

Spis treści

Spis treści

O autorze	21
O recenzentach	22
Wstęp	23
ROZDZIAŁ 1	
Wprowadzenie do narzędzi i umiejętności dla .NET	28
Wprowadzenie do tej książki i jej zawartości	29
Dodatkowe książki jako podstawa do nauki	29
Odbiorcy tej książki	30
Narzędzia	31
Umiejętności	32
Testowanie	32
Projektowanie systemów i rozwój kariery	33
Konfigurowanie środowiska programistycznego	34
Wybór odpowiedniego narzędzia i typu aplikacji do nauki	34
Jakiego sprzętu i oprogramowania używałem?	36
Wdrażanie wieloplatformowe	37
Pobieranie i instalowanie Visual Studio	37
Pobieranie i instalowanie Visual Studio Code	40
Pobieranie i instalowanie Ridera	43
Inne narzędzia firmy JetBrains	44
Narzędzia AI w Chrome	45
Wykorzystywanie repozytorium GitHuba dla tej książki	45
Pobieranie kodu rozwiązań z repozytorium GitHuba	45
Gdzie szukać pomocy?	46
Dokumentacja Microsoft Learn	46
Uzyskiwanie pomocy do narzędzia dotnet	46
Modele językowe, takie jak ChatGPT	47

Uzyskiwanie pomocy na Discordzie i innych forach	50
Konfigurowanie bazy danych i projektów na potrzeby tej książki	52
Używanie przykładowej relacyjnej bazy danych	53
Konfigurowanie SQL Server i bazy danych Northwind	54
Tworzenie biblioteki klas dla modeli encji przy użyciu serwera SQL Server	56
Tworzenie biblioteki klas dla kontekstu danych używanego w SQL Server	59
Tworzenie projektu testowego w celu sprawdzenia integracji bibliotek klas	62
Uruchamianie testów	63
Używanie .NET 9 w pracach z tą książką	64
Ćwiczenie i dalsza nauka	66
Ćwiczenie 1.1. Materiały dostępne wyłącznie online	66
Ćwiczenie 1.2. Ćwiczenia praktyczne	66
Ćwiczenie 1.3. Sprawdź swoją wiedzę	67
Ćwiczenie 1.4. Eksploracja tematów	67
Podsumowanie	67
ROZDZIAŁ 2	
Pełne wykorzystanie edytora kodu	68
Wprowadzenie do powszechnie używanych narzędzi i funkcji	68
Funkcje refaktoryzacji	69
Wycinki kodu	70
Konfiguracja edytora	70
Asystenci AI	72
Narzędzia w Visual Studio 2022	72
Funkcje refaktoryzacji	73
Wycinki kodu	80
Konfiguracja edytora	86
Asystenci AI — GitHub Copilot	89

Nawigowanie w Visual Studio	91
Funkcje usprawniające edycję	91
Narzędzia w Visual Studio Code	95
Funkcje refaktoryzacji	96
Fragmenty kodu	96
Dekompilowanie zestawów .NET	98
Tworzenie aplikacji konsoli do zdekompilowania	98
Dekompilacja za pomocą rozszerzenia ILSpy dla Visual Studio	100
Przeglądanie kodu źródłowego za pomocą Visual Studio	103
Nie, nie można zapobiec dekompilacji	104
Obniżanie poziomu kodu C#	106
Własne szablony projektów i elementów	111
Tworzenie projektu dla szablonu	112
Testowanie szablonu projektu	117
Ćwiczenia i dalsza nauka	118
Ćwiczenie 2.1. Materiały dostępne wyłącznie online	118
Ćwiczenie 2.2. Ćwiczenia praktyczne	118
Ćwiczenie 2.3. Sprawdź swoją wiedzę	119
Podsumowanie	120
ROZDZIAŁ 3	
Zarządzanie kodem źródłowym za pomocą Gita	121
Wprowadzenie do zarządzania kodem źródłowym	122
Funkcje zarządzania kodem źródłowym	122
Rodzaje systemów SCM	123
Popularne systemy SCM	123
Wprowadzenie do Gita	124
Funkcje Gita	125
Dlaczego nauka Gita jest trudna?	126
Role w zespole korzystającym z Gita	126
Pobieranie najnowszej wersji Gita	126

Integracja Gita z Visual Studio	126
Konfiguracja tożsamości w Gicie	127
Wymuszanie podpisów SSH w Gicie	128
Konfiguracja domyślnej gałęzi w Gicie	130
Uzyskiwanie pomocy dla poleceń Gita	130
Praca z Gitem	130
Rozpoczynanie pracy z repozytorium Gita	131
Tworzenie i dodawanie plików do repozytorium Gita — teoria	132
Śledzenie zmian w Gicie	133
Tworzenie repozytorium Gita — praktyka	134
Tworzenie nowego projektu	134
Umieszczanie plików w repozytorium	138
Cofanie commitów w Gicie	139
Czyszczenie commitu	140
Przechowywanie	141
Ignorowanie plików	143
Przeglądanie repozytoriów Gita	145
Wyświetlanie różnic w plikach	146
Wyświetlanie historii commitów	148
Filtrowanie historii commitów	152
Zarządzanie zdalnymi repozytoriami	152
Rozgałęzianie i scalanie	155
Przykład rozgałęziania i scalania	156
Usuwanie gałęzi i wyświetlanie ich listy	161
Podsumowanie najczęściej używanych poleceń Gita	161
Ćwiczenia i dalsza nauka	162
Ćwiczenie 3.1. Materiały dostępne wyłącznie online	163
Ćwiczenie 3.2. Ćwiczenia praktyczne	163
Ćwiczenie 3.3. Sprawdź swoją wiedzę	164
Ćwiczenie 3.4. Dalsza nauka	164

Podsumowanie	164
ROZDZIAŁ 4	
Debugowanie i rozwiązywanie problemów z pamięcią	165
Strategie debugowania	165
Wprowadzenie do strategii debugowania	166
Zrozumieć problem	167
Jak rozpocząć debugowanie?	169
Kiedy przestać debugować?	170
Interaktywne debugowanie w Visual Studio	170
Tworzenie kodu do debugowania	170
Ustalanie punktu przerwania i rozpoczęcie debugowania	173
Pasek narzędzi debugowania	175
Okna debugowania	177
Kontrolowanie danych wyświetlanych w panelach debugowania	178
Debugowanie projektów testowych	180
Prośzenie usługi GitHub Copilot Chat o pomoc w debugowaniu	182
Pamięć stosu i sterty	184
Jak typy referencyjne i typy wartości są przechowywane w pamięci	184
Czym jest niebezpieczny kod?	187
Wskaźniki	187
Czym jest boxing?	191
Mechanizm oczyszczania pamięci	192
Zarządzanie zasobami za pomocą interfejsu IDisposable	193
Narzędzia do analizy pamięci	195
Popularne narzędzia i umiejętności analizy pamięci	195
Narzędzia Visual Studio	196
Korzystanie z narzędzia do analizy pamięci w Visual Studio	197
Ćwiczenia i dalsza nauka	200
Ćwiczenie 4.1. Materiały dostępne wyłącznie online	200
Ćwiczenie 4.2. Ćwiczenia praktyczne	201

Ćwiczenie 4.3. Sprawdź swoją wiedzę	201
Ćwiczenie 4.4. Dalsza nauka	201
Podsumowanie	201
ROZDZIAŁ 5	
Protokołowanie, śledzenie i metryki obserwowalności	203
Protokołowanie i śledzenie w .NET	203
Interfejs ILogger w .NET	205
Jak używać interfejsu ILogger?	207
Tworzenie usługi sieciowej do protokołowania	209
Testowanie podstawowych funkcji usługi sieciowej	214
Monitorowanie za pomocą metryk w .NET	215
Pojęcia dotyczące metryk i alertów	215
Implementowanie metryk	217
Wyświetlanie metryk	221
OpenTelemetry	222
Obsługiwane pakiety	223
Instrumentacja projektu ASP.NET Core	224
Przeglądanie danych telemetry	226
Ćwiczenia i dalsza nauka	230
Ćwiczenie 5.1. Materiały dostępne wyłącznie online	230
Ćwiczenie 5.2. Ćwiczenia praktyczne	230
Ćwiczenie 5.3. Sprawdź swoją wiedzę	230
Ćwiczenie 5.4. Dalsza nauka	231
Podsumowanie	231
ROZDZIAŁ 6	
Dokumentowanie kodu, API i serwisów	232
Wprowadzenie do dokumentacji	232
Korzyści z dokumentacji	233
Kiedy nie dokumentować?	234
Dokumentowanie kodu źródłowego	235

Kiedy należy dokumentować kod źródłowy?	235
Dobre praktyki komentowania kodu źródłowego	237
Dokumentowanie publicznych API w bibliotekach klas	237
Dokumentowanie za pomocą komentarzy XML-a	239
Generowanie dokumentacji przy użyciu DocFX	246
Dodawanie własnych treści do dokumentacji	253
Język znaczników Markdown	254
Dokumentowanie usług	257
Kluczowe kwestie przy dokumentowaniu usług	257
Narzędzia do dokumentowania usług	258
Poznanie specyfikacji OpenAPI (OAS)	259
Wykorzystanie OpenAPI do dokumentowania usługi Minimal API	260
Dokumentowanie wizualne za pomocą diagramów Mermaid	264
Rysowanie diagramów Mermaid	266
Schematy blokowe	267
Diagramy klas	268
Zapisywanie diagramów Mermaid w formacie SVG	270
Ćwiczenia i dalsza nauka	271
Ćwiczenie 6.1. Materiały dostępne wyłącznie online	271
Ćwiczenie 6.2. Ćwiczenia praktyczne	272
Ćwiczenie 6.3. Sprawdź swoją wiedzę	272
Ćwiczenie 6.4. Dalsza nauka	273
Podsumowanie	273
ROZDZIAŁ 7	
Obserwowanie kodu i dynamiczne wpływanie na jego wykonanie	274
Praca z refleksją i atrybutami	275
Metadane w zestawach .NET	275
Dynamiczne ładowanie zestawów i wykonywanie metod	283
Ostrzeżenie dotyczące refleksji i natywnego AOT	289
Ulepszenia refleksji w .NET 9	290

Więcej możliwości z refleksją	290
Praca z drzewami wyrażeń	291
Elementy drzewa wyrażeń	293
Wykonywanie najprostszego drzewa wyrażeń	293
Tworzenie generatorów kodu źródłowego	294
Implementacja najprostszego generatora źródeł	294
Praktyka i dalsza nauka	299
Ćwiczenie 7.1. Materiał dostępny wyłącznie online	299
Ćwiczenie 7.2. Ćwiczenia praktyczne	299
Ćwiczenie 7.3. Sprawdź swoją wiedzę	299
Ćwiczenie 7.4. Dalsza nauka	300
Podsumowanie	300
ROZDZIAŁ 8	
Ochrona danych i aplikacji za pomocą kryptografii	301
Terminologia związana z ochroną danych	302
Techniki ochrony danych	302
Klucze i ich rozmiary	303
Wektory inicjujące i rozmiary bloków	304
Sól	305
Generowanie kluczy i wektorów IV	305
Szyfrowanie i rozszyfrowywanie danych	306
Szyfrowanie symetryczne z użyciem AES	307
Haszowanie danych	313
Haszowanie z użyciem algorytmu SHA-256	314
Podpisywanie danych	318
Podpisywanie z wykorzystaniem SHA-256 i RSA	319
Generowanie liczb losowych na potrzeby kryptografii	321
Uwierzytelnianie i autoryzacja użytkowników	323
Mechanizmy uwierzytelniania i autoryzacji	324
Implementowanie uwierzytelniania i autoryzacji	326

Ochrona funkcji udostępnianych przez aplikację	329
Uwierzytelnianie i autoryzacja w rzeczywistych zastosowaniach	330
Co nowego w .NET 9?	330
Metoda <code>CryptographicOperations.HashData()</code>	330
Algorytm KMAC	331
Ćwiczenia i dalsza nauka	331
Ćwiczenie 8.1. Materiały dostępne wyłącznie online	331
Ćwiczenie 8.2. Ćwiczenia praktyczne	332
Ćwiczenie 8.3. Sprawdź swoją wiedzę	332
Ćwiczenie 8.4. Dalsza nauka	333
Podsumowanie	333
ROZDZIAŁ 9	
Tworzenie chatu używającego modelu LLM	334
Wprowadzenie do modeli LLM	335
Jak działają modele LLM?	335
Uzyskiwanie dostępu do modelu LLM	337
Używanie Semantic Kernel z modelem OpenAI	340
Czym jest Semantic Kernel?	341
Czym są funkcje?	347
Dodawanie funkcji	348
Dodanie pamięci sesji i włączanie wielu funkcji	353
Strumieniowanie wyników	355
Dodawanie protokołowania i zwiększanie niezawodności	356
Korzystanie z lokalnych modeli LLM	358
Hugging Face	358
Ollama	359
LM Studio	365
Ćwiczenia i dalsza nauka	367
Ćwiczenie 9.1. Materiały dostępne wyłącznie online	367
Ćwiczenie 9.2. Ćwiczenia praktyczne	368

Ćwiczenie 9.3. Sprawdź swoją wiedzę	369
Ćwiczenie 9.4. Dalsza lektura	369
Podsumowanie	369
ROZDZIAŁ 10	
Wstrzykiwanie zależności, kontenery i czas życia serwisów	370
Wprowadzenie do wstrzykiwania zależności	371
Dlaczego warto korzystać z DI?	372
Mechanizmy wstrzykiwania zależności w .NET	372
Przykłady we współczesnym .NET	374
Czas życia rejestrowanych zależności	376
Rejestrowanie wielu implementacji	377
Kiedy rzucają wyjątki?	378
Najlepsze praktyki wstrzykiwania zależności	379
Implementacja generycznego hosta w .NET	379
Najważniejsze funkcje generycznego hosta .NET	379
Budowanie generycznego hosta .NET	380
Poznanie usług i zdarzeń hosta	384
Metody rejestrowania usług	388
Grafy zależności i rozwiązywanie usług	389
Usuwanie usług	390
Wstrzykiwanie zależności w ASP.NET Core	390
Rejestrowanie usług dla funkcji za pomocą metod rozszerzających	391
Gdy nie można użyć wstrzykiwania przez konstruktor	391
Rozwiązywanie usług podczas uruchamiania aplikacji	393
Wstrzykiwanie zależności w widokach	393
Wstrzykiwanie do metod akcji i minimalnego API	394
Ćwiczenia i dalsza nauka	394
Ćwiczenie 10.1. Materiały dostępne wyłącznie online	394
Ćwiczenie 10.2. Ćwiczenia praktyczne	394
Ćwiczenie 10.3. Sprawdź swoją wiedzę	395

Ćwiczenie 10.4. Dalsza nauka	395
Podsumowanie	395
ROZDZIAŁ 11	
Testowanie i mockowanie	396
Wprowadzenie do wszystkich rodzajów testów	397
Testy jednostkowe	397
Testy integracyjne, end-to-end i bezpieczeństwa	398
Testy wydajności, obciążenia i wytrzymałości	398
Testy funkcjonalne i testy użyteczności	399
Terminologia testowania	399
Cechy wszystkich dobrych testów	399
Wyniki testów	401
Dublery, mocki i zatyczki	401
Przyjęcie odpowiedniego nastawienia do testowania	402
Zalety i wady TDD	403
Główne zasady TDD	403
Zalety TDD	403
Wady TDD	404
Dobre praktyki TDD	404
Testy jednostkowe	405
Jak bardzo izolowane powinny być testy jednostkowe?	405
Nazewnictwo używane w testach jednostkowych	406
Tworzenie testów jednostkowych za pomocą xUnit	406
Często używane atrybuty xUnit	408
Tworzenie klasy do przetestowania	409
Pisanie prostych testów jednostkowych	410
Metody testujące z parametrami	413
Pozytywne i negatywne wyniki testów	417
Sygnały ostrzegawcze w testach jednostkowych	418
Wyświetlanie wyników podczas wykonywania testów	418

Przygotowanie i czyszczenie środowiska testowego	419
Kontrola przygotowania testów	422
Mockowanie w testach	425
Biblioteki do mockowania	427
Wykorzystanie NSubstitute do tworzenia dubli	428
Przykład mockowania z użyciem NSubstitute	429
Tworzenie płynnych asercji w testach jednostkowych	432
Tworzenie asercji dla ciągów znaków	432
Tworzenie asercji dla kolekcji i tablic	433
Tworzenie asercji dla dat i godzin	434
Generowanie fałszywych danych z biblioteką Bogus	435
Projekt testujący fałszywe dane	437
Pisanie metody z użyciem fałszywych danych	439
Ćwiczenia i dalsza nauka	440
Ćwiczenie 11.1. Materiały dostępne wyłącznie online	441
Ćwiczenie 11.2. Ćwiczenia praktyczne	441
Ćwiczenie 11.3. Sprawdź swoją wiedzę	442
Ćwiczenie 11.4. Dalsza nauka	442
Podsumowanie	443
ROZDZIAŁ 12	
Testy integracyjne i testy bezpieczeństwa	444
Podstawy testów integracyjnych	445
Jakie systemy zewnętrzne testować?	446
Współdzielenie środowiska w testach integracyjnych	446
Analiza przykładowego testu integracyjnego	447
Testy integracyjne z wykorzystaniem magazynów danych	449
Deweloperskie instancje bazy danych i migracje baz danych	449
Cykl życia danych	451
Testowanie usług za pomocą tuneli deweloperskich	453
Instalowanie interfejsu wiersza poleceń tunelu deweloperskiego	454

Używanie wiersza poleceń tunelu deweloperskiego i serwisu echo	454
Używanie tunelu deweloperskiego w projekcie ASP.NET Core	456
Wprowadzanie do testów bezpieczeństwa	460
Open Web Application Security Project	462
OWASP Top 10	462
Modelowanie zagrożeń	467
Ćwiczenia i dalsza nauka	468
Ćwiczenie 12.1. Materiały dostępne wyłącznie online	469
Ćwiczenie 12.2. Ćwiczenia praktyczne	469
Ćwiczenie 12.3. Sprawdź swoją wiedzę	469
Ćwiczenie 12.4. Dalsza nauka	470
Podsumowanie	470
ROZDZIAŁ 13	
Mierzenie wydajności i testy obciążeniowe	471
Mierzenie wydajności	471
Znaczenie pomiaru bazowego	472
Notacja dużego O	473
Metryki statystyczne	475
Używanie biblioteki BenchmarkDotNet do mierzenia wydajności	476
Unikanie błędów podczas testów wydajności	482
Rozpoznawanie kiepskich postów na temat wydajności	485
Testy obciążeniowe i wytrzymałościowe	486
Apache JMeter	489
Bombardier — szybkie wieloplatformowe narzędzie do testów wydajności żądań HTTP	489
Używanie Bombardiera	490
Pobieranie Bombardiera	491
Porównywanie serwisu sieciowego kompilowanego za pomocą AOT i tradycyjnie	491
Interpretacja wyników Bombardiera	498

NBomber — framework do testów obciążeniowych	500
Scenariusze w NBomberze	500
Symulacje obciążenia	501
Typy NBombera	501
Przykład projektu NBombera	501
Ćwiczenia i dalsza nauka	505
Ćwiczenie 13.1. Materiały dostępne wyłącznie online	505
Ćwiczenie 13.2. Ćwiczenia praktyczne	506
Ćwiczenie 13.3. Sprawdź swoją wiedzę	506
Ćwiczenie 13.4. Dalsza nauka	506
Podsumowanie	507
ROZDZIAŁ 14	
Testy funkcjonalne i end-to-end	508
Testy funkcjonalne i end-to-end	508
Przykład 1. Testowanie usługi Web API	509
Przykład 2. Testowanie strony internetowej ASP.NET Core	509
Przykład 3. Testowanie aplikacji w czasie rzeczywistym SignalR	510
Testowanie webowych interfejsów użytkownika przy użyciu Playwrighta	511
Korzyści dla programistów .NET	512
Alternatywy dla Playwrighta	513
Typy testów w Playwrightcie	514
Metody automatyzacji strony w Playwrightcie	514
Metody lokalizacji elementów w Playwrightcie	515
Metody automatyzacji lokalizatorów w Playwrightcie	516
Testowanie typowych scenariuszy za pomocą aplikacji eShopOnWeb	517
Interakcja z interfejsem użytkownika	524
Wybieranie elementów z list rozwijanych i klikanie elementów	524
Przesyłanie formularzy, uwierzytelnianie i walidacja	526
Testowanie responsywnego projektu	526

Aplikacje SPA i dynamiczna zawartość	528
Generowanie testów z Playwright Inspector	529
Testowanie serwisów przy użyciu xUnit	533
Tworzenie serwisu internetowego na potrzeby testów	534
Tworzenie projektu testowego	535
Ćwiczenia i dalsza nauka	536
Ćwiczenie 14.1. Materiały dostępne wyłącznie online	536
Ćwiczenie 14.2. Ćwiczenia praktyczne	536
Ćwiczenie 14.3. Sprawdź swoją wiedzę	537
Ćwiczenie 14.4. Dalsza nauka	537
Podsumowanie	537
ROZDZIAŁ 15	
Konteneryzacja przy użyciu Dockera	538
Wprowadzenie do konteneryzacji	538
Jak działają kontenery i jakie są ich zalety?	540
Docker, Kubernetes i .NET Aspire	541
Rejestry kontenerów	543
Pojęcia związane z Dockerem	545
Narzędzia i technologie Dockera	546
Polecenia interfejsu wiersza poleceń Dockera	546
Tworzenie obrazów przy użyciu plików Dockerfile	548
Konfigurowanie portów i uruchamianie kontenera	551
Tryb interaktywny	552
Zmienne środowiskowe	553
Często używane obrazy kontenerów Dockera	554
Obrazy kontenerów .NET	554
System CVE i odchudzone Ubuntu	555
Zarządzanie kontenerami za pomocą Dockera	556
Instalowanie Dockera i używanie gotowych obrazów	556
Hierarchia obrazów Dockera i warstwy obrazów	559

Konteneryzacja własnych projektów .NET	563
Konteneryzacja projektu aplikacji konsoli	563
Publikowanie aplikacji do kontenera Dockera	565
Konteneryzacja projektu aplikacji ASP.NET Core	570
Praca z kontenerami testowymi	573
Jak działa Testcontainers dla .NET?	573
Przykład użycia	574
Ćwiczenia i dalsza nauka	575
Ćwiczenie 15.1. Materiały dostępne wyłącznie online	575
Ćwiczenie 15.2. Ćwiczenia praktyczne	575
Ćwiczenie 15.3. Sprawdź swoją wiedzę	575
Ćwiczenie 15.4. Dalsza nauka	576
Podsumowanie	576
ROZDZIAŁ 16	
Oprogramowanie chmurowe z .NET Aspire	577
Wprowadzenie do .NET Aspire	578
Co mówi zespół Aspire?	579
Aspire w edytorach kodu i w interfejsie wiersza poleceń	580
Uruchamianie rozwiązania Aspire	580
Typy projektów Aspire	581
Typy zasobów Aspire	582
Model aplikacji Aspire i orkiestracja	582
Szablony projektów Aspire	584
Szablon aplikacji startowej Aspire	585
Tworzenie aplikacji startowej Aspire	586
Przeglądanie rozwiązania startowego Aspire	589
Dokładniejsze spojrzenie na Aspire	592
Pulpit nawigacyjny dla deweloperów	592
Projekt AppHost — orkiestracja zasobów	593
Projekt ServiceDefaults — centralizacja konfiguracji	596

Inne projekty funkcyjne w rozwiązaniu	597
Konfigurowanie Redisa	599
Komponenty Aspire	600
Protokołowanie, śledzenie i metryki obserwowalności	602
Docker kontra Podman	602
Oczekiwanie na gotowość kontenerów	603
A co z Dapr, Orleans i Project Tye?	603
Aspire w nowych i istniejących rozwiązaniach	606
Tworzenie nowego rozwiązania Aspire	606
Aspire i PostgreSQL	611
Używanie woluminów danych i konfigurowanie stabilnego hasła	612
Dodanie Aspire do istniejącego rozwiązania	612
Aplikacja referencyjna eShop	614
Wdrażanie aplikacji za pomocą Aspire	619
Ćwiczenia i dalsza nauka	620
Ćwiczenie 16.1. Materiały dostępne wyłącznie online	621
Ćwiczenie 16.2. Ćwiczenia praktyczne	621
Ćwiczenie 16.3. Sprawdź swoją wiedzę	622
Ćwiczenie 16.4. Dalsza nauka	622
Podsumowanie	622
ROZDZIAŁ 17	
Wzorce i zasady projektowe	623
Zasady SOLID	624
Zasada pojedynczej odpowiedzialności (Single Responsibility Principle — SRP)	624
Zasada otwarte — zamknięte (Open/Closed Principle — OCP)	627
Zasada podstawień Liskov (Liskov Substitution Principle — LSP)	629
Zasada segregacji interfejsów (Interface Segregation Principle — ISP)	633
Zasada odwracania zależności (Dependency Inversion Principle — DIP)	637

Wzorce projektowe	640
Wzorce kreatywne	645
Strukturalne wzorce projektowe	647
Behawioralne wzorce projektowe	650
Zasady projektowe	653
Zasada DRY	653
Zasada KISS	653
YAGNI	654
Prawo Demeter	655
Kompozycja jest lepsza od dziedziczenia	656
Zasada najmniejszego zaskoczenia	658
Algorytmy i struktury danych	660
Algorytmy sortowania	661
Algorytmy wyszukiwania	661
Algorytmy struktury danych	662
Algorytmy haszujące	662
Algorytmy rekurencyjne	662
Gdzie dowiesz się więcej o algorytmach i strukturach danych?	663
Ćwiczenie i dalsza nauka	663
Ćwiczenie 17.1. Materiały dostępne wyłącznie online	663
Ćwiczenie 17.2. Ćwiczenia praktyczne	663
Ćwiczenie 17.3. Sprawdź swoją wiedzę	664
Ćwiczenie 17.4. Dalsza nauka	664
Podsumowanie	664
ROZDZIAŁ 18	
Podstawy architektury rozwiązań i oprogramowania	665
Wprowadzenie do architektury oprogramowania i architektury rozwiązań	665
Architektura oprogramowania	666
Architektura rozwiązań	667
Koncepcje architektury oprogramowania	667

Style architektury oprogramowania	669
Koncepcje architektury rozwiązania	675
Wnioski	676
Czysta architektura według Wujka Boba	677
Koncepcje czystej architektury	678
Dobre praktyki w czystej architekturze w .NET	680
Rysowanie diagramów za pomocą Mermaid	681
Mermaid w architekturze oprogramowania i rozwiązań	681
Rodzaje diagramów Mermaid	682
Schematy blokowe w Mermaid	683
Ćwiczenie i dalsza nauka	691
Ćwiczenie 18.1. Materiał dostępny wyłącznie online	692
Ćwiczenie 18.2. Ćwiczenia praktyczne	692
Ćwiczenie 18.3. Sprawdź swoją wiedzę	692
Ćwiczenie 18.4. Dalsza nauka	693
Podsumowanie	694
ROZDZIAŁ 19	
Kariera, praca zespołowa i rozmowy kwalifikacyjne	695
Praca w zespole programistycznym	695
Praca inżyniera oprogramowania	696
Ścieżka kariery	697
Role osób w zespole programistycznym, z którymi będziesz współpracować	698
Proces onboardingu	701
Jak poprosić o szkolenie i rozwój?	701
Programowanie w parach	704
Poszukiwanie pracy	707
Przed złożeniem podania	707
Przygotowanie do rozmowy kwalifikacyjnej	710
Przykładowe pytania na rozmowie kwalifikacyjnej	730

1. Narzędzia wiersza poleceń w .NET	731
2. Podstawy Gita	732
3. Entity Framework Core	734
4. Interfejsy i klasy abstrakcyjne	736
5. Właściwości i indeksery	736
6. Typy generyczne	736
7. Delegaty i zdarzenia	736
8. LINQ	736
9. Programowanie asynchroniczne z użyciem słów kluczowych async i await	737
10. Zarządzanie pamięcią i jej oczyszczanie	737
11. Różnice między nowoczesnym .NET a .NET Framework	737
12. Możliwości wieloplatformowe	737
13. .NET Standard	737
14. Wstrzykiwanie zależności w .NET	737
15. Oprogramowanie pośredniczące w ASP.NET Core	738
16. Konfiguracja i wzorzec Opcje	738
17. Hosting i serwer Kestrel	738
18. Typy danych	738
19. Globalizacja i lokalizacja	738
18 Zostań ekspertem .NET 8	
20. Struktury sterujące	738
21. Obsługa wyjątków	738
22. Strategie tworzenia gałęzi w Gicie	739
23. Przeglądy kodu i programowanie w parach	739
24. Metodyki agile i scrum	739
25. Standardy dokumentacji	739
26. Umiejętność rozwiązywania problemów	739
27. Narzędzia do zarządzania projektami	739
28. Techniki szacowania	739

29. Współpraca w zespole	740
30. Przywództwo i mentoring	740
31. Wzorzec MVC	740
32. Składnia Razora	740
33. Web API	740
34. Najlepsze praktyki dotyczące usług RESTful	740
35. SignalR do pracy w czasie rzeczywistym	740
36. Zarządzanie stanem	741
37. Uwierzytelnianie i autoryzacja	741
38. Blazor WebAssembly	741
39. Korzyści wynikające z używania mikroserwisów	741
40. Wyzwania związane z architekturą mikroserwisów	741
41. Kontenery Dockera i .NET	741
42. Wzorce komunikacji w mikroserwisach	741
43. Odporność na błędy i obsługa błędów przejściowych	742
44. Śledzenie rozproszone	742
45. Kontrola stanu zdrowia i monitorowanie aplikacji	742
46. AutoMapper, metody rozszerzające i operator jawny	742
47. Podstawy ADO.NET	742
48. Optymalizacja wydajności Entity Framework Core	742
49. Frameworki do testów jednostkowych, np. xUnit	743
50. Frameworki do mockowania, np. NSubstitute	743
51. Strategie testowania integracyjnego	743
52. Testowanie wydajności	743
53. Testowanie bezpieczeństwa	743
54. Automatyczne testowanie interfejsu użytkownika	743
55. Zasady SOLID	743
56. Wzorzec projektowy Singleton	744
57. Wzorzec projektowy Fabryka	744
58. Wykrywanie wycieków pamięci	744

59. Metodyki programistyczne	744
60. Notacja wielkiego O	744
Uczysz się, kiedy popełniasz błędy	744
Ćwiczenia i dalsza nauka	745
Ćwiczenie 19.1. Materiały dostępne wyłącznie online	745
Ćwiczenie 19.2. Ćwiczenia praktyczne	745
Ćwiczenie 19.3. Sprawdź swoją wiedzę	745
Ćwiczenie 19.4. Dalsza nauka	746
Podsumowanie	746
ROZDZIAŁ 20	
Epilog	747
Kolejne kroki na ścieżce nauki .NET	747
Książki uzupełniające, które pomogą kontynuować naukę	747
Dziewiąte wydanie C# 12 i .NET 8 już wkrótce dla .NET 9	748
Planowana trylogia .NET 10	749
Książki do dalszej nauki	750
Powodzenia!	750
Skorowidz	751