

Table of Contents

Preface	xxiii
----------------	--------------

Section 1: Principles and Methodologies

Chapter 1: Introduction	1
--------------------------------	----------

What is a design pattern?	2
---------------------------------	---

Anti-patterns and code smells	3
-------------------------------------	---

Anti-patterns • 3

Anti-pattern – God Class • 3

Code smells • 4

Code smell – Control Freak • 4

Code smell – Long Methods • 4

Understanding the web – request/response	5
--	---

Getting started with .NET	9
---------------------------------	---

 .NET SDK versus runtime • 9

 .NET 5+ versus .NET Standard • 9

 Visual Studio Code versus Visual Studio versus the command-line interface • 10

An overview of project templates • 11

Running and building your program • 12

 Technical requirements • 12

Summary	13
---------------	----

Questions	13
-----------------	----

Further reading	13
-----------------------	----

Chapter 2: Automated Testing	15
-------------------------------------	-----------

An overview of automated testing	15
--	----

 Unit testing • 16

Integration testing • 16	
End-to-end testing • 17	
Other types of tests • 17	
Picking the right test style • 18	
Testing approaches • 19	
Refactoring • 20	
Technical debt • 20	
Testing .NET applications	21
Creating an xUnit test project • 21	
Getting started with xUnit • 21	
<i>Facts • 21</i>	
<i>Assertions • 23</i>	
<i>Theories • 25</i>	
<i>Closing words • 30</i>	
Arrange, Act, Assert • 31	
Organizing your tests • 31	
<i>Unit tests • 32</i>	
<i>Integration tests • 34</i>	
ASP.NET Core integration testing • 35	
<i>Classic web application • 35</i>	
<i>Minimal hosting • 37</i>	
Important testing principles	39
Summary	40
Questions	40
Further reading	40
 Chapter 3: Architectural Principles	 41
The SOLID principles	41
Single responsibility principle (SRP) • 42	
<i>Project – BookStore • 42</i>	
<i>What is an interface? • 52</i>	
Open/Closed principle (OCP) • 53	
<i>Project – IAttacker • 53</i>	
Liskov substitution principle (LSP) • 62	
<i>Project – HallOfFame • 64</i>	
<i>Conclusion • 73</i>	
Interface segregation principle (ISP) • 73	

<i>Project – Ninja versus Pirate</i> • 74	
<i>Project – Bookstore update</i> • 77	
<i>Conclusion</i> • 80	
Dependency inversion principle (DIP) • 81	
<i>Direct dependency</i> • 81	
<i>Inverted dependency</i> • 81	
<i>Inverting subsystems using DIP</i> • 82	
<i>Project – Dependency inversion</i> • 83	
<i>Conclusion</i> • 86	
<i>What's next?</i> • 86	
Other important principles	87
Separation of concerns • 87	
Don't repeat yourself (DRY) • 88	
Keep it simple, stupid (KISS) • 89	
Summary	89
Questions	90
Further reading	90

Section 2: Designing for ASP.NET Core

Chapter 4: The MVC Pattern Using Razor	93
The Model View Controller design pattern	93
Goal • 94	
Design • 94	
Anatomy of ASP.NET Core MVC • 95	
<i>Directory structure</i> • 96	
<i>Controller</i> • 97	
<i>Model</i> • 98	
<i>View</i> • 99	
<i>Default routing</i> • 100	
Conclusion • 101	
The View Model design pattern	101
Goal • 101	
Design • 102	
Project – View models (a list of students) • 103	
Conclusion • 105	
Summary	106

Questions	106
Further reading	106

Chapter 5: The MVC Pattern for Web APIs **107**

An overview of REST	107
----------------------------------	------------

HTTP methods • 108

Status code • 108

HTTP headers • 109

Versioning • 110

Default versioning strategy • 110

Versioning strategy • 111

Wrapping up • 111

The Model View Controller design pattern	111
---	------------

Goal • 112

Design • 112

Anatomy of ASP.NET Core web APIs • 113

The entry point • 113

Directory structure • 114

Controller • 114

Returning values • 115

Attribute routing • 117

Conclusion • 119

The Data Transfer Object design pattern	119
--	------------

Goal • 119

Design • 120

Project – DTO • 120

Conclusion • 127

API contracts	128
----------------------------	------------

Summary	131
----------------------	------------

Questions	132
------------------------	------------

Further reading	132
------------------------------	------------

Chapter 6: Understanding the Strategy, Abstract Factory, and Singleton Design Patterns **133**

The Strategy design pattern	133
--	------------

Goal • 134

Design • 134

Project – Strategy • 135	
Conclusion • 140	
The Abstract Factory design pattern	141
Goal • 141	
Design • 141	
Project – AbstractVehicleFactory • 143	
Conclusion • 147	
The Singleton design pattern	148
Goal • 148	
Design • 149	
An alternate (better) way • 151	
Code smell – Ambient Context • 152	
Conclusion • 154	
Summary	155
Questions	155
 Chapter 7: Deep Dive into Dependency Injection	 157
What is dependency injection?	157
The composition root • 159	
Registering your features elegantly • 159	
Object lifetime • 161	
Code smell – Control Freak • 163	
Using external IoC containers	164
Revisiting the Strategy pattern	166
Constructor injection • 166	
Property injection • 166	
Method injection • 167	
Project – Strategy • 168	
<i>Adding the View Model • 170</i>	
<i>Adding a guard clause • 171</i>	
Conclusion • 172	
Revisiting the Singleton pattern	173
Project – Application state • 173	
Project – Wishlist • 178	
Conclusion • 184	
Understanding the Service Locator pattern	184
Project – ServiceLocator • 184	

<i>Implementing method injection</i> • 187	
<i>Implementing constructor injection</i> • 187	
<i>Implementing a minimal API</i> • 188	
Conclusion • 188	
Revisiting the Factory pattern	189
Project – Factory • 189	
Conclusion • 192	
Summary	192
Questions	193
Further reading	193
 Chapter 8: Options and Logging Patterns	 195
The Options pattern	195
Getting started • 195	
<i>IOptionsMonitor<TOptions></i> • 197	
<i>IOptionsFactory<TOptions></i> • 197	
<i>IOptionsSnapshot<TOptions></i> • 197	
<i>IOptions<TOptions></i> • 198	
Project – CommonScenarios • 198	
<i>Named options</i> • 201	
<i>Using settings</i> • 204	
Project – OptionsConfiguration • 206	
<i>Implementing a configurator object</i> • 208	
<i>Using multiple configurator objects</i> • 209	
<i>Exploring other configuration possibilities</i> • 210	
Project – OptionsValidation • 211	
<i>Eager validation</i> • 211	
<i>Data annotations</i> • 212	
<i>Validation types</i> • 213	
Project – OptionsValidationFluentValidation • 216	
Workaround – Injecting options directly • 218	
Conclusion • 220	
Becoming familiar with .NET logging abstractions	221
About logging • 221	
Writing logs • 222	
Log levels • 227	
Logging providers • 229	

Configuring logging • 229

Structured logging • 233

Conclusion • 234

Summary 234

Questions 235

Further reading 235

Section 3: Designing at Component Scale

Chapter 9: Structural Patterns 239

Implementing the Decorator design pattern 239

Goal • 239

Design • 240

Project – Adding behaviors • 243

DecoratorA • 244

DecoratorB • 245

Project – Decorator using Scrutor • 246

Conclusion • 248

Implementing the Composite design pattern 249

Goal • 249

Design • 249

Project – BookStore • 250

Conclusion • 261

Implementing the Adapter design pattern 261

Goal • 261

Design • 262

Project – Greeter • 263

Conclusion • 265

Implementing the Façade design pattern 265

Goal • 265

Design • 266

Project – The façades • 267

Opaque façade • 267

Transparent façade • 270

The program • 273

Flexibility in action • 275

Alternative façade patterns • 276

Conclusion • 276	
Summary	277
Questions	278
Further reading	278
Chapter 10: Behavioral Patterns	279
Implementing the Template Method pattern	279
Goal • 279	
Design • 279	
Project – Building a search machine • 280	
Conclusion • 284	
Implementing the Chain of Responsibility pattern	285
Goal • 285	
Design • 285	
Project – Message interpreter • 286	
Project – Improved message interpreter • 291	
Project – A final, finer-grained design • 294	
Conclusion • 298	
Summary	299
Questions	299
Chapter 11: Understanding the Operation Result Design Pattern	301
The Operation Result pattern	301
Goal • 302	
Design • 302	
Project – Implementing different Operation Result patterns • 303	
<i>The consumer • 303</i>	
<i>The simplest form of the Operation Result pattern • 304</i>	
<i>A single error message • 305</i>	
<i>Adding a return value • 307</i>	
<i>Multiple error messages • 308</i>	
<i>Adding message severity • 310</i>	
<i>Sub-classes and factories • 315</i>	
Advantages and disadvantages • 318	
<i>Advantages • 318</i>	
<i>Disadvantages • 318</i>	
Summary	319

Questions	320
Further reading	320

Section 4: Designing at Application Scale

Chapter 12: Understanding Layering **323**

Introducing layering	323
Classic layering model • 324	
Splitting the layers • 327	
Layers versus tiers versus assemblies • 328	
<i>What is an assembly?</i> • 328	
Responsibilities of the common layers	329
Presentation • 329	
Domain • 329	
<i>Rich domain model</i> • 330	
<i>Anemic domain model</i> • 331	
<i>Service layer</i> • 333	
Data • 334	
<i>Repository pattern</i> • 335	
<i>Unit of Work pattern</i> • 337	
Abstract layers	339
Sharing the model	342
Clean Architecture	346
Implementing layering in real life	351
To be or not to be a purist? • 351	
Building a façade over a database • 352	
Summary	354
Questions	355
Further reading	355

Chapter 13: Getting Started with Object Mappers **357**

Object mapper	357
Goal • 358	
Design • 358	
Project – Mapper • 359	
Code smell – Too many dependencies • 364	

Pattern – Aggregate Services • 365	
Pattern – Mapping Façade • 367	
Project – Mapping service • 368	
Project – AutoMapper • 371	
Conclusion • 378	
Summary	378
Questions	379
Further reading	379
Chapter 14: Mediator and CQRS Design Patterns	381
A high-level overview of Vertical Slice Architecture	381
Implementing the Mediator pattern	383
Goal • 383	
Design • 383	
Project – Mediator (IMediator) • 385	
Project – Mediator (IChatRoom) • 389	
Conclusion • 394	
Implementing the CQRS pattern	394
Goal • 395	
Design • 395	
Project – CQRS • 396	
Code smell – Marker Interfaces • 410	
<i>Metadata • 410</i>	
<i>Dependency identifier • 411</i>	
Conclusion • 414	
Using MediatR as a mediator	415
Project – Clean Architecture with MediatR • 415	
Conclusion • 421	
Summary	421
Questions	422
Further reading	422
Chapter 15: Getting Started with Vertical Slice Architecture	423
Vertical Slice Architecture	423
What are the advantages and disadvantages? • 425	
Anti-pattern – Big Ball of Mud • 427	
Project – Vertical Slice Architecture • 427	

<i>Project organization</i> • 428	
<i>Exploring a feature</i> • 430	
<i>Request validation</i> • 437	
<i>Testing</i> • 442	
Continuing your journey • 446	
Conclusion • 448	
Summary	448
Questions	449
Further reading	449
Chapter 16: Introduction to Microservices Architecture	451
What are microservices?	452
Cohesive unit of business • 452	
Ownership of data • 453	
Microservice independence • 453	
An introduction to event-driven architecture	454
Domain events • 455	
Integration events • 455	
Application events • 455	
Enterprise events • 455	
Conclusion • 456	
Getting started with message queues	457
Conclusion • 459	
Implementing the Publish-Subscribe pattern	460
Message brokers • 463	
The event sourcing pattern • 463	
Example • 465	
Conclusion • 473	
Introducing Gateway patterns	474
Gateway Routing pattern • 475	
Gateway Aggregation pattern • 478	
Backends for Frontends pattern • 480	
Mixing and matching gateways • 481	
Conclusion • 482	
Revisiting the CQRS pattern	484
Conclusion • 487	

Exploring the Microservice Adapter pattern	488
Adapting an existing system to another • 489	
Decommissioning a legacy application • 491	
Adapting an event broker to another • 496	
Conclusion • 498	
Summary	499
Questions	500
Further reading	500

Section 5: Designing the Client Side

Chapter 17: ASP.NET Core User Interfaces 503

Getting familiar with Razor Pages	503
Design • 505	
Routing • 508	
Conclusion • 508	
Organizing the user interface	509
Partial views • 509	
<i>Project – Shared form • 510</i>	
<i>Conclusion • 513</i>	
Tag Helpers • 513	
<i>Built-in Tag Helpers • 515</i>	
<i>Creating a custom Tag Helper • 524</i>	
<i>Creating an RSS feed TagHelperComponent • 526</i>	
<i>Conclusion • 529</i>	
View components • 529	
<i>Project – Reusable employee count • 530</i>	
<i>Conclusion • 532</i>	
Display and editor templates	533
Display templates • 533	
Editor templates • 535	
Project – Composite BookStore revisited • 537	
Conclusion • 543	
Summary	544
Questions	544
Further reading	545

Chapter 18: A Brief Look into Blazor 547

Overview of Blazor Server	548
Overview of Blazor WebAssembly	549
Getting familiar with Razor components	550
Creating Razor components • 551	
<i>C#-only components • 551</i>	
<i>Razor-only components • 552</i>	
<i>Razor and C# hybrid components • 553</i>	
CSS isolation • 554	
Component life cycle • 556	
Event handling • 558	
The Model-View-Update pattern	560
Goal • 561	
Design • 561	
Project – Counter • 562	
Conclusion • 566	
A medley of Blazor features	566
Summary	569
Questions	569
Further reading	570
An end is simply a new beginning	571

Appendices

Appendix A 575

Older C# features	575
The null-coalescing operator (C# 2.0) • 575	
Expression-bodied member (C# 6-7) • 576	
Throw expressions (C# 7.0) • 577	
Tuples (C# 7.0+) • 578	
Default literal expressions (C# 7.1) • 582	
Switch expressions (C# 8) • 583	
Discards (C# 7) • 584	
Async main (C# 7.1) • 585	
User-defined conversion operators (C# 1) • 586	
Local functions (C# 7) and a static local function (C# 8) • 588	

What's new in .NET 5 and C# 9?	592
Top-level statements • 592	
Target-typed new expressions • 593	
Init-only properties • 594	
Record classes • 594	
<i>Simplifying the record creation</i> • 595	
<i>The with keyword</i> • 596	
<i>Deconstruction</i> • 596	
<i>Equality comparison</i> • 597	
<i>Conclusion</i> • 598	
What's new in .NET 6 and C# 10?	598
File-scoped namespaces • 599	
Global using directives • 599	
Implicit using directives • 600	
Constant interpolated strings • 601	
Record struct • 601	
Minimal hosting model • 602	
Minimal APIs • 603	
Nullable reference types • 605	
Appendix B	607
An overview of containers	607
Docker	608
Docker Compose • 611	
Orchestration	615
Project Tye • 616	
Kubernetes • 616	
Scaling	618
Summary	619
Further reading	619
Assessment Answers	621
Chapter 1	621
Chapter 2	621
Chapter 3	621
Chapter 4	621
Chapter 5	622

Chapter 6	622
Chapter 7	622
Chapter 8	622
Chapter 9	623
Chapter 10	623
Chapter 11	623
Chapter 12	623
Chapter 13	624
Chapter 14	624
Chapter 15	624
Chapter 16	625
Chapter 17	625
Chapter 18	625

Acronyms Lexicon	627
-------------------------	------------

Other Books You May Enjoy	631
----------------------------------	------------

Index	635
--------------	------------
