

John Sharp

Microsoft Visual C# 2022

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 I Wprowadzenie do języka Visual C# i Microsoft Visual Studio 2022

1 Wprowadzenie do języka C#	3
Pisanie pierwszego programu w języku C#	3
Zaczynamy programować w środowisku Visual Studio 2022	10
Pisanie pierwszego programu przy użyciu Visual Studio 2022	15
Przestrzenie nazw	23
Przestrzenie nazw i asemblacje	25
Komentowanie kodu	26
Tworzenie aplikacji graficznej	27
Analiza aplikacji Universal Windows Platform	40
Dodawanie kodu do aplikacji graficznej	44
Podsumowanie	49
Krótki przegląd rozdziału 1	49
2 Zmienne, operatory i wyrażenia	51
Instrukcje	51
Identyfikatory	52
Identyfikowanie słów kluczowych	53
Zmienne	54
Nazywanie zmiennych	54
Deklarowanie zmiennych	55
Specyfikowanie wartości numerycznych	56
Podstawowe typy danych	57
Zmienne lokalne bez przypisanej wartości	57
Wyświetlanie wartości podstawowych typów danych	58
Posługiwanie się operatorami arytmetycznymi	66
Operatory i typy danych	66
Poznajemy operatory arytmetyczne	69
Kontrolowanie pierwszeństwa	75
Stosowanie zasad łączności przy wyznaczaniu wartości wyrażen	76
Zasady łączności a operator przypisania	77

Inkrementacja i dekrementacja wartości zmiennych	78
Formy przyrostkowe i przedrostkowe	79
Deklarowanie zmiennych lokalnych o niejawnie określonym typie danych	80
Podsumowanie	81
Krótki przegląd rozdziału 2	82
3 Tworzenie metod i stosowanie zakresów zmiennych	83
Tworzenie metod	83
Deklarowanie metody	84
Zwracanie danych przez metodę	85
Stosowanie metod wcielających wyrażenie	86
Wywoływanie metod	88
Składnia wywołania metody	88
Zwracanie wielu wartości z metody	91
Stosowanie zakresu	93
Definiowanie zakresu lokalnego	94
Definiowanie zakresu klasy	95
Przeciążanie metod	96
Tworzenie metod	97
Używanie debugera Visual Studio do krokowego wykonywania metod	102
Refaktoryzowanie kodu	110
Zagnieżdżanie metod	111
Stosowanie parametrów opcjonalnych i nazwanych argumentów	114
Definiowanie parametrów opcjonalnych	116
Przekazywanie nazwanych argumentów	117
Rozwiązywanie niejednoznaczności związanych z parametrami opcjonalnymi i argumentami nazwanymi	117
Podsumowanie	124
Krótki przegląd rozdziału 3	124
4 Instrukcje wyboru	127
Deklarowanie zmiennych logicznych	128
Stosowanie operatorów logicznych	128
Operatory równościowe oraz operatory relacji	129
Warunkowe operatory logiczne	129
Skracanie działania	130
Podsumowanie informacji o pierwszeństwie oraz łączności operatorów ..	131
Dopasowywanie wzorców	132
Podejmowanie decyzji przy użyciu instrukcji if	133
Składnia instrukcji if	133
Grupowanie instrukcji w bloki	134
Kaskadowe łączenie instrukcji if	136

Stosowanie instrukcji switch	142
Składnia instrukcji switch	143
Reguły stosowania instrukcji switch	144
Używanie wyrażeń switch z dopasowywaniem wzorców	149
Podsumowanie	152
Krótki przegląd rozdziału 4	152
5 Złożone instrukcje przypisania oraz instrukcje iteracji	155
Złożone operatory przypisania	155
Instrukcja while	157
Instrukcja for	163
Instrukcja do	166
Podsumowanie	176
Krótki przegląd rozdziału 5	176
6 Obsługa błędów i wyjątków	177
Próbowanie kodu i przechwytywanie wyjątków	178
Nieobsłużone wyjątki	180
Stosowanie kilku bloków obsługi	181
Przechwytywanie wielu wyjątków	182
Filtrowanie wyjątków	184
Propagowanie wyjątków	189
Wykonywanie operacji arytmetycznych z kontrolą lub bez kontroli	
przepełnienia	191
Pisanie instrukcji z kontrolą przepełnienia	192
Pisanie wyrażeń z kontrolą przepełnienia	192
Zgłaszanie wyjątków	196
Używanie wyrażeń throw	202
Stosowanie bloku finally	203
Podsumowanie	204
Krótki przegląd rozdziału 6	205

Część II Model obiektowy języka C#

7 Tworzenie i zarządzanie klasami oraz obiektami	209
Istota klasyfikacji	209
Cele hermetyzacji	210
Definiowanie i używanie klas	210
Kontrolowanie dostępności	213
Konstruktory	215
Przeciążanie konstruktorów	216
Dekonstrukcja obiektu	226

Metody i dane statyczne	227
Tworzenie pól współdzielonych	228
Tworzenie pól statycznych przy użyciu słowa kluczowego const	229
Klasy statyczne	229
Statyczne instrukcje using	230
Klasy anonimowe	233
Podsumowanie	235
Krótki przegląd rozdziału 7	235
8 Wartości i referencje	237
Kopiowanie klas oraz zmiennych typu wartościowego	237
Wartości null oraz typy danych dopuszczające stosowanie wartości null	245
Operatory warunkowe wartości null	246
Nullowalne typy danych	248
Właściwości typów nullowalnych	249
Używanie parametrów typu ref i out	250
Tworzenie parametrów typu ref	251
Tworzenie parametrów typu out	252
Organizacja pamięci komputera	254
Korzystanie ze stosu i ze sterty	256
Klasa System.Object	257
Opakowywanie typów danych	258
Rozpakowywanie typów danych	259
Bezpieczne rzutowanie danych	261
Operator is	261
Operator as	262
Ponowne odwiedziny instrukcji switch	262
Podsumowanie	266
Krótki przegląd rozdziału 8	267
9 Tworzenie typów wartościowych przy użyciu wyliczeń oraz struktur	269
Wyliczeniowe typy danych	269
Deklarowanie wyliczeniowego typu danych	270
Stosowanie wyliczeniowych typów danych	270
Wybór wartości literałów wyliczeniowych	271
Wybór typu danych używanego do wewnętrznego reprezentowania wartości wyliczeniowych	272
Struktury	275
Deklarowanie struktury	277
Różnice pomiędzy strukturami i klasami	278
Deklarowanie zmiennych strukturalnych	280

Inicjowanie struktur	280
Kopiowanie zmiennych strukturalnych	285
Podsumowanie	289
Krótki przegląd rozdziału 9	289
10 Tablice	291
Deklarowanie i tworzenie tablicy	292
Tworzenie instancji tablicy	292
Wypełnianie tablic danymi i ich używanie	294
Tworzenie tablic o niejawnie określonym typie elementów	295
Uzyskiwanie dostępu do elementów tablicy	296
Uzyskiwanie dostępu do serii elementów tablicy	297
Wykonywanie iteracji poprzez elementy tablicy	297
Przekazywanie tablic jako parametrów i zwracanie ich jako wartości metod	299
Kopiowanie tablic	301
Tablice wielowymiarowe	303
Tworzenie tablic nieregularnych	304
Dostęp do tablic zawierających typy wartościowe	315
Podsumowanie	319
Krótki przegląd rozdziału 10	319
11 Tablice parametrów	321
Przeciążanie: krótkie przypomnienie faktów	321
Używanie argumentów tablicowych	322
Deklarowanie tablicy params	324
Używanie parametru typu params object[]	326
Stosowanie tablicy parametrów typu params	328
Porównanie tablic parametrów z parametrami opcjonalnymi	331
Podsumowanie	334
Krótki przegląd rozdziału 11	334
12 Dziedziczenie	335
Czym jest dziedziczenie?	335
Używanie dziedziczenia	336
Powtórka informacji na temat klasy System.Object	338
Wywoływanie konstruktora klasy bazowej	339
Przypisywanie klas	340
Deklarowanie nowych metod	342
Deklarowanie metod wirtualnych	343
Deklarowanie metod przesłaniających	345
Dostęp chroniony	348

Tworzenie metod rozszerzających	354
Podsumowanie	359
Krótki przegląd rozdziału 12	359
13 Tworzenie interfejsów oraz definiowanie klas abstrakcyjnych	361
Interfejsy	361
Definiowanie interfejsu	363
Implementowanie interfejsu	363
Odwoływanie się do klasy za pomocą jej interfejsu	365
Praca z wieloma interfejsami	366
Jawne implementowanie interfejsu	367
Obsługa wersjonowania a interfejsy	369
Ograniczenia interfejsów	371
Definiowanie i używanie interfejsów	371
Klasy abstrakcyjne	381
Metody abstrakcyjne	383
Klasy zamknięte	384
Metody zamknięte	384
Implementowanie i używanie klas abstrakcyjnych	385
Podsumowanie	392
Krótki przegląd rozdziału 13	393
14 Oczyszczanie pamięci i zarządzanie zasobami	395
Żywot obiektów	395
Tworzenie finalizatorów	397
Dlaczego trzeba używać kolektorów śmieci?	400
Działanie procesu oczyszczania pamięci	402
Zalecenia	403
Zarządzanie zasobami	404
Metody sprzątające	404
Sprzątanie w sposób odporny na występowanie wyjątków	405
Instrukcja using oraz interfejs IDisposable	406
Wywoływanie metody Dispose z poziomu finalizatora	407
Implementacja metody sprzątającej odpornej na występowanie wyjątków ..	410
Obsługiwanie zwalniania asynchronicznego	420
Podsumowanie	421
Krótki przegląd rozdziału 14	422

Część III Rozszerzalne typy danych w języku C#

15	Implementacja właściwości zapewniających dostęp do pól	427
	Implementacja hermetyzacji przy użyciu metod	428
	Czym są właściwości?	430
	Używanie właściwości	433
	Właściwości tylko do odczytu	433
	Właściwości tylko do zapisu	434
	Dostępność właściwości	434
	Ograniczenia właściwości	435
	Deklarowanie właściwości interfejsu	437
	Zastępowanie metod właściwościami	439
	Dopasowywanie wzorców we właściwościach	443
	Generowanie automatycznych właściwości	444
	Inicjowanie obiektów przy użyciu właściwości	446
	Automatyczne właściwości i niezmiennosc	448
	Używanie rekordów z właściwościami do implementowania lekkich struktur	451
	Podsumowanie	457
	Krótki przegląd rozdziału 15	457
16	Indeksatory i obsługa danych binarnych	461
	Czym jest indeksator?	461
	Przechowywanie danych binarnych	462
	Wyświetlanie wartości dwójkowych	463
	Manipulowanie wartościami dwójkowymi	463
	Ten sam przykład z wykorzystaniem indeksatorów	465
	Akcesory indeksatora	467
	Porównanie indeksatorów i tablic	468
	Indeksatory w interfejsach	470
	Stosowanie indeksatorów w aplikacjach Windows	471
	Podsumowanie	478
	Krótki przegląd rozdziału 16	478
17	Typy generyczne	481
	Problem niewłaściwego użycia typu object	481
	Rozwiązanie z użyciem typów generycznych	485
	Typy generyczne a klasy uogólnione	488
	Typy generyczne i nakładanie ograniczeń	488
	Tworzenie klasy generycznej	488
	Teoria drzew binarnych	489
	Budowanie klasy drzewa binarnego przy użyciu typu generycznego	492
	Tworzenie metody generycznej	502

Definiowanie metody generycznej do budowy drzewa binarnego	503
Wariancja i interfejsy generyczne	508
Interfejsy kowariantne	510
Interfejsy kontrawariantne	511
Podsumowanie	514
Krótki przegląd rozdziału 17	514
18 Kolekcje	517
Czym są klasy kolekcji?	517
Klasa kolekcji List<T>	519
Klasa kolekcji LinkedList<T>	522
Klasa kolekcji Queue<T>	524
Klasa kolekcji PriorityQueue<TElement, TPriority>	525
Klasa kolekcji Stack<T>	526
Klasa kolekcji Dictionary<TKey, TValue>	527
Klasa kolekcji SortedList<TKey, TValue>	528
Klasa kolekcji HashSet<T>	529
Używanie inicjalizatorów kolekcji	531
Metody Find, predykaty i wyrażenia lambda	532
Różne formy wyrażen lambda	534
Wyrażenia lambda i metody anonimowe	536
Porównanie tablic i kolekcji	536
Podsumowanie	541
Krótki przegląd rozdziału 18	542
19 Wyliczanie kolekcji	545
Wyliczanie elementów kolekcji	545
Ręczna implementacja modułu wyliczającego	547
Implementowanie interfejsu IEnumerable	552
Implementowanie modułu wyliczającego przy użyciu iteratora	554
Prosty iterator	555
Definiowanie modułu wyliczającego dla klasy Tree<TItem> przy użyciu iteratora	557
Podsumowanie	559
Krótki przegląd rozdziału 19	560
20 Wydzielanie logiki aplikacji i obsługa zdarzeń	561
Czym są delegaty	562
Przykłady delegatów w bibliotece klas .NET	563
Scenariusz zautomatyzowanej fabryki	564
Deklarowanie i używanie delegatów	569
Delegaty i wyrażenia lambda	578

Włączanie powiadomień za pomocą zdarzeń	580
Deklarowanie zdarzenia	580
Subskrypcja zdarzenia	581
Anulowanie subskrypcji zdarzenia	582
Zgłaszanie zdarzenia	582
Zdarzenia interfejsu użytkownika	583
Używanie zdarzeń	584
Podsumowanie	590
Krótki przegląd rozdziału 20	591
21 Odpytywanie danych w pamięci przy użyciu wyrażeń w języku zapytań	593
Czym jest LINQ?	594
Używanie LINQ w aplikacjach C#	594
Wybieranie danych	596
Filtrowanie danych	599
Porządkowanie, grupowanie i agregowanie danych	600
Złączanie danych	602
Operatory zapytań	603
Odpytywanie danych w obiektach Tree<TItem>	606
LINQ i opóźnione przetwarzanie	612
Podsumowanie	615
Krótki przegląd rozdziału 21	616
22 Przeciążanie operatorów	619
Czym są operatory	619
Ograniczenia operatorów	620
Operatory przeciążone	621
Tworzenie operatorów symetrycznych	622
Przetwarzanie złożonych instrukcji przypisania	624
Deklarowanie operatorów inkrementujących i dekrementujących	624
Operatory porównań w strukturach i klasach	625
Definiowanie par operatorów	626
Implementowanie operatorów	627
Przesłanie operatorów równościowych	631
Operatory konwersji	634
Wbudowane metody konwersji	634
Implementowanie własnych operatorów konwersji	635
Tworzenie operatorów symetrycznych – uzupełnienie	636
Tworzenie operatorów konwersji	637
Podsumowanie	639
Krótki przegląd rozdziału 22	640

Część IV Tworzenie aplikacji Universal Windows Platform w języku C#

23	Przyspieszanie działania za pomocą zadań	643
	Dlaczego realizować wielozadaniowość przy użyciu przetwarzania równoległego?	643
	Narodziny procesora wielordzeniowego	644
	Implementowanie wielozadaniowości w .NET	646
	Zadania, wątki i pula wątków	646
	Tworzenie, uruchamianie i kontrolowanie zadań	648
	Używanie klasy Task do implementacji równoległości	651
	Tworzenie abstrakcji zadań za pomocą klasy Parallel	661
	Kiedy nie używać klasy Parallel	666
	Anulowanie zadań i obsługa wyjątków	668
	Mechanizm anulowania kooperatywnego	669
	Obsługiwanie wyjątków zadań za pomocą klasy AggregateException	681
	Kontynuowanie w przypadku zadań anulowanych lub przerwanych z powodu wyjątku	683
	Podsumowanie	684
	Krótki przegląd rozdziału 23	684
24	Skracanie czasu reakcji za pomocą działań asynchronicznych	687
	Implementowanie metod asynchronicznych	688
	Definiowanie metod asynchronicznych: problem	689
	Definiowanie metod asynchronicznych: rozwiązanie	692
	Definiowanie metod asynchronicznych zwracających wartości	698
	Pułapki metod asynchronicznych	700
	Metody asynchroniczne i interfejsy API środowiska Windows Runtime	701
	Zadania, alokacje pamięci i wydajność	704
	Używanie PLINQ do zrównoleglenia deklaratywnego dostępu do danych	707
	Wykorzystanie PLINQ do poprawy wydajności przy wykonywaniu iteracji po elementach kolekcji	708
	Anulowanie zapytania PLINQ	713
	Synchronizowanie współbieżnych operacji dostępu do danych	714
	Blokowanie danych	717
	Elementarne narzędzia synchronizacji umożliwiające koordynowanie zadań	718
	Anulowanie synchronizacji	721
	Współbieżne klasy kolekcji	721
	Wykorzystanie kolekcji współbieżnej i blokady do implementacji dostępu do danych przystosowanego do trybu wielowątkowego	722

Podsumowanie	733
Krótki przegląd rozdziału 24.....	734
25 Implementowanie interfejsu użytkownika aplikacji	
Universal Windows Platform.....	737
Cechy aplikacji Universal Windows Platform	739
Budowa aplikacji UWP przy użyciu szablonu Blank App	742
Implementowanie skalowalnego interfejsu użytkownika.....	746
Implementowanie układu tabelarycznego za pomocą kontrolki Grid.....	757
Dostosowywanie układu za pomocą menedżera stanów wizualnych	766
Stosowanie stylów do interfejsu użytkownika	774
Podsumowanie	785
Krótki przegląd rozdziału 25.....	785
26 Wyświetlanie i wyszukiwanie danych w aplikacjach	
Universal Windows Platform.....	787
Implementowanie wzorca projektowego Model-View-ViewModel	787
Wyświetlanie danych przy użyciu wiązania	788
Modyfikowanie danych przy użyciu wiązania danych.....	795
Stosowanie wiązania danych do kontrolki ComboBox	800
Tworzenie składnika ViewModel	802
Dodawanie poleceń do składnika ViewModel.....	807
Podsumowanie	817
Krótki przegląd rozdziału 26.....	818
27 Dostęp do zdalnej bazy danych z poziomu aplikacji	
Universal Windows Platform.....	819
Pobieranie informacji z bazy danych	819
Tworzenie modelu encji	830
Tworzenie i korzystanie z usługi web typu REST.....	837
Aktualizowanie aplikacji UWP, aby używała usługi web.....	853
Wyszukiwanie danych w aplikacji Customers	863
Wstawianie, aktualizacja i usuwanie danych za pośrednictwem usługi web typu REST.....	870
Podsumowanie	872
Krótki przegląd rozdziału 27.....	873
<i>Indeks.....</i>	<i>875</i>