

УДК 004.43
ББК 32.973.26-018.2
Р60

Jort Rodenburg

CODE LIKE A PRO IN C#

Original English language edition published by Manning Publications USA.

Copyright © 2021 by Manning Publications.

Russian-language edition copyright © 2024 by Eksmo Publishing House.

All rights reserved.

Роденбург, Йорт.

Р60 Пиши код как профи на C# / Йорт Роденбург ; [перевод с английского М. А. Райтмана]. — Москва : Эксмо, 2025. — 512 с. — (Manning: профессиональные книги для ИТ-специалистов).

ISBN 978-5-04-186609-9

«Пиши код как профи на C#» — это практическое руководство для разработчиков, желающих углубить навыки программирования на C#. Книга предназначена для тех, кто уже владеет основами программирования на объектно-ориентированных языках, таких как Java, и стремится перенести эти знания в мир C# и .NET. Вы узнаете, как использовать передовые подходы и инструменты для создания эффективного кода, а также получите навыки работы с корпоративными приложениями.

Внимание сосредоточено на переработке и оптимизации устаревшей кодовой базы. Вы будете применять актуальные методы и практики для обновления кода в соответствии с современными стандартами C#.

УДК 004.43
ББК 32.973.26-018.2

ISBN 978-5-04-186609-9

© Райтман М.А., перевод на русский язык, 2025
© Оформление. ООО «Издательство «Эксмо», 2025

Все права защищены. Книга или любая ее часть не может быть скопирована, воспроизведена в электронной или механической форме, в виде фотокопии, записи в память ЭВМ, репродукции или каким-либо иным способом, а также использована в любой информационной системе без получения разрешения от издателя. Копирование, воспроизведение и иное использование книги или ее части без согласия издателя является незаконным и влечет уголовную, административную и гражданскую ответственность.

Производственно-практическое издание
MANNING: ПРОФЕССИОНАЛЬНЫЕ КНИГИ ДЛЯ ИТ-СПЕЦИАЛИСТОВ

Роденбург Йорт
ПИШИ КОД КАК ПРОФИ НА С#

Главный редактор *Р. Фасхутдинов*
Руководитель направления *В. Обручев*
Продюсер *В. Палитко*
Ответственный редактор *Л. Салихова*
Научный редактор *Р. Егорочкин*
Литературный редактор *Ю. Войтко*
Младший редактор *П. Смирнов*
Художественный редактор *К. Доброслов*
Компьютерная верстка *Э. Брегис*
Корректоры *Ю. Киреева, А. Ярлыкова*

Страна происхождения: Российская Федерация
Шығарушы ел: Ресей Федерациясы

ООО «Издательство «Эксмо»
123308, Россия, г. Москва, ул. Зорге, д. 1, стр. 1, эт. 20, каб. 2013. Тел.: 8 (495) 411-68-86.
Home page: www.eksmo.ru E-mail: info@eksmo.ru
Өндіруші: «Издательство «Эксмо» ЖШҚ
123308, Ресей, Мәскеу қаласы, Зорге көшесі, 1-үй, 1-құрылыс, 20 қабат, 2013-каб.
Тел.: 8 (495) 411-68-86. Home page: www.eksmo.ru E-mail: info@eksmo.ru.
Тауар белгісі: «Эксмо»

Интернет-магазин : www.book24.ru
Интернет-магазин : www.book24.kz
Интернет-дуken : www.book24.kz

Импортер в Республику Казахстан ТОО «РДЦ-Алматы».
Қазақстан Республикасына импорттаушы «РДЦ-Алматы» ЖШС.
Дистрибьютор и представитель по приему претензий на продукцию
в Республике Казахстан: ТОО «РДЦ-Алматы»
Дистрибьютор және Қазақстан Республикасында өнімге шағымдар
қабылдау жөніндегі өкіл: «РДЦ-Алматы» ЖШС.

Алматы қ., Домбровский көш., 3 «а», литер Б, офис 1.
Тел.: 8 (727) 251-59-90/91/92. E-mail: RDC-Almaty@eksmo.kz

Сведения о подтверждении соответствия издания согласно законодательству РФ
о техническом регулировании можно получить на сайте Издательства «Эксмо»:
www.eksmo.ru/certification

Техникалық реттеу туралы РФ заңнамасына сай басылымның сәйкестігін растау
туралы мәліметтерді мына адрес бойынша алуға болады: <http://eksmo.ru/certification/>

Произведено в Российской Федерации
Ресей Федерациясында өндірілген
Сертификаттауға жатпайды

Дата изготовления / Подписано в печать 22.11.2024.
Формат 70x100^{1/16}. Печать офсетная. Усл. печ. л. 41,48.
Тираж экз. Заказ

ISBN 978-5-04-186609-9



9 785041 866099 >

12+

Москва. ООО «Торговый Дом «Эксмо»

Адрес: 123308, г. Москва, ул. Зорге, д.1, строение 1.
Телефон: +7 (495) 411-50-74. **E-mail:** reception@eksmo-sale.ru

По вопросам приобретения книг «Эксмо» зарубежными оптовыми покупателями обращаться в отдел зарубежных продаж ТД «Эксмо»
E-mail: international@eksmo-sale.ru

International Sales: International wholesale customers should contact Foreign Sales Department of Trading House «Eksmo» for their orders.
international@eksmo-sale.ru

По вопросам заказа книг корпоративным клиентам, в том числе в специальном оформлении, обращаться по тел.: +7 (495) 411-68-59, доб. 2151.
E-mail: borodkin.da@eksmo.ru

Оптовая торговля бумажно-беловыми и канцелярскими товарами для школы и офиса «Канц-Эксмо»:
Компания «Канц-Эксмо»: 142702, Московская обл., Ленинский р-н, г. Видное-2, Белокаменное ш., д. 1, а/я 5. Тел./факс: +7 (495) 745-28-87 (многоканальный).
e-mail: kanc@eksmo-sale.ru, сайт: **www.kanc-eksmo.ru**

Филиал «Торгового Дома «Эксмо» в Нижнем Новгороде

Адрес: 603094, г. Нижний Новгород, улица Карпинского, д. 29, бизнес-парк «Грин Плаза»
Телефон: +7 (831) 216-15-91 (92, 93, 94). **E-mail:** reception@eksmonn.ru

Филиал ООО «Издательство «Эксмо» в г. Санкт-Петербурге

Адрес: 192029, г. Санкт-Петербург, пр. Обуховской обороны, д. 84, лит. «Е»
Телефон: +7 (812) 365-46-03 / 04. **E-mail:** server@szko.ru

Филиал ООО «Издательство «Эксмо» в г. Екатеринбург

Адрес: 620024, г. Екатеринбург, ул. Новинская, д. 2щ
Телефон: +7 (343) 272-72-01 (02/03/04/05/06/08)

Филиал ООО «Издательство «Эксмо» в г. Самаре

Адрес: 443052, г. Самара, пр-т Кирова, д. 75/1, лит. «Е»
Телефон: +7 (846) 207-55-50. **E-mail:** RDC-samara@mail.ru

Филиал ООО «Издательство «Эксмо» в г. Ростове-на-Дону

Адрес: 344023, г. Ростов-на-Дону, ул. Страны Советов, 44А
Телефон: +7(863) 303-62-10. **E-mail:** info@rnd.eksmo.ru

Филиал ООО «Издательство «Эксмо» в г. Новосибирске

Адрес: 630015, г. Новосибирск, Комбинатский пер., д. 3
Телефон: +7(383) 289-91-42. **E-mail:** eksmo-nsk@yandex.ru

Обособленное подразделение в г. Хабаровске

Фактический адрес: 680000, г. Хабаровск, ул. Фрунзе, 22, оф. 703
Почтовый адрес: 680020, г. Хабаровск, А/Я 1006
Телефон: (4212) 910-120, 910-211. **E-mail:** eksmo-khv@mail.ru

Республика Беларусь: ООО «ЭКСМО АСТ Си энд Си»

Центр оптово-розничных продаж Cash&Carry в г. Минске
Адрес: 220014, Республика Беларусь, г. Минск, проспект Жукова, 44, пом. 1-17, ТЦ «Outlet»
Телефон: +375 17 251-40-23; +375 44 581-81-92
Режим работы: с 10.00 до 22.00. **E-mail:** exmoast@yandex.by

Казахстан: «РДЦ Алматы»

Адрес: 050039, г. Алматы, ул. Домбровского, 3А
Телефон: +7 (727) 251-58-12, 251-59-90 (91,92,99). **E-mail:** RDC-Almaty@eksmo.kz

Полный ассортимент продукции ООО «Издательство «Эксмо» можно приобрести в книжных магазинах «Читай-город» и заказать в интернет-магазине: www.chитай-gorod.ru.

Телефон единой справочной службы: 8 (800) 444-8-444. Звонок по России бесплатный.

Интернет-магазин ООО «Издательство «Эксмо»

www.eksmo.ru

Розничная продажа книг с доставкой по всему миру.
Тел.: +7 (495) 745-89-14. **E-mail: imarket@eksmo-sale.ru**

Литрес

Я ТАК ЧИТАЮ



ТЕРИТОРИЯ
КНИЖНЫЙ МАГАЗИН
Официальная франшиза
издательства «Эксмо»



Хочешь стать
автором «Эксмо»?



БОМБОРА – лидер на рынке полезных и вдохновляющих книг.
Мы любим книги и создаем их, чтобы вы могли творить, открывать
мир, пробовать новое, расти. Быть счастливыми. Быть на волне.

[@ bombora.ru](#) [bomborabooks](#) [bombora](#)



eksmo.ru

Официальный
интернет-магазин
издательства «Эксмо»

Оглавление

<i>Предисловие</i>	11
<i>Благодарности</i>	13
<i>Об этой книге</i>	15
Кому стоит прочитать эту книгу	15
Структура книги	16
О коде	18
<i>Об авторе</i>	20
<i>Об иллюстрации на обложке</i>	21

Часть I. Использование C# и .NET 23

<i>Глава 1. Знакомство с C# и .NET</i>	<i>25</i>
1.1. Преимущества C#	27
<i>Причина первая: финансовый фактор</i> 28 ■ <i>Причина</i> <i>вторая: сопровождаемость</i> 28 ■ <i>Причина третья: удобство</i> <i>и простота в использовании</i> 30	
1.2. Когда C# не подойдет?	31
<i>Разработка операционных систем</i> 31 ■ <i>Разработка</i> <i>встроенных RTOS-ориентированных</i> <i>приложений</i> 32 ■ <i>Численные вычисления</i> 33	
1.3. Переход на C#	34
1.4. Чему вы научитесь благодаря этой книге	37
1.5. Чего вы не узнаете из этой книги	38

Глава 2. .NET и компиляция	42
2.1. Что такое .NET Framework?	43
2.2. Что такое .NET 5?	44
2.3. Как компилируются языки, совместимые с CLI	46
<i>Этап первый: код на C# (высокий уровень)</i>	48
<i>Этап второй: промежуточный CIL-код (сборка)</i>	52
<i>Этап третий: нативный код (уровень процессора)</i>	60
Часть II. Исходная кодовая база	65
Глава 3. Недостатки исходного кода	67
3.1. Авиакомпания Flying Dutchman Airlines	68
3.2. Фрагменты пазла: обзор требований	70
<i>Объектно-реляционное отображение</i>	71
<i>Конечная точка GET /flight: получение информации обо всех рейсах</i>	72
<i>Конечная точка GET /flight/{flightNumber}: Получение информации о конкретном рейсе</i>	73
<i>Конечная точка POST /booking/{flightNumber}: бронирование рейса</i>	74
3.3. Анализ существующей кодовой базы	76
<i>Анализ схемы и таблиц существующей БД</i>	76
<i>Исходная кодовая база. Файлы конфигурации веб-службы</i>	78
<i>Обзор моделей и представлений в исходной кодовой базе</i>	87
Глава 4. Контроль над неуправляемыми ресурсами	97
4.1. Класс FlightController: конечная точка GET /flight	99
<i>Конечная точка GET /flight в действии</i>	99
<i>Сигнатура метода: предназначение атрибута ResponseType и оператора typeof</i>	101
<i>Сбор информации о рейсе с помощью коллекций</i>	103
<i>Строки подключения, или как довести ИБ-специалиста до белого каления</i>	105
<i>Применение интерфейса IDisposable для освобождения неуправляемых ресурсов</i>	106
<i>Запрос к БД с помощью команды SqlCommand</i>	108
4.2. Класс FlightController: конечная точка GET /flight/{flightNumber}	113
4.3. Класс FlightController: конечная точка POST /flight	117
4.4. Класс FlightController: конечная точка DELETE /flight/{flightNumber}	122

Часть III. Уровень доступа к БД 127

Глава 5. Настройка проекта и БД с помощью

<i>Entity Framework Core</i>	129
5.1. Создание решения и проекта .NET 5	131
5.2. Установка и настройка веб-службы	135
<i>Конфигурирование веб-службы .NET 5</i> 136 ■ <i>Создание и использование класса HostBuilder</i> 139 ■ <i>Реализация класса Startup</i> 142 ■ <i>Применение шаблона проектирования Repository/Service («репозиторий/служба») к архитектуре веб-службы</i> 146	
5.3. Реализация уровня доступа к БД	147
<i>Entity Framework Core и «обратная разработка»</i> 148 ■ <i>Тип DbSet и рабочий процесс Entity Framework Core</i> 151 ■ <i>Методы конфигурации и переменные среды</i> 154 ■ <i>Настройка переменной среды в Windows</i> 156 ■ <i>Настройка переменной среды в macOS</i> 156 ■ <i>Взаимодействие с переменными среды во время выполнения</i> 157	

Часть IV. Уровень репозитория 163

Глава 6. Разработка через тестирование и внедрение зависимостей

.....	165
6.1. Разработка через тестирование	167
6.2. Метод CreateCustomer	172
<i>Почему вы должны всегда проверять аргументы ввода</i> 174	
<i>Применение шаблона AAA при разработке модульных тестов</i> 175 ■ <i>Проверка допустимости вводимых символов</i> 176 ■ <i>Встроенное тестирование с атрибутом [DataRow]</i> 180 ■ <i>Инициализаторы объектов и автогенерируемый код</i> 181 ■ <i>Конструкторы, рефлексия и асинхронное программирование</i> 183 ■ <i>Блокировка, мьютексы и семафоры</i> 187 ■ <i>Синхронное и асинхронное выполнение (продолжение)</i> 189 ■ <i>Тестирование Entity Framework Core</i> 191 ■ <i>Управление зависимостями с помощью внедрения зависимостей</i> 193	

Глава 7. Сравнение объектов

7.1. Метод GetCustomerByName	204
<i>Вопросительные знаки: нулевые типы и их применение</i> 207 ■ <i>Пользовательские исключения, LINQ и методы расширения</i> 208	

7.2. Конгруэнтность: от Средневековья до C#	214
Создание класса-компаратора с помощью <i>EqualityComparer<T></i> 216 ■ Проверка равенства путем переопределения метода <i>Equals</i> 220 ■ Перегрузка оператора равенства 221	
Глава 8. Заглушки, обобщения и связанность	229
8.1. Реализация репозитория <i>Booking</i>	230
8.2. Проверка входных данных, разделение ответственности и связанность	233
8.3. Инициализаторы объектов	241
8.4. Модульное тестирование с заглушками	244
8.5. Программирование с обобщениями	249
8.6. Предоставление аргументов по умолчанию с помощью необязательных параметров	252
8.7. Условные выражения, тип <i>Func</i> , переключатели и выражения <i>switch</i>	254
Тернарный условный оператор 255 ■ Ветвление с использованием массива функций 256 ■ Операторы и выражения <i>switch</i> 257 ■ Запрос на ожидающие изменения в <i>Entity Framework Core</i> 259	
Глава 9. Методы расширения, потоки и абстрактные классы	266
9.1. Реализация <i>AirportRepository</i>	268
9.2. Получение аэропорта из БД по его идентификатору	269
9.3. Проверка входного параметра <i>AirportID</i>	272
9.4. Выходные потоки и конкретная абстракция	273
9.5. Запрос в БД объекта <i>Airport</i>	280
9.6. Реализация <i>FlightRepository</i>	290
Метод расширения <i>IsPositive</i> и «магические числа» 292 ■ Получение рейса из БД 300	
Часть V. Уровень службы	305
Глава 10. Рефлексия и фиктивные объекты	307
10.1. Шаблон проектирования «репозиторий/служба»: новый взгляд	308
Для чего нужен класс службы? 309	

10.2. Реализация <code>CustomerService</code>	311
<i>Настройтесь на успех: создание шаблонов классов 311 ■ Как удалить собственный код 314</i>	
10.3. Реализация <code>BookingService</code>	317
<i>Модульное тестирование на всех архитектурных уровнях 323 ■ Разница между заглушкой и фиктивным объектом 325 ■ Применение фиктивного объекта к классу с помощью библиотеки <code>Moq</code> 326 ■ Вызов репозитория из службы 335</i>	
Глава 11. Динамический анализ типа и обработка ошибок ..	341
11.1. Проверка входных параметров метода уровня службы	342
<i>Проверка типа с помощью операторов <code>is</code> и <code>as</code> 347 ■ Проверка типа с помощью оператора <code>is</code> 348 ■ Проверка типа с помощью оператора <code>as</code> 349 ■ Что мы сделали в разделе 11.1? 350</i>	
11.2. Завершение класса <code>BookingServiceTests</code>	351
11.3. Ограничения внешних ключей в классах службы	354
<i>Вызов <code>FlightRepository</code> из класса службы 355</i>	
Глава 12. Интерфейс <code>IAsyncEnumerable<T></code> и оператор <code>yield return</code>	372
12.1. Нужен ли нам класс <code>AirportService</code> ?	373
12.2. Реализация класса <code>FlightService</code>	375
<i>Получение информации о конкретном рейсе из репозитория <code>FlightRepository</code> 375 ■ Объединение двух потоков данных в представление 380 ■ Ключевые слова <code>yield return</code> с блоками <code>try-catch</code> 391 ■ Реализация <code>GetFlightByFlightNumber</code> 396</i>	
Часть VI. Уровень контроллера	407
Глава 13. Промежуточное ПО, HTTP-маршрутизация и HTTP-ответы	409
13.1. Класс контроллера в шаблоне «репозиторий/служба»	410
13.2. Определяем, какие контроллеры необходимо реализовать	413
13.3. Реализация <code>FlightController</code>	414
<i>Возврат HTTP-ответов с помощью интерфейса <code>IActionResult (GetFlights)</code> 416 ■ Внедрение</i>	

зависимостей в контроллер с помощью промежуточного
ПО 420 ■ Реализация конечной точки GET /Flight/
{FlightNumber} 430

13.4. Маршрутизация HTTP-запросов к контроллерам
и методам 436

Глава 14. Сериализация/десериализация JSON

и пользовательская привязка моделей 444

14.1. Реализация класса BookingController 445

Введение в десериализацию данных 447 ■ Атрибут
[FromBody] для десериализации входящих данных
HTTP 452 ■ Пользовательская привязка модели и атрибут
метода для привязки модели 454 ■ Реализация логики метода
конечной точки CreateBooking 458

14.2. Приемочное тестирование
и промежуточное ПО Swagger 465

Ручное приемочное тестирование со спецификацией
OpenAPI 466 ■ Генерация спецификации OpenAPI во время
выполнения 471

14.3. Конец пути 478

Приложение А. Ответы на упражнения 480

Приложение Б. Памятка по чистому коду 488

Приложение В. Руководства по установке 491

Приложение Г. OpenAPI FlyTomorrow 496

Приложение Д. Список литературы 500

Алфавитный указатель 506

Предисловие

Я познакомился с C# в 2016 году, когда устроился в компанию Fujifilm Medical Systems (FMS). Я умел программировать на Java и Python, но, освоив C#, в дальнейшем пользовался только им. Мне понравились легкость обучения и (поначалу страшно раздражающий) упор на явную типизацию. Работая в FMS, я досаждал коллегам вопросами о C# и о том, как лучше писать на нем код. Начать было легко, а вот стать опытным специалистом — совсем другое дело. Приложение вида Hello, World! кто угодно напишет за 10 минут, независимо от уровня подготовки, а вот на то, чтобы освоить все сильные стороны языка и понять, почему те или иные моменты реализованы именно так, уйдет немало времени. Несколько позже я заметил, что мой уровень владения C# перестал повышаться, и принялся искать ресурсы, которые принесли бы мне новые знания. Вскоре я понял, что существует три основных типа книг, посвященных .NET и C#: 1) руководства по темам, выходящим за рамки самого языка C# (чистота кода, архитектура, инфраструктура и т. п.), 2) самоучители программирования на C# и 3) настолько продвинутые издания, что, прочитав их, вы легко займете должность технического директора Microsoft. Я же искал нечто среднее между вышеперечисленными типами: труд, где описывался бы чистый код, который заполнил бы промежуток между простейшими и продвинутыми темами. Такой книги на тот момент не существовало, поэтому я решил написать ее. И вот она у вас в руках.

Если вы — инженер-программист (разработчик, кодер, или как там вас называют в штатном расписании) с опытом работы на одном из (желательно объектно-ориентированных) языков программирования

и хотите перейти на C#, эта книга для вас. Вам не придется изучать синтаксис оператора `if`, и я не буду объяснять, что такое объект. Здесь подняты темы, готовящие к более глубокому познанию языка и платформы. Разумеется, я не могу гарантировать, что охватил все необходимое вам перед знакомством с более продвинутым учебником, но, с учетом того, что страниц тут не бесконечное количество, я постарался от души. Очень надеюсь, что книга вам понравится и вы узнаете что-то новое. А если нет, что ж, повторение — мать учения.

Благодарности

Когда я впервые заговорил с сотрудниками издательства об этом проекте, то не представлял, что он отнимет примерно год моей жизни. Честно говоря, меня неоднократно предупреждали, что авторы обычно недооценивают время, требуемое на написание книги. Я, будучи упрямым, думал, что стану исключением. И ошибся. С декабря 2019 по март 2021 года я отдал многие часы работе над текстом. Неоднократно я думал про себя, что «это уж точно конец». И всегда (кроме последнего раза, разумеется) это было не так. К счастью, у меня много свободного времени и очень терпеливая жена.

Я благодарю ее за непоколебимую поддержку в период творчества, без которой мне не удалось бы закончить проект, и извиняюсь за то, что практически исчез из ее жизни на целый год. Также хочу выразить признательность своей семье, с нетерпением ждавшей разговоров о новых возможностях разработки и обновлениях. Выбирая, как назвать генерального директора компании в бизнес-кейсе, рассматриваемом в книге, я позволил себе взять имя у моего деда по материнской линии (Ален), а фамилию — у моей бабушки по отцовской линии (ван дер Меулен).

Мне также следует поблагодарить несравненных работников издательства Mapping, и в особенности Марину Майклс. Редактируя мой текст, она превратила его в нечто большее, чем хаотичный сборник бессвязных разглагольствований. После общения с Мариной у меня появился здоровый страх перед словом «будет» в рукописях. Кроме того, говорю спасибо крайне ценной группе сотрудников в лице Жана-Франсуа Морина, Тани Вильке, Эрика Липперта, Рича Уорда, Энрико Буонанно и Кэти Теннант. Эта межконтинентальная команда супергероев/

ниндзя/рок-звезд предоставила потрясающие отзывы и выявила огромное количество (часто весьма бестолковых) технических ошибок. Я также хочу поблагодарить всех рецензентов и участников MEAP¹, которые прочитали рукопись перед версткой и дали фантастические отзывы, часто в недвусмысленных выражениях. Я не утверждаю, что моя книга — шедевр, но все же надеюсь, что вы извлечете из нее что-то полезное.

Спасибо всем моим рецензентам: Арно Байли, Кристиану Тудалу, Даниэлю Васкес Эступиньяну, Эдину Капичу, Фостеру Хейнсу, Джорджу Томасу, Гетцу Хеллеру, Густаво Филипе Рамос Гомесу, Хильде Ван Гизель, Джареду Дункану, Джейсону Халесу, Жан-Франсуа Морену, Джеффу Нейману, Картикеярджану Раджендрану, Луису Му, Марку Рулло, Марио Солому, Ною Бетцену, Оливеру Кортену, Патрику Ригану, Прабхути Пракашу, Раймонду Чунгу, Резе Зейнали, Ричарду Б. Уорду, Ричарду Хоффу, Сау Фай Фонгу, Славомиру Фурману, Тане Уилки, Томасу Ф. Гету, Виктору М. Пересу и Виктору Беку. Ваши предложения помогли мне улучшить текст.

Наконец, еще несколько человек, которые помогли либо в работе над книгой, либо на протяжении моей карьеры в целом. Прежде всего обращусь к Дэвиду Лавиэлю и Дункану Хендерсону: спасибо, что рискнули и впервые наняли меня для разработки ПО. Джерри Финеган: спасибо, что познакомил меня с C# и позволил задавать друг за дружкой глупые вопросы. Твое терпение, замечания и пожелания крайне ценны. Майкл Бричер: ты приложил руку к формированию некоторых материалов по конгруэнтности в этой книге (под влиянием моих поздних ночных сообщений, в которых я задавал странные математические вопросы об обозначениях), и текст от этого стал еще шикарнее. Шимон Зуберек: первый черновик главы 2 я написал в твоей нью-йоркской квартире. Спасибо, что пускал нас поваляться на диване в любое время суток и, традиционно, за интересные темы для бесед. А еще благодарю замечательных людей из компаний Acronis и Workiva², которым, кажется, целую вечность пришлось слушать, как я бубню о «книге, которую пишу». Вы выдержали это испытание (в целом).

¹ MEAP, Manning Early Access Program (англ. «Программа раннего доступа от Manning») — платная услуга, предоставляемая издательством Manning. Читатели, оформившие подписку, получают возможность как знакомиться с текстом будущей книги еще во время ее написания, так и давать советы и рекомендации автору. — *Здесь и далее прим. лит. ред., если не указано иное.*

² Acronis — компания-разработчик программного обеспечения, зарегистрирована в Швейцарии, основная деятельность ведется в Сингапуре, США и Болгарии. Основана как отдельное предприятие в 2003 году выходцами из бывшего СССР. Workiva — американская компания, работающая в области «программное обеспечение как услуга», основана в Калифорнии в 2008 году.

Об этой книге

Данная работа поможет вам либо плавно повысить уровень программирования, либо перейти на C# с Java или другого объектно-ориентированного языка, опираясь на уже имеющиеся у вас навыки. Вы научитесь писать идиоматический код на C#, необходимый для разработки корпоративных приложений. Здесь рассматриваются основные навыки, требуемые в области бэкенда, и вы сможете применить их на практике при решении задачи, часто встречающейся в карьере любого программиста: рефакторинга унаследованной кодовой базы для обеспечения безопасности, чистоты и читабельности. К концу книги вы будете разбираться в C# на профессиональном уровне и подготовитесь к тому, чтобы начать углубленное изучение с использованием материалов, ориентированных на узкую специализацию.

Здесь нет примеров программ вида Hello, World! или основ программирования с нуля. Вы будете учиться, перерабатывая устаревшую кодовую базу: использовать новые методы, инструменты и лучшие практические приемы, чтобы привести ее в соответствие с современными стандартами C#. Мы возьмем существующую кодовую базу (написанную на .NET Framework) и проведем ее рефакторинг, упростив API, на .NET 5.

Кому стоит прочитать эту книгу

Если вы — разработчик, владеющий одним из объектно-ориентированных языков программирования (ЯП), будь то Java, Dart, C++ или какой-то еще, этот текст поможет вам освоить C# и .NET, причем вам