różne formy wyrażen lambda	491
Porównanie tablic i kolekcji	493
Wykorzystanie klas kolekcji do gry w karty	494
Podsumowanie	498
Krótki przegląd rozdziału 18	499
19 Wyliczanie kolekcji	501
Wyliczanie elementów kolekcji	501
Ręczna implementacja modułu wyliczającego	503
Implementowanie interfejsu <i>IEnumerable</i>	508

Implementowanie modułu wyliczającego przy użyciu iteratora	510
Prosty iterator	510
Definiowanie modułu wyliczającego dla klasy <i>Tree<TItem></i> przy użyciu iteratora	512
Podsumowanie	514
Krótki przegląd rozdziału 19	515
20 Wydzielanie logiki aplikacji i obsługa zdarzeń	517
Co to są delegaty	518
Przykłady delegatów w bibliotece klas .NET Framework	519
Przykład zautomatyzowanej fabryki	520
Implementowanie systemu sterowania fabryką bez użycia delegatów	522
Implementowanie sterowania fabryką przy użyciu delegata	522
Deklarowanie i używanie delegatów	525
Delegaty i wyrażenia lambda	534
Tworzenie adaptera metody	534
Włączanie powiadomień za pomocą zdarzeń	535
Deklarowanie zdarzenia	535
Subskrypcja zdarzenia	536
Anulowanie subskrypcji zdarzenia	537
Zgłaszanie zdarzenia	537
Zdarzenia interfejsu użytkownika	538
Używanie zdarzeń	540
Podsumowanie	546
Krótki przegląd rozdziału 20	546
21 Odpytywanie danych w pamięci przy użyciu wyrażen w języku zapytań	549
Co to jest LINQ?	549
Używanie LINQ w aplikacjach C#	550
Wybieranie danych	552
Filtrowanie danych	555
Porządkowanie, grupowanie i agregowanie danych	556
Złączanie danych	558
Operatory zapytań	559
Odpytywanie danych w obiektach <i>Tree<TItem></i>	562
LINQ i opóźnione przetwarzanie	568
Podsumowanie	572
Krótki przegląd rozdziału 21	572
22 Przeciążanie operatorów	575
Czym są operatory	575
Ograniczenia operatorów	576
Operatory przeciążone	577

Tworzenie operatorów symetrycznych	578
Złożone instrukcje przypisania	580
Deklarowanie operatorów inkrementujących i dekrementujących	581
Operatory porównań w strukturach i klasach	582
Definiowanie par operatorów	583
Implementowanie operatorów	584
Operatory konwersji	591
Wbudowane metody konwersji	591
Implementowanie własnych operatorów konwersji	592
Tworzenie operatorów symetrycznych – uzupełnienie	593
Zapisywanie operatorów konwersji	594
Podsumowanie	596
Krótki przegląd rozdziału 22	597

Część VIII Tworzenie aplikacji Universal Windows Platform w języku C#

23 Przyspieszanie działania za pomocą zadań	601
Po co stosować wielozadaniowość przy użyciu przetwarzania równoległego? ..	601
Narodziny procesora wielordzeniowego	602
Implementowanie wielozadaniowości w .NET Framework	604
Zadania, wątki i pula wątków	604
Tworzenie, uruchamianie i kontrolowanie zadań	606
Używanie klasy <i>Task</i> do implementacji równoległości	609
Tworzenie abstrakcji zadań za pomocą klasy <i>Parallel</i>	623
Kiedy nie używać klasy <i>Parallel</i>	627
Anulowanie zadań i obsługa wyjątków	630
Mechanizm anulowania kooperatywnego	630
Kontynuowanie w przypadku zadań anulowanych lub przerwanych z powodu wyjątku	644
Podsumowanie	645
Krótki przegląd rozdziału 23	645
24 Skracanie czasu reakcji za pomocą działań asynchronicznych	649
Implementowanie metod asynchronicznych	650
Definiowanie metod asynchronicznych: problem	651
Definiowanie metod asynchronicznych: rozwiązanie	654
Definiowanie metod asynchronicznych zwracających wartości	661
Wskazówki dotyczące metod asynchronicznych	662
Metody asynchroniczne i interfejsy API środowiska Windows Runtime	664
Zadania, alokacje pamięci i wydajność	666
Zrównoleglanie deklaratywnego dostępu do danych za pomocą PLINQ	670
Wykorzystanie PLINQ do poprawy wydajności przy wykonywaniu iteracji po elementach kolekcji	670

Anulowanie zapytania PLINQ	675
Synchronizowanie współbieżnych operacji dostępu do danych	676
Blokowanie danych	679
Elementarne narzędzia synchronizacji umożliwiające koordynowanie zadań	680
Anulowanie synchronizacji	683
Współbieżne klasy kolekcji	683
Wykorzystanie kolekcji współbieżnej i blokady do implementacji dostępu do danych przystosowanego do trybu wielowątkowego	684
Podsumowanie	695
Krótki przegląd rozdziału 24	696
25 Implementowanie interfejsu użytkownika aplikacji Universal Windows Platform	699
Cechy aplikacji Universal Windows Platform	701
Budowa aplikacji UWP przy użyciu szablonu Blank App	704
Implementowanie skalowalnego interfejsu użytkownika	707
Stosowanie stylów do interfejsu użytkownika	738
Podsumowanie	748
Krótki przegląd rozdziału 25	748
26 Wyświetlanie i wyszukiwanie danych w aplikacjach Universal Windows Platform	749
Implementowanie wzorca projektowego Model-View-ViewModel	749
Wyświetlanie danych przy użyciu mechanizmu wiązania danych	750
Modyfikowanie danych przy użyciu wiązania danych	757
Stosowanie wiązania danych do kontrolki <i>ComboBox</i>	762
Tworzenie składnika ViewModel	764
Dodawanie poleceń do składnika ViewModel	768
Wyszukiwanie danych przy użyciu Cortany	779
Dostarczanie odpowiedzi głosowej na polecenia głosowe	792
Podsumowanie	796
Krótki przegląd rozdziału 26	796
27 Dostęp do zdalnej bazy danych z poziomu aplikacji Universal Windows Platform	799
Pobieranie informacji z bazy danych	799
Tworzenie modelu encji	807
Tworzenie i korzystanie z usługi web typu REST	817
Wstawianie, aktualizacja i usuwanie danych za pośrednictwem usługi web typu REST	833
Raportowanie błędów i aktualizacja interfejsu użytkownika	844
Podsumowanie	852
Krótki przegląd rozdziału 27	852
Indeks	855