

Spis treści

O autorze	21
O recenzentach	22
Wstęp	23
Rozdział 1. Cześć, C#! Witaj, .NET!	29
Konfigurowanie środowiska programistycznego	31
Wybieranie narzędzia i typu aplikacji właściwych do nauki	31
Instalowanie na wielu platformach	34
Pobieranie i instalowanie Visual Studio 2022 dla Windows	35
Pobieranie i instalowanie Microsoft Visual Studio Code	36
Poznawanie .NET	39
Poznawanie .NET Framework	39
Poznawanie projektów Mono, Xamarin i Unity	40
Poznawanie .NET Core	40
Droga do jednej platformy .NET	41
Plany obsługi platformy .NET	42
Co się zmienia w nowoczesnym .NET?	44
Motywy kolorystyczne w nowoczesnym .NET	45
Poznawanie .NET Standard	45
Platformy .NET i narzędzia używane w tym wydaniu	46
Poznawanie języka IL	47
Porównanie technologii .NET	47
Tworzenie aplikacji konsoli za pomocą Visual Studio 2022	48
Zarządzanie wieloma projektami w Visual Studio 2022	48
Pisanie kodu za pomocą Visual Studio 2022	48
Kompilowanie i uruchamianie kodu w Visual Studio	50
Pisanie programów najwyższego poziomu	51
Dodawanie drugiego projektu w Visual Studio 2022	52

Tworzenie aplikacji konsoli za pomocą Visual Studio Code	54
Zarządzanie wieloma projektami w Visual Studio Code	55
Pisanie kodu za pomocą Visual Studio Code	55
Kompilowanie i uruchamianie kodu za pomocą polecenia dotnet	57
Dodawanie drugiego projektu w Visual Studio Code	58
Zarządzanie wieloma plikami za pomocą Visual Studio Code	60
Badanie kodu w interaktywnych notatnikach .NET	60
Tworzenie notatnika	60
Pisanie i uruchamianie kodu w notatniku	61
Zapisywanie notatnika	62
Dodawanie do notatnika składni Markdown i poleceń specjalnych	62
Uruchamianie kodu w wielu komórkach	64
Używanie interaktywnych notatników .NET do pisania kodu z tej książki	65
Przeglądanie folderów i plików projektów	65
Wspólne foldery i pliki	66
Kod w repozytorium GitHuba	66
Wykorzystywanie repozytorium GitHuba w tej książce	67
Pobieranie kodu rozwiązań z repozytorium GitHuba	67
Używanie systemu Git w Visual Studio Code	68
Gdzie znaleźć pomoc?	69
Przeglądanie dokumentacji Microsoftu	69
Uzyskiwanie pomocy dla narzędzia dotnet	69
Przeglądanie definicji typów i ich elementów	69
Poszukiwanie odpowiedzi na Stack Overflow	72
Poszukiwanie odpowiedzi za pomocą Google	72
Subskrybowanie blogów	73
Filmy Scotta Hanselmana	73
Praktyka i ćwiczenia	73
Ćwiczenie 1.1 — sprawdź swoją wiedzę	73
Ćwiczenie 1.2 — ćwicz C# gdzie się da	74
Ćwiczenie 1.3 — dalsza lektura	74
Podsumowanie	74
Rozdział 2. Mówimy w C#	76
Wprowadzenie do języka C#	76
Rozpoznawanie wersji oraz funkcji języka	77
Standardy języka C#	81
Odczytywanie wersji używanego kompilatora C#	81
Poznananie gramatyki i słownictwa języka C#	83
Wyświetlanie numeru wersji kompilatora	84
Gramatyka języka C#	85
Słownictwo języka C#	87
Porównanie języków programowania do języków ludzkich	88
Zmiana schematu kolorów składni języka C#	88
Pomoc przy pisaniu kodu	88
Importowanie przestrzeni nazw	89
Czasowniki jako metody	93
Rzeczowniki to typy, pola i zmienne	93
Ujawnianie wielkości słownika języka C#	94

Praca ze zmiennymi	96
Nazywanie zmiennych	96
Literały	97
Przechowywanie tekstu	97
Przechowywanie liczb	98
Przechowywanie liczb rzeczywistych	101
Przechowywanie wartości logicznych	104
Zapisywanie obiektów dowolnego typu	104
Przechowywanie typów dynamicznych	105
Deklarowanie zmiennych lokalnych	107
Odczytywanie i ustalanie domyślnych wartości typów	109
Zapisywanie wielu wartości w tablicy	110
Dokładniejsze poznawanie aplikacji konsoli	112
Wyświetlanie informacji dla użytkownika	112
Pobieranie danych od użytkownika	115
Uproszczone korzystanie z konsoli	116
Odczytywanie naciśnień klawiszy	117
Odczytywanie parametrów aplikacji konsoli	118
Ustalanie opcji za pomocą argumentów	120
Obsługiwanie platform nieobsługujących wybranych API	122
Praktyka i ćwiczenia	123
Ćwiczenie 2.1 — sprawdź swoją wiedzę	123
Ćwiczenie 2.2 — sprawdź swoją wiedzę o typach liczbowych	124
Ćwiczenie 2.3 — poznaj wielkości i zakresy liczb	124
Ćwiczenie 2.4 — dalsza lektura	125
Podsumowanie	125
Rozdział 3. Sterowanie przepływem i konwertowanie typów	126
Działania na zmiennych	126
Operatory jednoargumentowe	127
Dwuargumentowe operatory arytmetyczne	128
Operatory przypisania	129
Operatory logiczne	130
Warunkowe operatory logiczne	131
Operatory bitowe i operatory przesunięć	132
Operatory różne	134
Instrukcje wyboru	135
Instrukcja if	135
Dopasowywanie wzorców z instrukcją if	136
Instrukcja switch	137
Dopasowywanie wzorców z instrukcją switch	139
Upraszczenie instrukcji switch za pomocą wyrażeń switch	140
Instrukcje iteracji	141
Instrukcja while	141
Instrukcja do	142
Instrukcja for	143
Instrukcja foreach	143

Rzutowanie i konwertowanie między typami	144
Jawne i niejawne rzutowanie liczb	145
Używanie typu System.Convert	146
Zaokrąglanie liczb	147
Kontrolowanie sposobu zaokrąglania	148
Konwersja z dowolnego typu na ciąg znaków	149
Konwertowanie obiektu binarnego na ciąg znaków	149
Parsowanie ciągów znaków z liczbami, datami i czasem	150
Obsługa wyjątków	152
Instrukcja try	153
Wykrywanie przepełnień	157
Instrukcja checked	157
Instrukcja unchecked	158
Praktyka i ćwiczenia	159
Ćwiczenie 3.1 — sprawdź swoją wiedzę	159
Ćwiczenie 3.2 — pętle i przepełnienia	160
Ćwiczenie 3.3 — pętle i operatory	160
Ćwiczenie 3.4 — obsługa wyjątków	160
Ćwiczenie 3.5 — sprawdź swoją wiedzę o operatorach	161
Ćwiczenie 3.6 — dalsza lektura	161
Podsumowanie	162
Rozdział 4. Pisanie, debugowanie i testowanie funkcji	163
Tworzenie funkcji	163
Przykład z tabliczką mnożenia	164
Pisanie funkcji zwracającej wartość	166
Zamiana liczebników głównych na porządkowe	167
Rekurencyjne obliczanie silni	169
Dokumentowanie funkcji za pomocą komentarzy XML	172
Używanie wyrażenia lambda w implementacji funkcji	173
Debugowanie tworzonego programu	175
Tworzenie aplikacji z celowym błędem	176
Tworzenie punktu przerwania	177
Pasek narzędzi debugowania	180
Okna debugowania	181
Krokowe wykonywanie kodu	181
Dostosowywanie punktów przerwania	183
Protokołowanie błędów	185
Dostępne opcje protokołowania	186
Wykorzystywanie typów Debug i Trace	186
Konfigurowanie obiektów nasłuchujących	188
Przełączanie poziomów śledzenia	189
Testy jednostkowe	194
Różne rodzaje testów	195
Tworzenie biblioteki klas wymagającej testowania	195
Tworzenie testów jednostkowych	197
Rzucanie i wychwytywanie wyjątków w funkcjach	199
Rozróżnienie błędów użycia i błędów wykonania	199
Wyjątki często rzucające w funkcjach	200

Czym jest stos wywołań?	201
Gdzie należy wychwytywać wyjątki?	203
Ponowne rzucanie wyjątku	203
Implementowanie wzorca tester-wykonawca	205
Praktyka i ćwiczenia	206
Ćwiczenie 4.1 — sprawdź swoją wiedzę	206
Ćwiczenie 4.2 — tworzenie funkcji z wykorzystaniem debugowania i testów jednostkowych	207
Ćwiczenie 4.3 — dalsza lektura	207
Podsumowanie	208
Rozdział 5. Tworzenie własnych typów w programowaniu obiektowym	209
Programowanie obiektowe	210
Tworzenie bibliotek klas	211
Tworzenie biblioteki klas	211
Definiowanie klasy	212
Elementy klasy	213
Tworzenie obiektów	214
Importowanie przestrzeni nazw	215
Poznanie obiektów	215
Przechowywanie danych w polach	217
Definiowanie pól	217
Modyfikatory dostępu	217
Ustalanie i wypisywanie wartości pól	218
Zapisywanie wartości za pomocą słowa kluczowego enum	219
Przechowywanie wielu wartości w typie enum	220
Zapisywanie wielu wartości za pomocą kolekcji	222
Kolekcje generyczne	222
Tworzenie pól statycznych	223
Tworzenie stałych pól	224
Tworzenie pól tylko do odczytu	225
Inicjowanie pól w konstruktorach	226
Tworzenie i wywoływanie metod	227
Zwracanie wartości z metody	227
Łączenie wielu wartości za pomocą krotki	228
Sterowanie przekazywaniem parametrów	232
Przeciążanie metod	232
Parametry opcjonalne i nazywane	233
Sposoby przekazywania parametrów	235
Zwracanie wartości ze słowem kluczowym ref	236
Dzielenie klas na części	236
Kontrola dostępu za pomocą właściwości i indeksów	237
Definiowanie właściwości tylko do odczytu	237
Definiowanie właściwości z możliwością przypisania	239
Wymaganie podania wartości właściwości przy tworzeniu obiektu	240
Definiowanie indeksów	241
Dopasowywanie wzorców z obiektami	242
Tworzenie i używanie biblioteki klas .NET 6	242
Definiowanie listy pasażerów	242
Rozszerzenia dopasowywania wzorców w C# 9 i nowszych	244

Praca z rekordami	245
Właściwości wyłącznie inicjalizowane	245
Rekordy	246
Pozycyjne elementy danych w rekordach	247
Praktyka i ćwiczenia	248
Ćwiczenie 5.1 — sprawdź swoją wiedzę	248
Ćwiczenie 5.2 — dalsza lektura	248
Podsumowanie	248
Rozdział 6. Implementowanie interfejsów i dziedziczenie klas	249
Konfigurowanie biblioteki klas i aplikacji konsoli	250
Więcej informacji o metodach	251
Implementowanie działań w metodzie	251
Implementowanie działań za pomocą operatora	253
Definiowanie funkcji lokalnych	254
Wywoływanie i obsługa zdarzeń	255
Wywoływanie metod za pomocą delegatów	255
Definiowanie i obsługa delegatów	257
Definiowanie i obsługiwane zdarzeń	258
Wykorzystywanie typów generycznych	259
Praca z typami nienegenerycznymi	259
Praca z typami generycznymi	260
Implementowanie interfejsów	261
Typowe interfejsy	262
Porównywanie obiektów podczas sortowania	262
Porównywanie obiektów za pomocą osobnej klasy	265
Jawne i niejawne implementowanie interfejsów	266
Definiowanie interfejsów z domyślnymi implementacjami	267
Zarządzanie pamięcią za pomocą typów referencyjnych i typów wartości	268
Definiowanie typów referencyjnych i typów wartości	269
Sposób przechowywania w pamięci typów referencyjnych i typów wartości	270
Równość typów	271
Definiowanie typu kategorii struct	272
Praca z typami record struct	273
Zwalnianie niezarządzanych zasobów	273
Wymuszanie wywołania metody Dispose	275
Praca z wartościami null	276
Przekształcanie typu wartości w typ nullable	276
Poznanie nullable typów referencyjnych	277
Włączanie nullable i non-nullable typów referencyjnych	278
Deklarowanie non-nullable zmiennych i parametrów	278
Sprawdzanie wartości null	280
Dziedziczenie klas	281
Rozbudowywanie klasy	282
Ukrywanie elementów	282
Pokrywanie elementów klasy	283
Dziedziczenie po klasach abstrakcyjnych	284
Blokowanie dziedziczenia i pokrywania	286
Polimorfizm	286

Rzutowanie w ramach hierarchii dziedziczenia	288
Rzutowanie niejawne	288
Rzutowanie jawne	288
Obsługa wyjątków rzutowania	289
Dziedziczenie i rozbudowywanie typów .NET	290
Dziedziczenie po wyjątku	291
Rozszerzanie typów, po których nie można dziedziczyć	292
Stosowanie analizatorów, aby tworzyć lepszy kod	294
Ukrywanie komunikatów o ostrzeżeniach	296
Praktyka i ćwiczenia	300
Ćwiczenie 6.1 — sprawdź swoją wiedzę	300
Ćwiczenie 6.2 — tworzenie hierarchii dziedziczenia	300
Ćwiczenie 6.3 — dalsza lektura	301
Podsumowanie	301
Rozdział 7. Poznawanie typów .NET	302
Wprowadzenie do .NET 6	302
.NET Core 1.0	304
.NET Core 1.1	304
.NET Core 2.0	304
.NET Core 2.1	304
.NET Core 2.2	305
.NET Core 3.0	305
.NET Core 3.1	305
.NET 5.0	305
.NET 6.0	306
Poprawki wydajności między .NET Core 2.0 a .NET 5	306
Sprawdzenie dostępności aktualizacji .NET SDK	307
Zestawy i przestrzenie nazw	307
Zestawy, pakiety i przestrzenie nazw	308
Poznawanie pakietów SDK dla projektów .NET	309
Przestrzenie nazw i typy w zestawach	309
Pakiety NuGet	310
Czym są frameworki?	310
Importowanie przestrzeni nazw w celu użycia typu	311
Związki słów kluczowych języka C# z typami .NET	311
Odwzorowywanie aliasów języka C# na typy .NET	312
Wieloplatformowe współdzielenie kodu z bibliotekami klas .NET Standard	314
Domyślne ustawienia bibliotek klas w różnych wersjach SDK	315
Tworzenie biblioteki klas .NET Standard 2.0	316
Kontrolowanie wersji .NET SDK	317
Publikowanie własnych aplikacji	318
Tworzenie aplikacji konsoli do publikacji	319
Poznawanie polecenia dotnet	320
Pobieranie informacji na temat platformy .NET i jej środowiska	320
Zarządzanie projektami	322
Publikowanie samodzielnej aplikacji	322
Publikowanie aplikacji jednoplikowej	324
Zmniejszanie wielkości aplikacji	325

Dekompilowanie zestawów	326
Dekompilowanie za pomocą rozszerzenia ILSpy w Visual Studio 2022	327
Dekompilowanie za pomocą rozszerzenia ILSpy w Visual Studio Code	328
Nie, nie można zablokować możliwości dekompilowania	331
Przygotowywanie własnych pakietów NuGet	333
Dodawanie odwołania do pakietu	333
Tworzenie pakietu dla NuGet	334
Przeszukiwanie pakietów NuGet	338
Testowanie pakietu	339
Przenoszenie kodu z .NET Framework do .NET Core	340
Co można przenieść?	340
Co należy przenieść?	341
Różnice między .NET Framework i nowoczesnym .NET	341
Korzystanie z programu .NET Portability Analyzer	342
Asystent uaktualniania programu .NET	342
Używanie bibliotek spoza .NET	342
Praca z proponowanymi funkcjami	344
Wymaganie proponowanych funkcji	345
Włączanie proponowanych funkcji	345
Matematyka typów generycznych	346
Praktyka i ćwiczenia	346
Ćwiczenie 7.1 — sprawdź swoją wiedzę	346
Ćwiczenie 7.2 — dalsza lektura	347
Ćwiczenie 7.3 — PowerShell	347
Podsumowanie	347
Rozdział 8. Używanie typów .NET	348
Praca z liczbami	348
Praca z wielkimi liczbami całkowitymi	349
Praca z liczbami zespolonymi	350
Kwaterniony	351
Praca z tekstem	351
Odczytywanie długości ciągu znaków	351
Odczytywanie znaków z ciągu	352
Dzielenie ciągu znaków	352
Pobieranie części ciągu znaków	353
Poszukiwanie tekstu w ciągu	353
Inne elementy klasy string	354
Wydajne tworzenie ciągów znaków	355
Praca z datami i czasem	356
Definiowanie wartości daty i czasu	356
Globalizacja dla zapisu daty i czasu	358
Praca z samą datą lub samym czasem	360
Dopasowywanie wzorców za pomocą wyrażeń regularnych	360
Kontrolowanie cyfr wprowadzonych jako tekst	361
Poprawianie wydajności wyrażeń regularnych	362
Składnia wyrażenia regularnego	362
Przykłady wyrażeń regularnych	363
Dzielenie złożonych ciągów znaków rozdzielanych przecinkami	364

Praca z kolekcjami	365
Wspólne funkcje wszystkich kolekcji	366
Poprawianie wydajności przez zdefiniowanie pojemności kolekcji	367
Poznananie kolekcji	368
Praca z listami	372
Praca ze słownikami	373
Praca z kolejkami	374
Sortowanie kolekcji	377
Używanie specjalizowanych kolekcji	378
Używanie kolekcji niezmiennych	378
Dobre praktyki w pracy z kolekcjami	379
Praca z typem Span, indeksami i zakresami	380
Wydajne korzystanie z pamięci za pomocą typu Span	380
Określanie pozycji za pomocą typu Index	380
Definiowanie zakresów za pomocą typu Range	381
Używanie indeksów i zakresów	381
Praca z zasobami sieciowymi	382
Praca z adresami URI, serwerami DNS i adresami IP	383
Pingowanie serwera	384
Praca z refleksją i atrybutami	385
Numery wersji zestawów	386
Odczytywanie metadanych zestawu	386
Tworzenie własnych atrybutów	388
Inne możliwości refleksji	391
Praca z obrazami	391
Internacjonalizacja kodu	393
Wykrywanie i zmienianie bieżącej kultury	393
Praktyka i ćwiczenia	396
Ćwiczenie 8.1 — sprawdź swoją wiedzę	396
Ćwiczenie 8.2 — wyrażenia regularne	396
Ćwiczenie 8.3 — metody rozszerzające	396
Ćwiczenie 8.4 — dalsza lektura	397
Podsumowanie	397
Rozdział 9. Praca z plikami, strumieniami i serializacją	398
Praca z systemem plików	398
Obsługa środowisk i systemów plików na wielu platformach	399
Obsługa napędów	401
Praca z katalogami	402
Praca z plikami	403
Praca ze ścieżkami	405
Odczytywanie informacji o pliku	405
Zarządzanie plikami	407
Odczytywanie i zapisywanie w strumieniach	407
Strumienie abstrakcyjne i konkretne	408
Praca ze strumieniami tekstowymi	410
Praca ze strumieniami XML	411
Zwalnianie zasobów plików	413
Strumienie kompresujące	415
Kompresowanie za pomocą algorytmu Brotli	417

Kodowanie tekstu	419
Kodowanie ciągu znaków jako tablicy bajtów	420
Kodowanie i dekodowanie tekstu w plikach	422
Serializacja obiektów	423
Serializacja do formatu XML	423
Generowanie kompaktowej struktury XML	426
Deserializacja danych z formatu XML	427
Serializowanie do formatu JSON	428
Wydajne przetwarzanie danych w formacie JSON	429
Kontrolowane przetwarzanie danych JSON	430
Nowe metody rozszerzające, które ułatwiają pracę z odpowiedziami HTTP	433
Przenoszenie kodu z biblioteki Newtonsoft do nowej biblioteki	433
Praktyka i ćwiczenia	433
Ćwiczenie 9.1 — sprawdź swoją wiedzę	433
Ćwiczenie 9.2 — serializowanie do formatu XML	434
Ćwiczenie 9.3 — dalsza lektura	435
Podsumowanie	435
Rozdział 10. Praca z bazami danych przy użyciu Entity Framework Core	436
Nowoczesne bazy danych	436
Czym jest Entity Framework?	437
Entity Framework Core	437
Tworzenie aplikacji konsoli do pracy z EF Core	438
Używanie przykładowej relacyjnej bazy danych	438
Używanie Microsoft SQL Server w systemie Windows	439
Tworzenie przykładowej bazy danych Northwind dla SQL Server	441
Zarządzanie przykładową bazą danych Northwind w eksploratorze serwera	442
Używanie SQLite	443
Tworzenie przykładowej bazy danych Northwind na serwerze SQLite	444
Zarządzanie przykładową bazą danych Northwind za pomocą SQLiteStudio	445
Konfigurowanie EF Core	446
Wybieranie dostawcy danych EF Core	446
Łączenie z bazą danych	447
Definiowanie klasy kontekstu bazy danych Northwind	448
Definiowanie modeli EF Core	450
Konwencje w EF Core	450
Atrybuty EF Core	451
Płynne API EF Core	452
Tworzenie modelu w EF Core	453
Dodawanie tabel do klasy kontekstu bazy danych Northwind	456
Konfigurowanie narzędzia dotnet-ef	457
Tworzenie modeli na podstawie istniejącej bazy danych	458
Konfigurowanie konwencji	462
Zapytania do modelu EF Core	462
Filtrowanie dołączanych encji	464
Filtrowanie i sortowanie produktów	466
Pobieranie generowanych instrukcji SQL	467
Protokołowanie w EF Core	468

Dopasowywanie wzorców za pomocą instrukcji Like	472
Definiowanie globalnych filtrów	474
Wzorce ładowania w EF Core	475
Chętne ładowanie encji	475
Włączenie leniwego ładowania	476
Jawne ładowanie encji	477
Manipulowanie danymi w EF Core	479
Wstawianie encji	479
Aktualizowanie encji	481
Usuwanie encji	482
Grupowanie kontekstów baz danych	483
Transakcje	483
Sterowanie transakcjami za pomocą poziomów izolacji	484
Jawne definiowanie transakcji	484
Modele Code First w EF Core	485
Migracje	491
Praktyka i ćwiczenia	491
Ćwiczenie 10.1 — sprawdź swoją wiedzę	492
Ćwiczenie 10.2 — eksportowanie danych z wykorzystaniem różnych formatów serializacji	492
Ćwiczenie 10.3 — dalsza lektura	492
Ćwiczenie 10.4 — poznawanie baz danych NoSQL	493
Podsumowanie	493
Rozdział 11. Odczytywanie danych i manipulowanie nimi za pomocą LINQ	494
Tworzenie wyrażeń LINQ	494
Z czego składa się LINQ?	495
Rozbudowa sekwencji za pomocą klas wyliczeniowych	495
Filtrowanie encji za pomocą metody Where	499
Korzystanie z metody nazwanej	501
Upraszczenie kodu przez usunięcie jawnego tworzenia delegata	501
Korzystanie z wyrażenia lambda	502
Sortowanie encji	502
Deklarowanie zapytania za pomocą słowa kluczowego var lub określonego typu	503
Filtrowanie według typu	504
Praca ze zbiorami	505
Używanie LINQ z EF Core	507
Tworzenie modelu danych EF Core	508
Filtrowanie i sortowanie sekwencji	510
Projekcje sekwencji na nowe typy	512
Łączenie i grupowanie	513
Agregowanie sekwencji	517
Upiększanie składni	518
Używanie wielu wątków w równoległych zapytaniach LINQ	519
Tworzenie aplikacji korzystającej z wielu wątków	519
Tworzenie własnych metod rozszerzających dla LINQ	522
Próba użycia nowych metod rozszerzających	524
Próba użycia metod Mediana i Dominanta	525

Praca z LINQ to XML	526
Generowanie danych XML za pomocą LINQ to XML	526
Odczytywanie danych XML za pomocą LINQ to XML	527
Praktyka i ćwiczenia	528
Ćwiczenie 11.1 — sprawdź swoją wiedzę	528
Ćwiczenie 11.2 — zapytania LINQ	529
Ćwiczenie 11.3 — dalsza lektura	529
Podsumowanie	529
Rozdział 12. Poprawianie wydajności i skalowalności za pomocą wielozadaniowości	530
<hr/>	
Procesy, wątki i zadania	530
Monitorowanie wydajności i wykorzystania zasobów	531
Ocena wydajności typów	532
Monitorowanie wydajności i zużycia pamięci	533
Pomiar wydajności pracy z ciągami znaków	536
Monitorowanie wydajności i zużycia pamięci za pomocą biblioteki Benchmark.NET	537
Asynchroniczne uruchamianie zadań	540
Synchroniczne uruchamianie wielu operacji	541
Asynchroniczne uruchamianie wielu operacji z wykorzystaniem zadań	542
Oczekiwanie na zadania	543
Kontynuowanie pracy w innym zadaniu	545
Zadania zagnieżdżone i potomne	546
Tworzenie obiektów typu Task dla innych obiektów	547
Synchronizowanie dostępu do wspólnych zasobów	549
Używanie wspólnego zasobu w wielu wątkach	549
Nakładanie na zasoby wzajemnie wykluczającej blokady	551
Synchronizowanie zdarzeń	554
Tworzenie operacji atomowych	554
Stosowanie innych rodzajów synchronizacji	555
Słowa kluczowe async i await	556
Poprawianie reakcji aplikacji konsoli	556
Poprawianie reakcji aplikacji z interfejsem graficznym	557
Poprawianie skalowalności aplikacji i serwisów WWW	561
Często używane typy umożliwiające pracę wielowątkową	562
Instrukcja await w bloku catch	562
Praca ze strumieniami asynchronicznymi	562
Praktyka i ćwiczenia	563
Ćwiczenie 12.1 — sprawdź swoją wiedzę	564
Ćwiczenie 12.2 — dalsza lektura	564
Podsumowanie	564
Rozdział 13. Praktyczne aplikacje w języku C# i w .NET	565
<hr/>	
Modele aplikacji w C# i w .NET	566
Tworzenie stron WWW za pomocą ASP.NET Core	566
Tworzenie serwisów sieciowych	568

Tworzenie aplikacji mobilnych i stacjonarnych	569
Alternatywy dla .NET MAUI	570
Nowe funkcje w ASP.NET Core	571
ASP.NET Core 1.0	571
ASP.NET Core 1.1	571
ASP.NET Core 2.0	571
ASP.NET Core 2.1	571
ASP.NET Core 2.2	572
ASP.NET Core 3.0	572
ASP.NET Core 3.1	573
Blazor WebAssembly 3.2	573
ASP.NET Core 5.0	573
ASP.NET Core 6.0	574
Tworzenie aplikacji stacjonarnych dla systemów Windows	574
Starsze platformy aplikacji dla systemów Windows	574
Możliwości obsługi starszych platform Windows w nowoczesnym .NET	576
Struktury projektów	576
Struktura projektów w rozwiązaniu lub przestrzeni roboczej	576
Używanie innych szablonów projektów	578
Instalowanie dodatkowych pakietów szablonów	578
Tworzenie modelu danych dla bazy danych Northwind	579
Tworzenie biblioteki klas dla modelu encji Northwind	580
Tworzenie biblioteki klas modelu encji dla SQL Server	588
Praktyka i ćwiczenia	590
Ćwiczenie 13.1 — sprawdź swoją wiedzę	590
Ćwiczenie 13.2 — dalsza lektura	591
Podsumowanie	591
Rozdział 14. Tworzenie witryn WWW przy użyciu ASP.NET Core Razor Pages	592
Tworzenie w sieci WWW	592
Protokół HTTP	593
Używanie Google Chrome do wykonywania żądań HTTP	595
Tworzenie oprogramowania dla sieci WWW po stronie klienta	597
ASP.NET Core	598
Klasyczna ASP.NET kontra ASP.NET Core	599
Tworzenie pustego projektu ASP.NET Core	599
Testowanie i zabezpieczanie witryny	601
Kontrola środowiska hostingowego	606
Rozdzielanie konfiguracji serwisu i potoku	608
Włączanie plików statycznych	610
Technologia Razor Pages	612
Włączanie technologii Razor Pages	612
Definiowanie strony Razor	613
Używanie wspólnego układu w wielu stronach Razor	614
Używanie plików code-behind w stronach Razor	617
Używanie Entity Framework Core z ASP.NET Core	619
Konfigurowanie Entity Framework Core jako serwisu	619
Manipulowanie danymi na stronach Razor	622
Wstrzykiwanie zależnego serwisu na stronę Razor	623

Używanie bibliotek klas Razor	624
Tworzenie biblioteki klas Razor	624
Wyłączanie kompaktowych folderów w Visual Studio Code	625
Implementowanie funkcji pracowników za pomocą EF Core	626
Implementowanie widoku cząstkowego do wyświetlania danych pracownika	628
Używanie i testowanie biblioteki klas Razor	629
Konfigurowanie serwisów i potoku obsługi żądań HTTP	629
Routowanie punktów końcowych	630
Przeglądanie konfiguracji routowania punktów końcowych w naszym projekcie	631
Podsumowanie najważniejszych metod rozszerzających oprogramowania pośredniego	634
Wizualizacja potoku HTTP	635
Implementowanie oprogramowania pośredniego jako anonimowego delegata	636
Praktyka i ćwiczenia	638
Ćwiczenie 14.1 — sprawdź swoją wiedzę	638
Ćwiczenie 14.2 — tworzenie witryny obsługującej dane	638
Ćwiczenie 14.3 — zastępowanie aplikacji konsoli stronami WWW	638
Ćwiczenie 14.4 — dalsza lektura	639
Podsumowanie	639
Rozdział 15. Tworzenie aplikacji WWW przy użyciu ASP.NET Core MVC	640
Konfigurowanie witryny ASP.NET Core MVC	640
Tworzenie witryny ASP.NET Core MVC	641
Tworzenie bazy danych uwierzytelniania na serwerze SQL Server LocalDB	643
Przeglądanie domyślnej witryny ASP.NET Core MVC	643
Przegląd szablonu projektu ASP.NET Core MVC	645
Sprawdzanie bazy danych ASP.NET Core Identity	647
Poznanie mechanizmów ASP.NET Core MVC	648
Rozruch ASP.NET Core	648
Czym jest domyślna ścieżka?	651
Kontrolery i akcje	652
Konwencja wyszukiwania ścieżek widoku	655
Protokołowanie	655
Czym są filtry?	657
Modele encji i widoków	662
Widoki	665
Dostosowywanie witryny ASP.NET Core MVC	668
Definiowanie własnych stylów	668
Konfigurowanie obrazków dla kategorii	668
Poznanie składni stron Razor	669
Definiowanie typu dla widoku	670
Testowanie zmienionej strony startowej	672
Przekazywanie parametrów przy użyciu wartości ścieżki	673
Wiązanie modeli	675
Sprawdzanie poprawności modelu	679
Poznanie pomocniczych metod widoku	682
Odczytywanie danych z bazy i używanie szablonów wyświetlania	684
Poprawianie skalowalności za pomocą asynchronicznych zadań	687
Przygotowanie asynchronicznych metod akcji kontrolera	687

Praktyka i ćwiczenia	688
Ćwiczenie 15.1 — sprawdź swoją wiedzę	689
Ćwiczenie 15.2 — implementowanie wzorca MVC w tworzonej stronie danych kategorii	689
Ćwiczenie 15.3 — poprawianie skalowalności przez poznawanie i implementowanie asynchronicznych metod akcji	690
Ćwiczenie 15.4 — testy jednostkowe dla kontrolerów	690
Ćwiczenie 15.5 — dalsza lektura	690
Podsumowanie	690
Rozdział 16. Tworzenie i używanie serwisów sieciowych	691
Tworzenie serwisów w technologii ASP.NET Core Web API	691
Skróty stosowane w serwisach sieciowych	692
Żądania i odpowiedzi HTTP w Web API	693
Tworzenie projektu ASP.NET Core Web API	696
Sprawdzanie funkcji serwisu sieciowego	698
Tworzenie serwisu internetowego dla bazy danych Northwind	700
Tworzenie repozytorium danych dla encji	701
Implementowanie kontrolera Web API	705
Konfigurowanie repozytorium klientów i kontrolera Web API	706
Podawanie szczegółów problemu	711
Kontrola nad serializacją XML	711
Dokumentowanie i testowanie serwisów	712
Testowanie żądań GET za pomocą przeglądarki	712
Testowanie żądań HTTP za pomocą rozszerzenia REST Client	714
Włączanie narzędzia Swagger	717
Testowanie żądań w narzędziu SwaggerUI	719
Włączanie protokołowania HTTP	723
Używanie serwisu za pomocą klientów HTTP	725
Klasa HttpClient	725
Konfigurowanie klientów HTTP za pomocą klasy HttpClientFactory	725
Pobieranie w kontrolerze listy klientów w formacie JSON	726
Włączanie funkcji CORS	728
Implementowanie zaawansowanych funkcji	730
Implementowanie API Health Check	730
Implementowanie konwencji i analizatorów Open API	731
Implementowanie obsługi błędów przejściowych	732
Dodawanie zabezpieczających nagłówków HTTP	732
Tworzenie serwisów sieciowych używających minimalnego API	734
Tworzenie serwisu pogodowego z minimalnym API	735
Testowanie minimalnego serwisu pogodowego	736
Dodawanie prognozy pogody do głównej strony witryny Northwind	736
Praktyka i ćwiczenia	739
Ćwiczenie 16.1 — sprawdź swoją wiedzę	739
Ćwiczenie 16.2 — ćwiczenia w tworzeniu i usuwaniu klientów za pomocą HttpClient	740
Ćwiczenie 16.3 — dalsza lektura	740
Podsumowanie	740

Rozdział 17. Tworzenie interfejsów użytkownika w technologii Blazor	741
Technologia Blazor	742
JavaScript i podobne	742
Silverlight — C# i .NET w formie wtyczki	742
WebAssembly — podstawa technologii Blazor	742
Różne modele hostowania komponentów Blazora	743
Omówienie komponentów tworzonych za pomocą Blazora	744
Czym różnią się Blazor i Razor?	745
Porównanie szablonów projektów Blazor	745
Przeglądanie szablonu projektu Blazor Server	746
Routowanie do komponentów stronicowych Blazora	751
Uruchamianie szablonu projektu Blazor Server	754
Przeglądanie szablonu projektu Blazor WebAssembly	756
Tworzenie komponentów za pomocą Blazor Server	759
Definiowanie i testowanie prostego komponentu	760
Przekształcanie komponentu w routowalny komponent stronicowy	761
Dodawanie encji do komponentu	762
Tworzenie abstrakcji serwisu dla komponentu Blazora	765
Definiowanie formularzy za pomocą komponentu EditForm	767
Tworzenie i używanie komponentu formularza danych klienta	768
Testowanie komponentu formularza danych klienta	771
Tworzenie komponentów za pomocą Blazor WebAssembly	772
Konfigurowanie serwera dla projektu Blazor WebAssembly	772
Konfigurowanie klienta aplikacji Blazor WebAssembly	775
Testowanie komponentów i serwisu Blazor WebAssembly	778
Usprawnianie aplikacji tworzonych za pomocą Blazor WebAssembly	779
Włączanie funkcji Blazor WebAssembly AOT	780
Obsługa aplikacji PWA	781
Analizator zgodności przeglądarki dla aplikacji Blazor WebAssembly	783
Współdzielenie komponentów Blazora w bibliotece klas	784
Współpraca ze skryptami JavaScriptu	786
Biblioteki komponentów Blazora	789
Praktyka i ćwiczenia	789
Ćwiczenie 17.1 — sprawdź swoją wiedzę	789
Ćwiczenie 17.2 — przygotowanie komponentu tabliczki mnożenia	790
Ćwiczenie 17.3 — przygotowanie elementu nawigowania według krajów	790
Ćwiczenie 17.4 — dalsza lektura	790
Podsumowanie	790
Odpowiedzi na pytania z testów	791
Rozdział 1. Cześć, C#! Witaj, .NET!	791
Rozdział 2. Mówimy w C#	793
Ćwiczenie 2.1 — sprawdź swoją wiedzę	793
Ćwiczenie 2.2 — sprawdź swoją wiedzę o typach liczbowych	794
Rozdział 3. Sterowanie przepływem i konwertowanie typów	795
Ćwiczenie 3.1 — sprawdź swoją wiedzę	795
Ćwiczenie 3.2 — pętle i przepelnienia	796
Ćwiczenie 3.5 — sprawdź swoją wiedzę o operatorach	797

Rozdział 4. Pisanie, debugowanie i testowanie funkcji	797
Rozdział 5. Tworzenie własnych typów w programowaniu obiektowym	798
Rozdział 6. Implementowanie interfejsów i dziedziczenie klas	800
Rozdział 7. Poznawanie typów .NET	801
Rozdział 8. Używanie typów .NET	802
Rozdział 9. Praca z plikami, strumieniami i serializacją	804
Rozdział 10. Praca z bazami danych przy użyciu Entity Framework Core	805
Rozdział 11. Odczytywanie danych i manipulowanie nimi za pomocą LINQ	807
Rozdział 12. Poprawianie wydajności i skalowalności za pomocą wielozadaniowości	809
Rozdział 13. Praktyczne aplikacje w C# i .NET	810
Rozdział 14. Tworzenie witryn WWW przy użyciu ASP.NET Core Razor Pages	810
Rozdział 15. Tworzenie aplikacji WWW przy użyciu ASP.NET Core MVC	813
Rozdział 16. Tworzenie i używanie serwisów sieciowych	815
Rozdział 17. Tworzenie interfejsów użytkownika w technologii Blazor	817
Epilog	820
Następne kroki Twojej podróży w świecie C# i .NET	820
Poprawianie swoich umiejętności za pomocą wskazówek projektowych	820
Z których książek uczyć się dalej?	821
Opóźnienie .NET MAUI	821
Następne wydanie tej książki	822
Powodzenia!	822



Mark J. Price był pracownikiem Microsoftu specjalizującym się w programowaniu w języku C# i tworzeniu rozwiązań dla Microsoft Azure. Ma już ponad 30 lat doświadczenia w IT oraz

Microsoft
CERTIFIED
Microsoft Developer
App Builder

Microsoft
Specialist
Programming in C#

Od 2005 Mark miał ponad 50 egzaminów dla programistów organizowanych przez Microsoft i jego sukcesy się w programowaniu innych do tej chwili. W latach 2004 - 2005 Mark był zatrudniony w Redmond w USA, gdzie pisał oficjalne kursy dla Microsoftu jako członek programistowski platformy języka C#, i to jeszcze w momencie, gdy ten był w fazie wczesnej. Pracując przy dla Microsoftu Mark prowadził kursy „Język C# dla początkujących” w ramach szkoleń certyfikowanych trenerzy Microsoftu był zaangażowany z językiem C# i platformą .NET. Obecnie Mark prowadzi szkolenia ogólnokrajowe dla całego Digital Experience Platform (DXP). Używał narzędzi BSC (Hewlett) w doświadczenie zaimplementowanych