

Паттерны проектирования

для C# и платформы .NET CORE

Гаурав Арораа, Джеффри Чилберто



 ПИТЕР®

Packt>

Hands-On Design Patterns with C# and .NET Core

Write clean and maintainable code by using reusable solutions to common software design problems

Gaurav Arora
Jeffrey Chilberto

Packt>

BIRMINGHAM - MUMBAI

Паттерны проектирования

ДЛЯ C# И ПЛАТФОРМЫ .NET CORE

Гаурав Арораа
Джеффри Чилберто



Санкт-Петербург • Москва • Екатеринбург • Воронеж
Нижний Новгород • Ростов-на-Дону • Самара • Минск

2021

ББК 32.973.2-018-02
УДК 004.42
А84

Арораа Гаурав, Чилберто Джеффри

А84 Паттерны проектирования для C# и платформы .NET Core. — СПб.: Питер, 2021. — 352 с.: ил. — (Серия «Для профессионалов»).

ISBN 978-5-4461-1523-5

Паттерны проектирования — удобный прием программирования для решения рутинных задач разработки ПО. Грамотное использование паттернов позволяет добиться соответствия любым требованиям и снизить расходы. В этой книге описаны эффективные способы применения паттернов проектирования с учетом специфики языка C# и платформы .NET Core.

Кроме знакомых паттернов проектирования из книги «Банды четырех» вы изучите основы объектно-ориентированного программирования и принципов SOLID. Затем узнаете о функциональных, реактивных и конкурентных паттернах, с помощью которых будете работать с потоками и корутинами. Заключительная часть содержит паттерны для работы с микросервисными, бессерверными и облачно-ориентированными приложениями. Вы также узнаете, как сделать выбор архитектуры, например микросервисной или MVC.

16+ (В соответствии с Федеральным законом от 29 декабря 2010 г. № 436-ФЗ.)

ББК 32.973.2-018-02
УДК 004.42

Права на издание получены по соглашению с Packt Publishing. Все права защищены. Никакая часть данной книги не может быть воспроизведена в какой бы то ни было форме без письменного разрешения владельцев авторских прав.

Информация, содержащаяся в данной книге, получена из источников, рассматриваемых издательством как надежные. Тем не менее, имея в виду возможные человеческие или технические ошибки, издательство не может гарантировать абсолютную точность и полноту приводимых сведений и не несет ответственности за возможные ошибки, связанные с использованием книги. Издательство не несет ответственности за доступность материалов, ссылки на которые вы можете найти в этой книге. На момент подготовки книги к изданию все ссылки на интернет-ресурсы были действующими.

ISBN 978-1789133646 англ.

© Packt Publishing 2019.

First published in the English language under the title 'Hands-On Design Patterns with C# and .NET Core — (9781789133646)

ISBN 978-5-4461-1523-5

© Перевод на русский язык ООО Издательство «Питер», 2021

© Издание на русском языке, оформление ООО Издательство «Питер», 2021

© Серия «Для профессионалов», 2021

Краткое содержание

Предисловие	14
Об авторах	16
О научных редакторах.....	17
Введение.....	18

Часть I. Основы паттернов проектирования в C# и .NET Core

Глава 1. Обзор ООП в .NET Core и C#.....	24
Глава 2. Современные паттерны и принципы проектирования ПО	42

Часть II. Углубленное изучение утилит и паттернов .NET Core

Глава 3. Реализация паттернов проектирования — основы (часть 1).....	70
Глава 4. Реализация паттернов проектирования — основы (часть 2).....	100
Глава 5. Реализация паттернов проектирования в .NET Core.....	134
Глава 6. Реализация паттернов проектирования для веб-приложений (часть 1)	157
Глава 7. Реализация паттернов проектирования для веб-приложений (часть 2)	178

Часть III. Функциональное программирование, реактивное программирование и кодирование для облака

Глава 8. Конкурентное программирование в .NET Core	212
Глава 9. Функциональное программирование.....	229
Глава 10. Модели и методы реактивного программирования	245
Глава 11. Усовершенствованные методы проектирования и применения баз данных	284
Глава 12. Разработка облачных приложений	309

Приложения

Приложение А. Дополнительные практические рекомендации	336
Приложение Б. Ответы на вопросы	343

Оглавление

Предисловие	14
Об авторах	16
О научных редакторах.....	17
Введение.....	18
Кому подойдет эта книга	18
Структура издания.....	19
Как получить максимум от книги.....	20
Загрузка файлов с примерами	21
Код в действии.....	21
Полноцветные изображения	21
Условные обозначения.....	21
От издательства.....	22

Часть I. Основы паттернов проектирования в C# и .NET Core

Глава 1. Обзор ООП в .NET Core и C#.....	24
Технические требования.....	25
Установка Visual Studio.....	25
Установка .NET Core.....	25
Используемые в книге модели	26
Определение ООП. Как работают классы и объекты.....	28
Определение ООП	28
Класс.....	29
Объект.....	30
Интерфейс	33
Наследование.....	34
Инкапсуляция	36

Полиморфизм	36
Статический полиморфизм.....	37
Динамический полиморфизм.....	38
Резюме.....	40
Вопросы	41
Глава 2. Современные паттерны и принципы проектирования ПО	42
Технические требования.....	43
Установка Visual Studio.....	43
Установка .NET Core.....	43
Принципы проектирования.....	44
DRY — «Не повторяйся».....	44
KISS — «Делай проще, тупица»	44
YAGNI — «Вам это не понадобится»	45
MVP — «Продукт с минимальным функционалом».....	45
SOLID	45
Паттерны программного обеспечения	47
Паттерны «Банды четырех»	47
Паттерны интеграции корпоративных приложений.....	59
Паттерны жизненного цикла разработки программного обеспечения	65
Waterfall SDLC	65
Agile SDLC	66
Резюме.....	67
Вопросы	68
 Часть II. Углубленное изучение утилит и паттернов .NET Core	
Глава 3. Реализация паттернов проектирования — основы (часть 1).....	70
Технические требования.....	71
Установка Visual Studio.....	71
Установка .NET Core.....	71
Продукт с минимальным функционалом	71
Требования.....	72
Каковы преимущества MVP в дальнейшей разработке.....	74
Разработка через тестирование	75
Почему разработчики выбирают TDD.....	76
Настройка проектов.....	77
Определения начальных модульных тестов.....	80
Паттерн проектирования «Абстрактная фабрика»	82

Принципы SOLID	87
Принцип единственной ответственности (SRP)	87
Принцип открытости/закрытости (ОСР)	88
Принцип замещения Лисков (LSP)	89
Принцип сегрегации интерфейса (ISP)	90
Принцип инверсии зависимостей	91
Модульные тесты для InventoryCommand	93
Резюме	98
Вопросы	99
Глава 4. Реализация паттернов проектирования — основы (часть 2)	100
Технические требования	101
Установка Visual Studio	101
Установка .NET Core	101
Паттерн «Синглтон»	101
Процессы и потоки	102
Паттерн «Репозиторий»	104
Модульные тесты	105
Иллюстрация состояния гонки	112
Класс AddInventoryCommand	115
Класс UpdateQuantityCommand	119
Команда GetInventoryCommand	121
Паттерн «Фабрика»	123
Модульные тесты	124
Функциональность .NET Core	128
Интерфейс IServiceCollection	128
Интерфейс CatalogService	129
IServiceProvider	130
Консольное приложение	131
Резюме	133
Вопросы	133
Глава 5. Реализация паттернов проектирования в .NET Core	134
Технические требования	134
Установка Visual Studio	135
Установка .NET Core	135
Время жизни сервисов в .NET Core	135
Временная зависимость (Transient)	136
Время жизни области применения (Scoped)	136

«Одиночка» в .NET Core	136
Вернемся к FlixOne	136
Что такое время жизни области применения	142
Реализующая фабрика.....	143
Интерфейс IInventoryContext.....	144
Интерфейс IInventoryReadContext	144
Интерфейс IInventoryWriteContext	145
Класс InventoryCommandFactory	146
Класс InventoryCommand	147
Резюме	155
Вопросы	156
Глава 6. Реализация паттернов проектирования для веб-приложений (часть 1)	157
Технические требования.....	157
Установка Visual Studio.....	158
Установка .NET Core	158
Установка SQL Server	158
Создание веб-приложения .NET Core.....	159
Запуск проекта.....	159
Разработка веб-приложения	160
Реализация CRUD-страниц	166
Резюме	176
Вопросы	177
Дальнейшее чтение	177
Глава 7. Реализация паттернов проектирования для веб-приложений (часть 2)	178
Технические требования.....	178
Установка Visual Studio.....	179
Установка .NET Core	179
Установка SQL Server	179
Расширение веб-приложения .NET Core	180
Начало проекта	180
Аутентификация и авторизация	182
Аутентификация в действии.....	187
Авторизация в действии	197
Создание тестового проекта веб-приложения	205
Резюме	209
Вопросы	209
Дальнейшее чтение	210

Часть III. Функциональное программирование, реактивное программирование и кодирование для облака

Глава 8. Конкурентное программирование в .NET Core	212
Технические требования.....	212
Установка Visual Studio.....	213
Установка .NET Core.....	213
Установка SQL Server	213
Конкурентность в реальном мире.....	214
Многопоточное и асинхронное программирование.....	216
async/await — чем плоха блокировка	219
Конкурентная коллекция	220
Паттерны и рекомендации — TDD и параллельный LINQ.....	221
Резюме.....	227
Вопросы	228
Дальнейшее чтение	228
Глава 9. Функциональное программирование.....	229
Технические требования.....	229
Установка Visual Studio.....	230
Установка .NET Core.....	230
Установка SQL Server	230
Основы функционального программирования	231
Совершенствование приложения FlixOne	235
Требования.....	236
Вернемся к FlixOne.....	237
Паттерн «Стратегия» и функциональное программирование	242
Резюме.....	243
Вопросы	244
Глава 10. Модели и методы реактивного программирования	245
Технические требования.....	245
Установка Visual Studio.....	246
Установка .NET Core.....	246
Установка SQL Server	246
Принципы реактивного программирования.....	247
Будьте реактивны с реактивным программированием	249
Реактивность и интерфейс IObservable.....	257
Паттерн «Наблюдатель» — реализация с помощью IObservable<T>	257

Реактивные расширения: .NET Rx Extensions	264
Настройка приложения FlixOne	266
Начало проекта	266
Применение в приложении FlixOne фильтрации, пагинации и сортировки....	267
Паттерны и практики — MVVM	275
Реализация MVVM	277
Резюме	282
Вопросы	283
Дальнейшее чтение	283
Глава 11. Усовершенствованные методы проектирования и применения	
баз данных	284
Технические требования	284
Установка Visual Studio	285
Установка .NET Core	285
Установка SQL Server	285
Обсуждение примера использования	286
Начало проекта	286
Требования	287
Поговорим о базах данных	289
Обработка баз данных	289
OLTP	290
OLAP	291
Базы данных в стиле бухгалтерской книги	291
Реализация паттерна CQRS	293
Резюме	307
Вопросы	308
Глава 12. Разработка облачных приложений	309
Технические требования	310
Ключевые моменты, которые следует учитывать при разработке	
облачных решений	310
Масштабируемость	311
Рабочая нагрузка	311
Паттерны	312
Устойчивость/доступность	319
Паттерны решений	320
Безопасность	323
Паттерны решений	323

Проектирование приложения	325
Паттерны решений	325
DevOps	330
Паттерны решений	330
Резюме	333
Вопросы	334
Дальнейшее чтение	334

Приложения

Приложение А. Дополнительные практические рекомендации	336
Технические требования	336
Обсуждение примера использования	337
UML-диаграмма	338
Практические рекомендации	339
Другие паттерны проектирования	340
Резюме	341
Вопросы	342
Дальнейшее чтение	342
Приложение Б. Ответы на вопросы	343
Глава 1. Обзор ООП в .NET Core и C#	343
Глава 2. Современные паттерны и принципы проектирования ПО	343
Глава 3. Реализация паттернов проектирования — основы (часть 1)	344
Глава 4. Реализация паттернов проектирования — основы (часть 2)	345
Глава 5. Реализация паттернов проектирования в .NET Core	345
Глава 6. Реализация паттернов проектирования для веб-приложений (часть 1)	346
Глава 7. Реализация паттернов проектирования для веб-приложений (часть 2)	346
Глава 8. Конкурентное программирование в .NET Core	347
Глава 9. Функциональное программирование	348
Глава 10. Модели и методы реактивного программирования	348
Глава 11. Усовершенствованные методы проектирования и применения баз данных	350
Глава 12. Разработка облачных приложений	350
Приложение А. Практические рекомендации	351

*Моей матери Лате Шримати Сантош
и в память о моем отце, покойном
Ш. Рамкришане, за их жертвенность и твердость
духа. Моей младшей сестренке, также покойной
крошке Канчан, за ее любовь и за то, что была
моим маленьким талисманом.*

Гаурав Арораа

*Моим родителям Френсис и Джойс которые
неустанно поднимали на ноги своих детей
с любовью, заботой и добротой. Моим братьям:
Джеку — за мотивацию сохранять спокойствие
в череде испытаний и Майку — за напоминание
остановиться и насладиться жизнью.*

Джеффри Чилберто

Предисловие

При разработке качественного ПО программисты стремятся избегать дублирования кода. Мы весьма часто применяем прием DRY — Don't Repeat Yourself («Не повторяйся») — даже не задумываясь! Разработчики обычно разделяют функциональность, создают многократно используемые методы и пишут вспомогательные классы. Паттерны проектирования создавались и обновлялись годами. Это полезные, общепринятые и пригодные к многократному применению решения повседневных проблем.

Гаурав и Джефффри собрали лучшие и наиболее широко используемые паттерны и применили их в мире открытого C# и .NET Core. Вы начнете с ООП, классов, объектов и будете продвигаться далее к наследованию, инкапсуляции и полиморфизму. Авторы описали такие подходы, как DRY, KISS и SOLID (вы очень скоро поймете, что все это значит!), и применили их к классическим паттернам, которые помогут вам разрабатывать чистое и надежное программное обеспечение.

Книга содержит примеры рабочего кода, которые покажут, как использовать получаемые знания в создании ПО на .NET Core и C# в наши дни. Вы освоите паттерны «Строитель», «Декоратор», «Фабрика», «Посетитель», «Стратегия» и многие другие.

Эти методики будут применены сначала к простой программе, а затем к веб-приложениям. Далее будут рассмотрены более сложные темы, включая конкурентность и параллелизм. Наконец, вы сможете воспользоваться паттернами на принципиально более высоком уровне, применяя решения, которые помогут вам переместить проекты в облако, где их можно будет масштабировать и легко обслуживать.

Надеюсь, что, как и я, вы по достоинству оцените эту книгу. Желаю получать удовольствие, работая с .NET Core, так же, как получали его мы, когда разрабатывали эту среду!

*Скотт Хансельман,
руководитель партнерской программы
Microsoft .NET и Open Source Community*

Эта книга — библия для каждого разработчика, желающего не только улучшить свои навыки программирования, но и, самое главное, научиться создавать надежные, масштабируемые и удобные в сопровождении решения. Здесь описана большая часть практических рекомендаций, дополненных наглядными примерами.

Помимо паттернов проектирования, книга затрагивает архитектурные принципы и ключевые моменты работы в облаке, такие как безопасность и масштабирование.

Будучи архитектором решений, я ежедневно проектирую комплексные приложения, уделяя особое внимание разработке инфраструктур и инструментов, и все же поражаюсь качеству контента и охвату тем, представленных в этой книге. В ней содержится исчерпывающий список паттернов, рекомендуемых к ознакомлению всем, кто связан с разработкой и развертыванием ПО.

Такое полное и глубокое погружение в мир объектно-ориентированного программирования (ООП) и .NET Core полезно всем, кто заинтересован в разработке лучших приложений.

Стефани Айскенс, Azure MVP

Об авторах

Гаурав Арора получил степень магистра в области компьютерных наук, имеет сертификаты Microsoft MVP, Alibaba Cloud MVP, тренера/коуча по Scrum. Член *Компьютерного сообщества Индии (CSI)*, сотрудник IndiaMentor, сертифицированный специалист ITIL Foundation, APMG PRINCE-F и APMG PRINCE-P. Гаурав — разработчик открытого программного обеспечения, внесший вклад в развитие TechNet Wiki и основавший компанию Ovatic Systems Private Limited. За свою более чем 20-летнюю карьеру он обучил тысячи студентов. Вы можете написать Гаураву в Twitter по адресу [@g_arora](https://twitter.com/g_arora).

Спасибо моей жене Шуби Арора и моему ангелу (дочери) Аачи Арора, которые вытерпели мое отсутствие, пока я писал эту книгу. Спасибо всей команде Packt, особенно Чайтаньи, Акиште и Нее, координационная деятельность и коммуникативность которых были так необходимы, а также Дениму Пинто, порекомендовавшему меня в качестве потенциального автора.

Джеффри Чилберто — консультант в сфере разработки программного обеспечения, специализирующийся на технологическом стеке Microsoft, включая Azure, BizTalk, ASP.NET, MVC, WCF и SQL Server, с опытом работы в различных сферах. Он участвовал в разработке банковских, телекоммуникационных и медицинских систем в Новой Зеландии, Австралии и США. Имеет степень бакалавра в области информационных и компьютерных наук, а также степень магистра по информационным технологиям и вычислительной технике.

Хотел бы поблагодарить мою семью за любовь, поддержку и вдохновение.

О научных редакторах

Сьеки Цааль — консультант по управлению, архитектор облачных систем Microsoft и Microsoft Azure MVP с более чем 15-летним стажем в области создания архитектур, разработки, консультирования и системного дизайна. Она работает в Sargemini — одной из крупнейших в мире консалтинговых компаний в сфере менеджмента и информационных технологий. Сьеки любит делиться знаниями и принимает очень активное участие в сообществе Microsoft как одна из основателей нидерландских пользовательских групп SP&C NL и MixUG. Кроме того, является членом правления Azure Thursdays. Сьеки часто выступает с лекциями и участвует в организации мероприятий. Она написала несколько книг, ведет блоги и принимает активное участие в сообществе Microsoft Tech Community (<https://techcommunity.microsoft.com/>).

У *Эфраима Кирякидиса* за плечами более 20 лет опыта в разработке ПО. Он получил диплом инженера в Университете имени Аристотеля в Салониках (Греция). Эфраим пользуется .NET с момента создания, с версии 1.0. В своей работе он в целом сфокусирован на технологиях Microsoft. В данный момент занимает пост старшего инженера в компании Siemens AG в Германии.

Введение

В этой книге на конкретных примерах описываются способы использования паттернов в современной разработке приложений. Количество паттернов, применяемых для поиска решений, огромно. Чаще всего разработчики пользуются ими, не понимая в полной мере, что и как эти паттерны делают. В издании рассматриваются паттерны от низкоуровневого кода до высокоуровневых концептуальных решений, которые работают в облачных системах.

Несмотря на то что большинство паттернов не требуют применения конкретного языка, для иллюстрации многих из них мы используем C# и .NET Core. Эти технологии были выбраны из-за их популярности и удобства архитектуры, на основе которой можно создавать различные решения — от простых консольных приложений до крупных корпоративных распределенных систем.

Описывая большое количество паттернов, эта книга знакомит со многими из них, позволяет глубже понять их на практике. Представленные паттерны были выбраны, чтобы проиллюстрировать определенные аспекты проектирования. Кроме того, добавлены ссылки на дополнительные материалы, которые помогут читателю подробнее изучить конкретный интересующий его паттерн.

И в простых сайтах, и в огромных корпоративных системах правильно подобранный паттерн может в корне изменить заведомо неверное решение, ведущее к краху проекта из-за его высокой стоимости и низкой производительности, и превратить это решение в выгодное, долгоживущее и успешное. Паттерны, описанные в книге, призваны преодолеть неизбежные проблемы, которые иначе не позволили бы вам оставаться конкурентоспособными. Они также позволяют вашим приложениям достичь устойчивости и надежности, которыми должны обладать современные проекты.

Кому подойдет эта книга

Целевая аудитория — разработчики современных приложений. Поскольку книга содержит много кода, чтобы объяснить, как и где используются паттерны, подразумевается наличие у читателя опыта в разработке ПО. Не стоит рассматривать данную книгу как «программирование для чайников», скорее она отвечает на во-

прос «Как программировать лучше». Поэтому издание будет полезно начинающим и опытным программистам, архитекторам приложений и проектировщикам.

Структура издания

Глава 1 описывает объектно-ориентированное программирование и его применение в среде C#. Эта глава служит напоминанием о важных конструкциях и функциях ООП и C#, включающих наследование, инкапсуляцию и полиморфизм.

Глава 2 каталогизирует и представляет различные паттерны, используемые в современной разработке программного обеспечения. Эта глава исследует ряд паттернов и их каталогов, таких как SOLID, паттерны «Банды четырех» и паттерны корпоративной интеграции. Вдобавок в ней обсуждаются жизненный цикл и другие методики разработки ПО.

Глава 3 погрузит в паттерны проектирования, используемые для создания приложений на C#. На примере приложения-образца будут показаны разработка через тестирование, концепция продукта с минимальным функционалом и некоторые паттерны «Банды четырех».

Глава 4 продолжит знакомить с паттернами, используемыми в приложениях на C#. Будут введены такие понятия, как внедрение зависимости и инверсия управления, а также продолжится изучение паттернов проектирования, включая «Одиночку» и «Фабрику».

Глава 5 основана на главах 3 и 4 и описывает паттерны в .NET Core. Некоторые паттерны, включая «Внедрение зависимости» и «Фабрику», будут также пересмотрены с учетом среды .NET Core.

Глава 6 продолжает знакомить с .NET Core, описывая функции, поддерживаемые в разработке веб-приложений, в процессе создания тестового приложения. Эта глава содержит руководство по созданию базового веб-приложения, представляет важные характеристики, которыми оно должно обладать, и рассказывает, как создавать веб-страницы с поддержкой CRUD.

Глава 7 тоже посвящена разработке веб-приложений с помощью .NET Core и представляет различные архитектурные паттерны, а также варианты решений вопроса безопасности. Кроме того, описывает аутентификацию и авторизацию, а также модульные тесты с использованием Moq — фреймворка для создания имитаций реальных объектов.

Глава 8 углубляется в разработку веб-приложений и описывает понятие «конкурентность» при разработке приложений на C# и .NET Core. Будет рассмотрен паттерн `async/await`, а также многопоточность и конкурентность, которым посвящен

отдельный раздел. Кроме того, речь пойдет о Parallel LINQ с отложенным запуском и приоритетом потоков.

Глава 9 описывает функциональное программирование в .NET Core. Сюда входят функции языка C#, поддерживающие функциональное программирование, и их применение в тестовом приложении вместе с паттерном «Стратегия».

Глава 10 продолжает тему разработки веб-приложений .NET Core с помощью паттернов реактивного программирования и методик, используемых для создания адаптивных и масштабируемых сайтов. Глава описывает принципы реактивного программирования, включая паттерны **Reactive** и **IObservable**. Кроме того, рассматривает различные фреймворки, включая популярный проект .NET Rx Extensions, а также демонстрирует паттерн «*Модель — представление — модель представления*» (Model — View — View Model, MVVM).

Глава 11 освещает паттерны, используемые в проектировании баз данных, а также описывает сами БД. Будет показан практический пример применения паттерна CQRS, в том числе с помощью проектирования базы данных в виде журнала учета.

Глава 12 представляет разработку приложений в контексте облачных решений, включая пять ключевых аспектов: масштабирование, доступность, безопасность, проектирование приложений и DevOps. Будут описаны важнейшие паттерны, используемые в облачно-ориентированных решениях, включая различные типы масштабирования, и представлены паттерны, применяемые для событийно-ориентированных архитектур, интегрированной безопасности, кэша и телеметрии.

Приложение А содержит обсуждение дополнительных паттернов и практические рекомендации. Сюда включены разделы о моделировании примеров использования, рекомендованных методик и дополнительных паттернов, таких как пространственная архитектура и контейнерные приложения.

Приложение Б дает ответы на вопросы, размещенные в конце каждой главы.

Как получить максимум от книги

Предполагается, что вы уже немного знакомы с принципами ООП и языком C#. В книге описываются продвинутые темы, однако материал нельзя воспринимать как пошаговую инструкцию к применению. Цель издания — улучшить навыки разработчиков и проектировщиков с помощью широкого спектра паттернов, методик и принципов. По аналогии с ящиком для инструментов книга предлагает современным разработчикам эти самые инструменты для перехода от низкоуровневого проектирования к созданию более высокоуровневых архитектур, а также важные паттерны и принципы, широко используемые в настоящий момент.

Вдобавок эта книга обращает внимание на следующие моменты, призванные дополнить знания читателя:

- ❑ знание принципов SOLID, практические рекомендации с примерами кода на C#7.x и NET Core 2.2;
- ❑ углубленное понимание классических паттернов проектирования (от «Банды четырех»);
- ❑ принципы функционального программирования и рабочие примеры на языке C#;
- ❑ реальные примеры архитектурных паттернов (MVC, MVVM);
- ❑ понимание облачной ориентированности, микросервисов и пр.

Загрузка файлов с примерами

Вы можете скачать примеры из этой книги на GitHub по ссылке github.com/PacktPublishing/Hands-On-Design-Patterns-with-C-and-.NET-Core. В случае обновления кода примеры также будут обновлены в репозитории.

Код в действии

Перейдите по ссылке bit.ly/2KuuNgQ, чтобы просмотреть видео с примерами кода.

Полноцветные изображения

Мы также предоставляем PDF-файл с цветными изображениями схем и рисунков, используемыми в оригинале данной книги: static.packt-cdn.com/downloads/9781789133646_ColorImages.pdf.

Условные обозначения

В книге используется ряд условных обозначений.

Моноширинным шрифтом обозначены код в тексте, имена таблиц баз данных и т. п. Например: «В данном классе есть три метода: `CounterA()`, `CounterB()` и `CounterC()`, представляющих отдельную стойку для получения билетов».

Блок кода (листинг) оформляется так:

```
3-counters are serving...
Next person from row
Person A is collecting ticket from Counter A
Person B is collecting ticket from Counter B
Person C is collecting ticket from Counter C
```

Когда мы хотим обратить ваше внимание на конкретную часть листинга, то необходимые и ключевые строки выделяем **полужирным моноширинным** шрифтом:

```
public bool UpdateQuantity(string name, int quantity)
{
    lock (_lock)
    {
        _books[name].Quantity += quantity;
    }

    return true;
}
```

Любой ввод и вывод в оболочке командной строки выглядит следующим образом:

```
dotnet new sln
```

Курсивным шрифтом обозначены новые термины и важные слова.

URL, команды меню, элементы управления и заголовки окон выделяются **шрифтом без засечек**, например: «Команда **Create New Product** (Создать новый продукт) позволит добавить новый продукт, а **Edit** (Редактировать) — изменять и обновлять уже имеющиеся продукты».



Важные замечания и предупреждения обозначены так.



Советы и подсказки обозначены таким образом.

От издательства

Ваши замечания, предложения, вопросы отправляйте по адресу comp@piter.com (издательство «Питер», компьютерная редакция).

Мы будем рады узнать ваше мнение!

На веб-сайте издательства www.piter.com вы найдете подробную информацию о наших книгах.