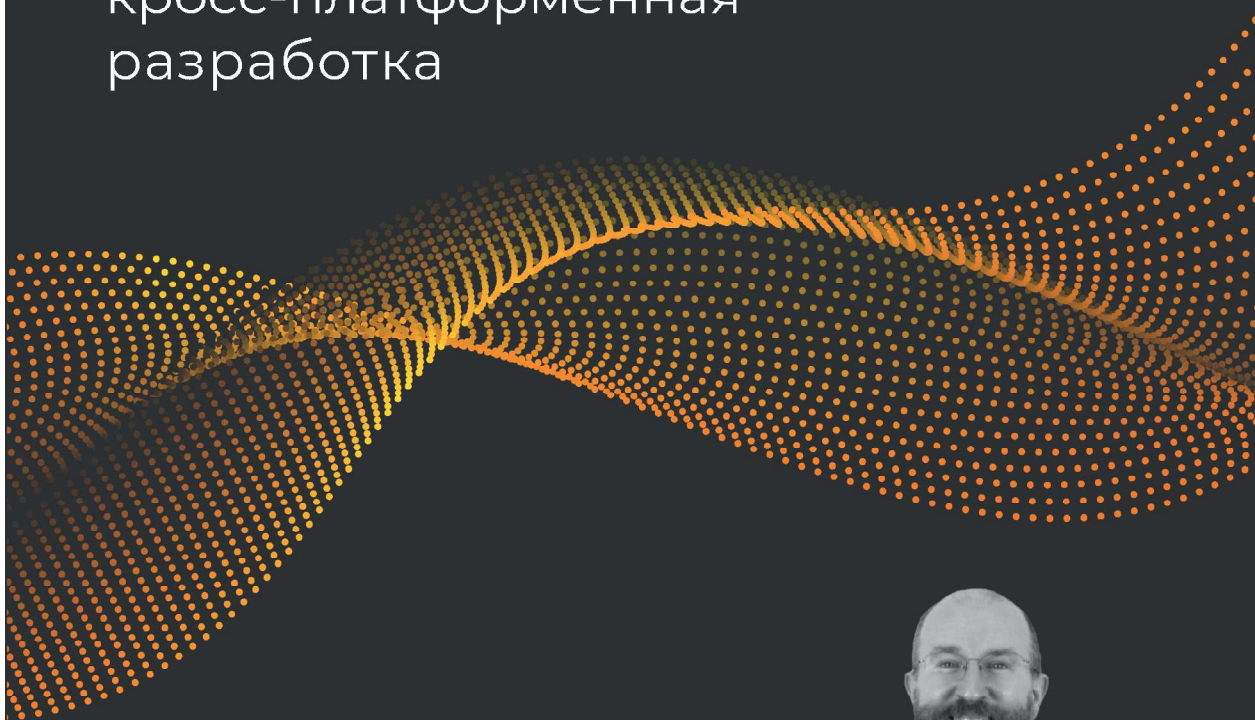




EXPERT INSIGHT

C# 10 и .NET 6

Современная
кросс-платформенная
разработка



Шестое издание



Марк Прайс

Packt

C# 10 and .NET 6 – Modern Cross-Platform Development

Sixth Edition

Build apps, websites, and services with ASP.NET Core 6,
Blazor, and EF Core 6 using Visual Studio 2022 and
Visual Studio Code

Mark J. Price

Packt

BIRMINGHAM - MUMBAI

C# 10 и .NET 6

Шестое издание

Современная
кросс-платформенная
разработка

Марк Прайс



Санкт-Петербург · Москва · Минск

2023

ББК 32.973.2-018.1
УДК 004.43
П68

Прайс Марк

П68 С# 10 и .NET 6. Современная кросс-платформенная разработка. — СПб.: Питер, 2023. — 848 с.: ил. — (Серия «Для профессионалов»).

ISBN 978-5-4461-2249-3

Шестое издание книги серьезно переработано, добавлены все новые функции, реализованные в версиях С# 10 и .NET 6.

Вы изучите принципы объектно-ориентированного программирования, научитесь писать, тестировать и отлаживать функции, реализовывать интерфейсы и наследовать классы. В издании рассматриваются API .NET, призванные решать такие задачи, как управление данными и их запросами, мониторинг и повышение производительности, а также работа с файловой системой, асинхронными потоками, сериализацией и шифрованием. В книге приведены примеры кода кросс-платформенных приложений, веб-сайтов и служб, которые вы можете создавать и развертывать на основе ASP.NET Core.

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

ББК 32.973.2-018.1
УДК 004.43

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 978-1801077361 англ.

© Packt Publishing 2021. First published in the English language under the title 'C# 10 and .NET 6 – Modern Cross-Platform Development - Sixth Edition.

ISBN 978-5-4461-2249-3

© Перевод на русский язык ООО «Прогресс книга», 2022

© Издание на русском языке, оформление ООО «Прогресс книга», 2022

© Серия «Для профессионалов», 2022

Краткое содержание

Об авторе.....	29
О научных редакторах.....	30
Предисловие	31
От издательства.....	37
Глава 1. Привет, C#! Здравствуй, .NET!.....	38
Глава 2. Говорим на языке C#	87
Глава 3. Управление потоком исполнения, преобразование типов и обработка исключений	140
Глава 4. Разработка, отладка и тестирование функций	179
Глава 5. Создание пользовательских типов с помощью объектно-ориентированного программирования.....	227
Глава 6. Реализация интерфейсов и наследование классов	270
Глава 7. Упаковка и распространение типов .NET.....	326
Глава 8. Работа с распространенными типами .NET.....	375
Глава 9. Работа с файлами, потоками и сериализация	429
Глава 10. Работа с данными с помощью Entity Framework Core.....	468
Глава 11. Создание запросов и управление данными с помощью LINQ.....	529
Глава 12. Улучшение производительности и масштабируемости с помощью многозадачности.....	568

Глава 13. Реальные приложения на C# и .NET.....	607
Глава 14. Разработка сайтов с помощью ASP.NET Core Razor Pages	637
Глава 15. Разработка сайтов с помощью паттерна MVC.....	686
Глава 16. Разработка и использование веб-сервисов.....	739
Глава 17. Создание пользовательских интерфейсов с помощью Blazor.....	791
Послесловие.....	843
Дополнительные материалы	845

Оглавление

Об авторе.....	29
О научных редакторах.....	30
Предисловие	31
Примеры исходного кода.....	32
Структура книги	32
Необходимое программное обеспечение.....	35
Скачивание цветных изображений для книги	35
Условные обозначения.....	36
От издательства.....	37
Глава 1. Привет, С#! Здравствуй, .NET!	38
Настройка среды разработки	40
Выбор подходящего инструмента и типа приложения для обучения.....	40
Кросс-платформенное развертывание	43
Скачивание и установка среды Visual Studio 2022 для Windows	44
Скачивание и установка среды Visual Studio Code	45
Знакомство с .NET	48
Обзор .NET Framework.....	48
Проекты Mono, Xamarin и Unity.....	49
Обзор .NET Core	49
Обзор последующих версий .NET	50
Поддержка .NET Core	51
Особенность современной .NET.....	53

Темы современной .NET	54
Обзор .NET Standard	54
Платформы .NET и инструменты, используемые в изданиях этой книги	55
Знакомство с промежуточным языком	56
Сравнение технологий .NET	57
Разработка консольных приложений с использованием Visual Studio 2022	57
Управление несколькими проектами с помощью Visual Studio 2022.....	57
Написание кода с помощью Visual Studio 2022	57
Компиляция и запуск кода с использованием Visual Studio.....	59
Написание программ верхнего уровня.....	61
Добавление второго проекта с помощью Visual Studio 2022.....	62
Создание консольных приложений с помощью Visual Studio Code	64
Управление несколькими проектами с помощью Visual Studio Code.....	64
Написание кода с помощью Visual Studio Code.....	65
Компиляция и запуск кода с помощью инструмента dotnet	67
Добавление второго проекта в программе Visual Studio Code.....	68
Управление несколькими файлами с помощью Visual Studio Code	70
Изучение кода с помощью .NET Interactive Notebooks	70
Создание блокнота	70
Написание и запуск кода в блокноте	72
Сохранение кода в блокноте	73
Добавление в блокнот Markdown и специальных команд	73
Выполнение кода в нескольких ячейках	74
Использование .NET Interactive Notebooks для кода в этой книге.....	75
Просмотр папок и файлов для проектов.....	75
Общие папки и файлы	76
Код решений на GitHub.....	77
Использование репозитория GitHub для этой книги	77
Уточнение вопросов	77
Обратная связь.....	78
Скачивание кода из репозитория GitHub.....	78
Использование системы Git в Visual Studio Code и командной строки.....	79
Поиск справочной информации.....	80
Знакомство с Microsoft Docs.....	80
Получение справки для инструмента dotnet	80
Получение определений типов и их элементов	81
Поиск ответов на Stack Overflow	83

Поисковая система Google.....	84
Подписка на официальный блог .NET	85
Видеоблог Скотта Хансельмана	85
Практические задания	85
Упражнение 1.1. Проверочные вопросы	85
Упражнение 1.2. Практическое задание.....	86
Упражнение 1.3. Дополнительные ресурсы.....	86
Резюме.....	86
Глава 2. Говорим на языке C#	87
Введение в C#	87
Обзор версий языка и их функций	88
Стандарты C#.....	92
Версии компилятора C#.....	93
Основы языка C#: грамматика и терминология	95
Вывод версии компилятора.....	95
Грамматика языка C#	97
Терминология языка C#	99
Сравнение языков программирования с естественными языками	99
Изменение цветовой схемы синтаксиса.....	100
Помощь в написании правильного кода	100
Импорт пространств имен	101
Глаголы = методы.....	105
Существительные = типы данных, поля, переменные и свойства.....	105
Определение объема словаря C#	106
Работа с переменными.....	108
Присвоение имен и значений	109
Литеральные значения.....	109
Хранение текста.....	110
Хранение чисел.....	111
Целые числа	113
Хранение логических значений	117
Хранение объектов любого типа	118
Хранение данных динамического типа	119
Объявление локальных переменных	120
Получение и определение значений по умолчанию для типов	123
Хранение нескольких значений в массиве.....	124

Дальнейшее изучение консольных приложений	125
Отображение вывода пользователю.....	125
Получение пользовательского ввода	129
Упрощение работы с командной строкой.....	129
Получение клавиатурного ввода от пользователя	130
Передача аргументов в консольное приложение.....	131
Настройка параметров с помощью аргументов.....	133
Работа с платформами, не поддерживающими некоторые API	135
Практические задания	137
Упражнение 2.1. Проверочные вопросы	137
Упражнение 2.2. Проверочные вопросы о числовых типах	138
Упражнение 2.3. Практическое задание — числовые размеры и диапазоны.....	138
Упражнение 2.4. Дополнительные ресурсы.....	139
Резюме.....	139

Глава 3. Управление потоком исполнения, преобразование типов и обработка исключений	140
Работа с переменными.....	140
Унарные операции	141
Арифметические бинарные операции	142
Операция присваивания.....	144
Логические операции	144
Условные логические операции.....	145
Побитовые операции и операции побитового сдвига	146
Прочие операции.....	148
Операторы выбора	149
Ветвление с помощью оператора if	149
Сопоставление с образцом с помощью операторов if.....	151
Ветвление с помощью оператора switch	152
Сопоставление с образцом с помощью оператора switch	153
Упрощение операторов switch с помощью выражений switch.....	155
Операторы цикла.....	156
Оператор while.....	156
Оператор do	157
Оператор for.....	158
Оператор foreach	158

Приведение и преобразование типов	159
Явное и неявное приведение типов	160
Преобразование с помощью типа System.Convert	162
Округление чисел	162
Контроль правил округления	163
Преобразование значения любого типа в строку	164
Преобразование двоичного (бинарного) объекта в строку	165
Разбор строк для преобразования в числа или значения даты и времени	166
Обработка исключений	168
Оборачивание потенциально ошибочного кода в оператор try	169
Проверка переполнения	173
Выброс исключений переполнения с помощью оператора checked	173
Отключение проверки переполнения с помощью оператора unchecked	174
Практические задания	176
Упражнение 3.1. Проверочные вопросы	176
Упражнение 3.2. Циклы и переполнение	176
Упражнение 3.3. Циклы и операции	177
Упражнение 3.4. Обработка исключений	177
Упражнение 3.5. Проверка знания операций	178
Упражнение 3.6. Дополнительные ресурсы	178
Резюме	178
Глава 4. Разработка, отладка и тестирование функций	179
Написание функций	179
Пример таблицы умножения	180
Функции, возвращающие значение	182
Преобразование чисел из кардинального в порядковое	184
Вычисление факториалов с помощью рекурсии	185
Документирование функций с помощью XML-комментариев	188
Использование лямбда-выражений в реализациях функций	189
Отладка в процессе разработки	192
Преднамеренное добавление ошибок в код	192
Установка точек останова и начало отладки	193
Навигация с помощью панели средств отладки	197
Панели отладки	197
Пошаговое выполнение кода	198
Настройка точек останова	200

Ведение журнала событий во время разработки и выполнения проекта	202
Ведение журнала событий	202
Работа с типами Debug и Trace	203
Настройка прослушивателей трассировки	205
Переключение уровней трассировки	206
Модульное тестирование	212
Виды тестирования	212
Создание библиотеки классов, требующей тестирования	213
Разработка модульных тестов	214
Генерация и перехват исключений в функциях	217
Ошибки использования и ошибки выполнения	217
Часто выбрасываемые исключения в функциях	217
Стек вызовов	218
Где перехватывать исключения	221
Повторное создание исключений	221
Реализация шаблона tester-doer	223
Практические задания	224
Упражнение 4.1. Проверочные вопросы	224
Упражнение 4.2. Функции, отладка и модульное тестирование	225
Упражнение 4.3. Дополнительные ресурсы	225
Резюме	226

Глава 5. Создание пользовательских типов с помощью объектно-ориентированного программирования	227
Коротко об объектно-ориентированном программировании	227
Разработка библиотек классов	229
Создание библиотек классов	229
Определение классов в пространстве имен	230
Члены	231
Создание экземпляров классов	232
Импорт пространства имен для использования типа	233
Работа с объектами	234
Хранение данных в полях	235
Определение полей	235
Модификаторы доступа	236

Установка и вывод значений полей	236
Хранение значения с помощью типа enum	238
Хранение группы значений с помощью типа enum	239
Хранение нескольких значений с помощью коллекций	240
Коллекции дженериков	241
Создание статического поля	242
Создание константного поля	243
Создание поля только для чтения	244
Инициализация полей с помощью конструкторов	245
Запись и вызов методов	247
Возвращение значений из методов	247
Возвращение нескольких значений с помощью кортежей	248
Определение и передача параметров в методы	251
Перегрузка методов	252
Передача необязательных и именованных параметров	253
Управление передачей параметров	255
Ключевое слово ref	256
Разделение классов с помощью ключевого слова partial	256
Управление доступом с помощью свойств и индексов	257
Определение свойств только для чтения	257
Определение изменяемых свойств	259
Использование модификатора required при определении свойств во время создания экземпляра	260
Определение индексов	261
Сопоставление с образцом с помощью объектов	262
Создание и работа с библиотеками классов .NET 6	262
Определение пассажиров рейса	262
Изменения сопоставления с образцом в C# 9 или более поздних версиях	264
Работа с записями	265
Свойства только для инициализации	265
Записи	266
Позиционные элементы данных в записях	267
Практические задания	268
Упражнение 5.1. Проверочные вопросы	268
Упражнение 5.2. Дополнительные ресурсы	269
Резюме	269

Глава 6. Реализация интерфейсов и наследование классов	270
Настройка библиотеки классов и консольного приложения	271
Дополнительные сведения о методах	272
Реализация функциональности с помощью методов	272
Реализация функциональности с помощью операций	274
Реализация функциональности с помощью локальных функций	275
Подъем и обработка событий	276
Вызов методов с помощью делегатов	277
Определение и обработка делегатов	278
Определение и обработка событий	280
Обеспечение безопасности многократно используемых типов с помощью дженериков	281
Работа с типами, не являющимися дженериками	281
Работа с типами-дженериками	282
Реализация интерфейсов	284
Универсальные интерфейсы	284
Сравнение объектов при сортировке	285
Сравнение объектов с помощью отдельных классов	287
Неявные и явные реализации интерфейса	288
Определение интерфейсов с реализациями по умолчанию	289
Управление памятью с помощью ссылочных типов и типов значений	291
Определение ссылочных типов и типов значений	291
Хранение в памяти ссылочных типов и типов значений	292
Равенство типов	293
Определение типов struct	294
Ключевое слово record и тип struct	295
Освобождение неуправляемых ресурсов	296
Обеспечение вызова метода Dispose	298
Работа со значениями null	298
Создание типа, допускающего значение null	298
Ссылочные типы, допускающие значение null	299
Включение ссылочных типов, допускающих и не допускающих значение null	300
Объявление переменных и параметров, не допускающих значение null	301
Проверка на null	302
Наследование классов	304
Расширение классов	305

Скрытие членов класса.....	305
Переопределение членов.....	306
Наследование от абстрактных классов	307
Предотвращение наследования и переопределения.....	309
Полиморфизм	310
Приведение в иерархиях наследования.....	311
Неявное приведение	311
Явное приведение	311
Обработка исключений приведения.....	312
Наследование и расширение типов .NET	314
Наследование исключений.....	314
Расширение типов при невозможности наследования.....	315
Использование анализатора для написания улучшенного кода	318
Подавление предупреждений.....	320
Практические задания	323
Упражнение 6.1. Проверочные вопросы	324
Упражнение 6.2. Создание иерархии наследования	324
Упражнение 6.3. Дополнительные ресурсы.....	325
Резюме.....	325
Глава 7. Упаковка и распространение типов .NET.....	326
Введение в .NET 6	326
.NET Core 1.0	327
.NET Core 1.1	328
.NET Core 2.0	328
.NET Core 2.1	328
.NET Core 2.2	329
.NET Core 3.0	329
.NET Core 3.1	329
.NET 5.0.....	330
.NET 6.0.....	330
Повышение производительности с .NET Core 2.0 до .NET 5	331
Проверка пакетов SDK для .NET на наличие обновлений	331
Компоненты .NET	331
Сборки, пакеты NuGet и пространства имен	332
Платформа Microsoft .NET и пакет SDK.....	333
Пространства имен и типы в сборках	334

Пакеты NuGet	334
Фреймворки.....	335
Импорт пространства имен в целях использования типа	335
Связь ключевых слов языка C# с типами .NET.....	336
Использование кода с устаревшими платформами, используя .NET Standard	339
Общие сведения о значениях по умолчанию для библиотек классов с различными пакетами SDK.....	340
Создание библиотеки классов .NET Standard 2.0	341
Управление пакетом SDK для .NET.....	341
Публикация и развертывание ваших приложений	343
Разработка консольного приложения для публикации	344
Команды dotnet	345
Получение информации о платформе .NET и ее окружении.....	346
Управление проектами.....	347
Публикация автономного приложения	347
Публикация однофайлового приложения	350
Уменьшение размера приложений с помощью обрезки	351
Декомпиляция сборок .NET	352
Декомпиляция с помощью расширения ILSpy для Visual Studio 2022.....	352
Декомпиляция с помощью расширения ILSpy для Visual Studio Code	353
Нет, вы не можете технически предотвратить декомпиляцию.....	357
Упаковка библиотек для распространения через NuGet.....	359
Ссылка на пакет NuGet.....	359
Упаковка библиотеки для NuGet	360
Изучение пакетов NuGet с помощью инструмента	364
Тестирование пакета библиотеки классов	365
Портирование приложений с .NET Framework на современной .NET.....	366
Можете ли вы портировать	366
Стоит ли портировать	367
Сравнение .NET Framework и современной .NET	368
.NET Portability Analyzer	368
.NET Upgrade Assistant	368
Использование библиотек, не скомпилированных для .NET Standard.....	369
Функции предварительного просмотра	371
Требование к функциям предварительного просмотра.....	371

Включение функций предварительного просмотра	372
Математические операции с дженериками	373
Практические задания	373
Упражнение 7.1. Проверочные вопросы	373
Упражнение 7.2. Дополнительные ресурсы	374
Упражнение 7.3. PowerShell	374
Резюме	374
Глава 8. Работа с распространенными типами .NET	375
Работа с числами	375
Большие целые числа	376
Работа с комплексными числами	377
Кватернионы	378
Работа с текстом	378
Извлечение длины строки	378
Извлечение символов строки	379
Разделение строк	379
Извлечение фрагмента строки	380
Проверка содержимого в строках	381
Конкатенация строк, форматирование и прочие члены типа string	382
Эффективное создание строк	383
Работа с датами и временем	383
Указание значений даты и времени	384
Глобализация с учетом дат и времени	386
Обработка дат/времени по отдельности	388
Сопоставление с образцом при помощи регулярных выражений	389
Проверка цифр, введенных в виде текста	389
Рост производительности регулярных выражений	391
Синтаксис регулярных выражений	391
Примеры регулярных выражений	392
Разбивка сложных строк, разделенных запятыми	393
Хранение нескольких объектов в коллекциях	394
Общие свойства коллекций	395
Повышение производительности за счет обеспечения пропускной способности коллекции	396
Выбор коллекции	396

Работа со списками	400
Работа со словарями	402
Работа с очередями.....	404
Сортировка коллекций	406
Более специализированные коллекции	407
Использование неизменяемых коллекций	407
Эффективные приемы работы с коллекциями.....	408
Работа с интервалами, индексами и диапазонами.....	409
Управление памятью с помощью интервалов.....	409
Идентификация позиций с помощью типа Index.....	410
Идентификация диапазонов с помощью типа Range.....	410
Использование индексов, диапазонов и интервалов	411
Работа с сетевыми ресурсами.....	412
Работа с URI, DNS и IP-адресами.....	412
Проверка соединения с сервером	414
Работа с отражением и атрибутами	415
Версии сборок	416
Чтение метаданных сборки	416
Создание пользовательских атрибутов.....	419
Возможности отражения.....	421
Работа с изображениями.....	421
Интернационализация кода.....	423
Обнаружение и изменение региональных настроек.....	424
Практические задания	426
Упражнение 8.1. Проверочные вопросы	426
Упражнение 8.2. Регулярные выражения.....	427
Упражнение 8.3. Методы расширения.....	427
Упражнение 8.4. Дополнительные ресурсы.....	428
Резюме.....	428
Глава 9. Работа с файлами, потоками и сериализация	429
Управление файловой системой	429
Работа с кросс-платформенными средами и файловыми системами	429
Управление дисками	431
Управление каталогами	432
Управление файлами	434
Управление путями	435

Извлечение информации о файле.....	436
Контроль работы с файлами	437
Чтение и запись с помощью потоков.....	438
Абстрактные и конкретные потоки	438
Запись в текстовые потоки	440
Запись в XML-потоки	442
Освобождение файловых ресурсов	443
Сжатие потоков	446
Сжатие с помощью алгоритма Brotli	448
Кодирование и декодирование текста.....	450
Кодировка строк в последовательности байтов	451
Кодирование и декодирование текста в файлах	454
Сериализация графов объектов	454
XML-сериализация	455
Генерация компактного XML	458
XML-десериализация.....	458
JSON-сериализация	459
Высокопроизводительная обработка JSON	460
Управление обработкой JSON	462
Новые методы расширения JSON для работы с ответами HTTP.....	465
Переход с Newtonsoft на новый JSON	465
Практические задания	465
Упражнение 9.1. Проверочные вопросы	465
Упражнение 9.2. XML-сериализация.....	466
Упражнение 9.3. Дополнительные ресурсы.....	467
Резюме.....	467
Глава 10. Работа с данными с помощью Entity Framework Core	468
Современные базы данных.....	468
Устаревшая Entity Framework	468
Entity Framework Core	469
Создание консольного приложения для работы с EF Core	470
Использование образца реляционной базы данных.....	470
Использование Microsoft SQL Server в Windows.....	472
Создание образца базы данных Northwind для SQL Server.....	473
Управление образцом базы данных Northwind с помощью Server Explorer	474
Использование SQLite.....	475

Создание образца базы данных Northwind для SQLite.....	476
Управление образцом базы данных Northwind в SQLiteStudio	477
Настройка EF Core	479
Выбор поставщика данных Entity Framework Core	479
Подключение к базе данных	480
Определение контекста базы данных Northwind	481
Определение моделей EF Core	483
Соглашения Entity Framework Core для определения модели	483
Использование атрибутов аннотаций Entity Framework Core для определения модели.....	484
Использование Entity Framework Core Fluent API для определения модели.....	485
Создание модели Entity Framework Core для таблиц Northwind.....	486
Добавление таблиц в контекстный класс базы данных Northwind.....	489
Настройка инструмента dotnet-ef	490
Создание шаблонов с использованием существующей базы данных	491
Настройка предварительных моделей.....	495
Запрос данных из моделей EF Core	496
Фильтрация включенных сущностей	498
Фильтрация и сортировка товаров.....	500
Получение сгенерированного SQL-кода	501
Логирование EF Core с помощью провайдера логов для пользователя.....	502
Сопоставление с образцом с помощью оператора Like	507
Определение глобальных фильтров.....	508
Схемы загрузки шаблонов с помощью EF Core	509
Жадная загрузка сущностей	509
Включение ленивой загрузки	510
Явная загрузка сущностей	511
Управление данными с помощью EF Core	514
Добавление сущностей.....	514
Обновление сущностей.....	515
Удаление сущностей	516
Объединение контекстов базы данных	517
Работа с транзакциями	518
Управление транзакциями с помощью уровней изоляции	518
Определение явной транзакции	519

Модели EF Core под названием Code First	520
Миграции	526
Практические задания	527
Упражнение 10.1. Проверочные вопросы	527
Упражнение 10.2. Экспорт данных с помощью различных форматов сериализации	527
Упражнение 10.3. Дополнительные ресурсы	528
Упражнение 10.4. Изучение базы данных NoSQL	528
Резюме	528
Глава 11. Создание запросов и управление данными с помощью LINQ	529
Написание выражений LINQ	529
Как работает LINQ	529
Создание выражений LINQ с помощью класса Enumerable	530
Фильтрация элементов с помощью метода Where	534
Ссылка на именованные методы	536
Упрощение кода за счет удаления явного создания экземпляра делегата	537
Использование лямбда-выражений	537
Сортировка элементов	538
Объявление запроса с помощью ключевого слова var или заданного типа	539
Фильтрация по типу	540
Работа с множествами с помощью LINQ	541
Использование LINQ с EF Core	543
Создание модели EF Core	544
Фильтрация и сортировка последовательностей	547
Проецирование последовательностей в новые типы	548
Объединение и группировка последовательностей	550
Агрегирование последовательностей	554
Подслащение синтаксиса LINQ с помощью синтаксического сахара	555
Использование нескольких потоков и параллельного LINQ	556
Разработка приложения с помощью нескольких потоков	556
Создание собственных методов расширения LINQ	559
Использование цепного метода расширения	562
Использование методов Mode и Median	562
Работа с LINQ to XML	563
Генерация XML с помощью LINQ to XML	563
Чтение XML с помощью LINQ to XML	564

Практические задания	565
Упражнение 11.1. Проверочные вопросы	566
Упражнение 11.2. Создание запросов LINQ.....	566
Упражнение 11.3. Дополнительные ресурсы	567
Резюме	567
Глава 12. Улучшение производительности и масштабируемости с помощью многозадачности.....	568
Процессы, потоки и задачи	568
Мониторинг производительности и использования ресурсов.....	569
Оценка эффективности типов.....	570
Мониторинг производительности и памяти с помощью диагностики	571
Измерение эффективности обработки строк.....	574
Мониторинг производительности и памяти с помощью Benchmark.NET.....	576
Асинхронное выполнение задач.....	580
Синхронное выполнение нескольких действий.....	580
Асинхронное выполнение нескольких действий с помощью задач	582
Ожидание выполнения задач	583
Задачи продолжения	584
Вложенные и дочерние задачи	586
Обертывание задач вокруг других объектов	587
Синхронизация доступа к общим ресурсам.....	589
Доступ к ресурсу из нескольких потоков	590
Применение взаимоисключающей блокировки к «раковине»	591
Синхронизация событий.....	594
Выполнение атомарных операций CPU	595
Использование других типов синхронизации.....	596
Ключевые слова <code>async</code> и <code>await</code>	597
Увеличение скорости отклика консольных приложений.....	597
Увеличение скорости отклика GUI-приложений	598
Улучшение масштабируемости клиент-серверных приложений	603
Общие типы, поддерживающие многозадачность	603
Ключевое слово <code>await</code> в блоках <code>catch</code>	604
Работа с асинхронными потоками	604

Практические задания	605
Упражнение 12.1. Проверочные вопросы	605
Упражнение 12.2. Дополнительные ресурсы	606
Резюме	606
Глава 13. Реальные приложения на C# и .NET	607
Модели приложений для C# и .NET	607
Разработка сайтов с помощью ASP.NET Core	608
Создание веб- и других сервисов	610
Создание мобильных и настольных приложений	611
Альтернативы .NET MAUI	612
Нововведения ASP.NET Core	613
ASP.NET Core 1.0	613
ASP.NET Core 1.1	613
ASP.NET Core 2.0	614
ASP.NET Core 2.1	614
ASP.NET Core 2.2	615
ASP.NET Core 3.0	615
ASP.NET Core 3.1	616
Blazor WebAssembly 3.2	616
ASP.NET Core 5.0	616
ASP.NET Core 6.0	616
Создание настольных приложений только для Windows	617
Устаревшие платформы приложений Windows	617
Современная поддержка .NET для устаревших платформ Windows	619
Структурирование проектов	619
Структурирование проектов в решении или рабочей области	619
Использование других шаблонов проектов	621
Установка дополнительных пакетов шаблонов	622
Разработка сущностной модели данных для базы данных Northwind	622
Разработка библиотеки классов для сущностных моделей с помощью SQLite	623
Создание библиотеки классов для сущностных моделей с помощью SQL Server	632
Практические задания	635
Упражнение 13.1. Проверочные вопросы	635
Упражнение 13.2. Дополнительные ресурсы	635
Резюме	635

Глава 14. Разработка сайтов с помощью ASP.NET Core Razor Pages	637
Веб-разработка	637
Протокол передачи гипертекста	637
Использование браузера Google Chrome для выполнения HTTP-запросов	639
Технологии клиентской веб-разработки.....	642
Обзор ASP.NET Core	643
Классический ASP.NET против современного ASP.NET Core	644
Создание пустого проекта ASP.NET Core	645
Тестирование и защита сайта.....	647
Управление средой хостинга.....	651
Разделение конфигурации для сервисов и конвейера	653
Как позволить сайту обрабатывать статический контент	655
Функция Razor Pages от ASP.NET Core	658
Добавление Razor Pages	658
Добавление кода к Razor Pages.....	658
Использование общих макетов в Razor Pages	660
Использование файлов с выделенным кодом в Razor Pages.....	662
Использование Entity Framework Core совместно с ASP.NET Core	665
Настройка Entity Framework Core как сервиса.....	665
Управление данными с помощью страниц Razor.....	667
Внедрение сервиса зависимостей в страницу Razor	669
Использование библиотек классов Razor	670
Создание библиотеки классов Razor.....	670
Отключение компактного режима просмотра папок в Visual Studio Code	671
Реализация функции сотрудников с помощью EF Core.....	672
Реализация частичного представления для отображения одного сотрудника	674
Использование и тестирование библиотеки классов Razor.....	675
Настройка сервисов и конвейера HTTP-запросов	676
Маршрутизация конечных точек	676
Проверка конфигурации маршрутизации конечных точек	677
Обобщение ключевых методов расширения промежуточного программного обеспечения	681
Визуализация HTTP-конвейера.....	682
Реализация анонимного встроенного делегата в качестве промежуточного программного обеспечения	682

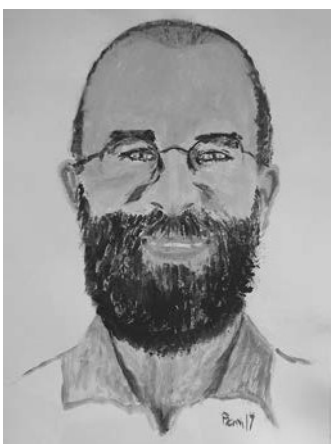
Практические задания	684
Упражнение 14.1. Проверочные вопросы	684
Упражнение 14.2. Веб-страница, управляемая данными	685
Упражнение 14.3. Создание веб-страниц для консольных приложений	685
Упражнение 14.4. Дополнительные ресурсы	685
Резюме	685
Глава 15. Разработка сайтов с помощью паттерна MVC.....	686
Настройка сайта ASP.NET Core MVC.....	686
Создание сайтов ASP.NET Core MVC	687
Создание базы данных аутентификации для SQL Server LocalDB	688
Сайт ASP.NET Core MVC по умолчанию	689
Структура проекта сайта MVC.....	691
Обзор базы данных ASP.NET Core Identity.....	693
Работа сайта ASP.NET Core MVC.....	694
Инициализация ASP.NET Core MVC	694
Маршрутизация MVC по умолчанию.....	697
Контроллеры и действия.....	698
Соглашение о пути поиска представлений.....	701
Ведение журнала	702
Фильтры.....	703
Сущности и модели представлений	710
Представления	713
Добавление собственного функционала на сайт ASP.NET Core MVC	715
Определение пользовательских стилей.....	715
Настройка категории изображений.....	716
Синтаксис Razor	716
Определение типизированного представления	717
Проверка измененной главной страницы	720
Передача параметров с помощью значения маршрута	721
Тонкости привязки моделей	723
Проверка модели.....	727
Методы вспомогательного класса для представления	730
Отправка запросов в базу данных и использование шаблонов отображения	732
Улучшение масштабируемости с помощью асинхронных задач	735
Преобразование методов действия контроллера в асинхронные.....	735

Практические задания	736
Упражнение 15.1. Проверочные вопросы	736
Упражнение 15.2. Реализация MVC для страницы, содержащей сведения о категориях	737
Упражнение 15.3. Улучшение масштабируемости за счет понимания и реализации асинхронных методов действий	737
Упражнение 15.4. Практика модульного тестирования контроллеров MVC	738
Упражнение 15.5. Дополнительные ресурсы	738
Резюме	738
Глава 16. Разработка и использование веб-сервисов	739
Разработка веб-сервисов с помощью Web API в ASP.NET Core	739
Аббревиатуры, типичные для веб-сервисов	739
HTTP-запросы и ответы для Web API	741
Разработка проекта Web API в ASP.NET Core	743
Функциональность веб-сервисов	746
Создание веб-сервиса для базы данных Northwind	748
Создание репозитория данных для сущностей	750
Реализация контроллера Web API	753
Настройка репозитория клиента и контроллера Web API	755
Указание сведений о проблеме	759
Управление сериализацией XML	760
Документирование и тестирование веб-сервисов	761
Тестирование GET-запросов в браузерах	761
Тестирование HTTP-запросов с помощью расширения REST Client	763
Swagger	766
Тестирование запросов с помощью Swagger UI	768
Протоколирование HTTP	773
Обращение к веб-сервисам с помощью HTTP-клиентов	774
Класс HttpClient	774
Настройка HTTP-клиентов с помощью HttpClientFactory	775
Получение контроллером списка клиентов в формате JSON	776
Совместное использование ресурсов между разными источниками	778
Реализация расширенных функций веб-сервисов	780
Реализация проверки работоспособности API	780
Реализация анализаторов и соглашений Open API	781

Обработка проходных отказов.....	782
Добавление HTTP-заголовков в целях безопасности.....	782
Создание веб-сервисов с помощью минимальных API.....	784
Создание сервиса погоды с помощью минимальных API.....	785
Тестирование минимального сервиса погоды.....	786
Добавление прогнозов погоды на главную страницу сайта Northwind.....	787
Практические задания.....	789
Упражнение 16.1. Проверочные вопросы.....	789
Упражнение 16.2. Создание и удаление клиентов с помощью класса HttpClient.....	790
Упражнение 16.3. Дополнительные ресурсы.....	790
Резюме.....	790
Глава 17. Создание пользовательских интерфейсов с помощью Blazor.....	791
Знакомство с Blazor.....	791
Недостатки JavaScript.....	792
Silverlight: C# и .NET на основе плагина.....	792
WebAssembly: цель для Blazor.....	792
Модели хостинга Blazor.....	793
Компоненты Blazor.....	794
Различия между Blazor и Razor.....	795
Сравнение шаблонов проектов Blazor.....	795
Обзор шаблона проекта Blazor Server.....	795
Маршрутизация Blazor по компонентам страницы.....	802
Запуск шаблона проекта Blazor Server.....	805
Обзор шаблона проекта Blazor WebAssembly.....	807
Сборка компонентов с помощью Blazor Server.....	811
Определение и тестирование простого компонента.....	811
Создание маршрутизируемого компонента страницы.....	812
Получение сущностей в компоненте.....	813
Абстрагирование сервиса для компонента Blazor.....	816
Определение форм с помощью компонента EditForm.....	818
Создание и использование компонента формы клиента.....	819
Тестирование компонента формы клиента.....	823
Создание компонентов с помощью Blazor WebAssembly.....	824
Настройки сервера для Blazor WebAssembly.....	824
Настройка клиента для Blazor WebAssembly.....	827
Тестирование компонентов и сервиса Blazor WebAssembly.....	830

Улучшение приложений Blazor WebAssembly.....	832
Включение Blazor WebAssembly AOT.....	832
Поддержка прогрессивных веб-приложений.....	834
Анализатор совместимости браузеров для Blazor WebAssembly	835
Совместное использование компонентов Blazor в библиотеке классов.....	836
Взаимодействие с JavaScript	838
Библиотеки компонентов Blazor	841
Практические задания	841
Упражнение 17.1. Проверочные вопросы	841
Упражнение 17.2. Упражнения по созданию компонента таблицы умножения.....	842
Упражнение 17.3. Упражнения по созданию элемента навигации по стране.....	842
Упражнение 17.4. Дополнительные ресурсы	842
Резюме	842
Послесловие.....	843
Дальнейшее изучение C# и .NET	843
Следующее издание.....	844
Дополнительные материалы.....	845

Об авторе



Марк Дж. Прайс — обладатель сертификатов Microsoft Specialist: Programming in C# и Microsoft Specialist: Architecting Microsoft Azure Solutions; опыт работы — более 20 лет.

С 1993 года Марк сдал более 80 экзаменов компании Microsoft по программированию и специализируется на подготовке других людей к успешному прохождению тестирования. В период с 2001 по 2003 год посвящал все свое время разработке официального обучающего программного обеспечения в штаб-квартире Microsoft (Редмонд, США). В составе команды написал первый обучающий курс по C#, когда существовала лишь ранняя альфа-версия языка. Во время сотрудничества с Microsoft работал инструктором по повышению квалификации сертифицированных компанией специалистов на тренингах, посвященных C# и .NET. В настоящее время разрабатывает и поддерживает обучающие курсы для системы Digital Experience Platform (DXP). Кроме того, получил степень бакалавра компьютерных наук с отличием в Бристольском университете (Великобритания).

Microsoft
CERTIFIED
Solutions Developer

App Builder

Microsoft
Specialist

Programming in C#

О научных редакторах

Дамир Арх — профессионал с многолетним опытом разработки и сопровождения программного обеспечения (ПО): от сложных корпоративных программных проектов до современных потребительских мобильных приложений. Работал со многими языками, однако любимым остается C#. Стремясь к совершенствованию процессов разработки, Дамир является сторонником разработки через тестирование, непрерывной интеграции и непрерывного развертывания. Он делится своими знаниями, выступая в группах пользователей и на конференциях, ведет блоги и пишет статьи. Десять раз подряд получил престижную премию Microsoft MVP в области технологий разработки. В свободное время всегда в движении: любит пеший туризм, геокэшинг, бег и скалолазание.

Джованни Альсате Сандовал — системный инженер из города Медельин (Колумбия). Ему нравится все, что связано с разработкой программного обеспечения, новыми технологиями, паттернами проектирования и архитектурой ПО. Более 14 лет трудится разработчиком, техническим руководителем и архитектором ПО, в основном с технологиями компании Microsoft. Ему нравится разрабатывать системы OSS, также он внес свой вклад в Asp.NET Core SignalR, Polly, Apollo Server и другие системы. Принял участие в создании OSS-библиотеки Simmy, библиотеки OSS (chaos engineering) для .NET, основанной на Polly. Кроме того, является приверженцем DDD и горячим поклонником облачных технологий. Джованни — участник .NET Foundation, соорганизатор сообществ MDE.NET и .NET developers в Медельине, Колумбия. В последние годы специализируется на разработке надежных распределенных систем на основе облачных технологий. И последнее, но не менее важное: он твердо верит в командную работу. Его слова: «Меня бы здесь не было, если бы я не научился многому у всех моих талантливых коллег».

Сейчас Джованни трудится техническим директором в калифорнийском стартапе Curbit.

Предисловие

В книжном магазине вы увидите объемные книги по программированию, цель которых — предоставить полный материал по языку C#, библиотекам .NET, моделям приложений, таким как сайты, веб-сервисы, настольные и мобильные приложения.

Эта книга другая. Материал излагается кратко и увлекательно и содержит практические руководства по каждой теме. Широта повествования достигается за счет несколько меньшей глубины, но при желании вы найдете здесь множество ссылок, которые позволяют продолжить погружение в тему.

Издание одновременно представляет собой пошаговое руководство по изучению современных проверенных практик на языке C# с использованием кросс-платформенной .NET и краткое введение в основные типы приложений, которые можно создавать с их помощью. Книга лучше всего подходит новичкам в C# и .NET или программистам, которые работали с C# раньше и хотят усовершенствовать свои навыки.

Если у вас уже есть опыт работы со старыми версиями языка C#, то в первом разделе главы 2, касающейся C#, вы можете ознакомиться со сравнительными таблицами новых языковых функций и перейти к ним.

Если вы уже имеете опыт работы со старыми версиями .NET, то в первом разделе главы 7 можете ознакомиться с таблицами новых функций и сразу перейти к ним.

Я расскажу об интересных возможностях и подводных камнях C# и .NET, чтобы вы могли впечатлить коллег и потенциальных работодателей и быстро повысить свою продуктивность. Я мог бы нудно разьяснять каждую деталь и тем самым утомлять некоторых читателей. Но все же я надеюсь, что вы умеете пользоваться поисковыми системами типа Google и найдете в Интернете подробную информацию о темах, которые я посчитал необязательным включать в гайд для пользователей начального и среднего уровня ввиду ограничений, свойственных печатной книге.

Примеры исходного кода

Файлы примеров для выполнения упражнений из данной книги вы можете бесплатно скачать со страницы репозитория GitHub (<https://github.com/markjprice/cs10dotnet6>). Инструкции о том, как это сделать, я предоставлю в конце главы 1.

Структура книги

Глава 1 «Привет, C#! Здравствуй, .NET!» посвящена тому, как настроить среду разработки и с помощью Visual Studio или Visual Studio Code создавать простейшее приложение, используя язык C# и .NET. Разработка упрощенных консольных приложений предполагает знание того, как использовать программную функцию верхнего уровня, написанную на C# 9. Простые языковые конструкции и возможности библиотеки вы можете изучить в .NET Interactive Notebooks. Кроме того, вы сможете узнать, где найти справочную информацию и как связаться со мной через хранилище GitHub, чтобы получить помощь в решении проблемы или дать обратную связь по книге, которая позволит усовершенствовать ее будущие переиздания.

Глава 2 «Говорим на языке C#» знакомит нас с версиями языка C# и приводит таблицы, помогающие понять, в какой версии были представлены те или иные функциональные особенности языка. Вдобавок в этой главе вы познакомитесь с грамматикой и лексикой C#, которыми будете пользоваться каждый день, создавая исходный код своих приложений. В частности, вы узнаете, как объявлять переменные разных типов и работать с ними.

Глава 3 «Управление потоком исполнения, преобразование типов и обработка исключений» описывает использование операций для выполнения простых действий с переменными, включая сравнения, написание кода, который принимает решения, сопоставление с образцом в C# 7 — C# 10, повторение блока операторов и выполнение преобразования между типами. Кроме того, в главе рассказывается о том, как писать код для обработки исключений при их неизбежном возникновении.

Глава 4 «Разработка, отладка и тестирование функций» посвящена соблюдению принципа программирования *DRY* (Do not Repeat Yourself — «Не повторяйся») при создании многократно используемых функций. Вы узнаете, как с помощью инструментов отладки отслеживать и устранять ошибки, мониторить код в процессе его выполнения для диагностики проблем. Кроме того, я расскажу, как тщательно тестировать код, чтобы устранять ошибки и обеспечивать его стабильность и надежность до развертывания его в производственной среде.

Глава 5 «Создание пользовательских типов с помощью объектно-ориентированного программирования» знакомит с различными категориями элементов, которые может иметь тип, в том числе с полями для хранения данных и методами для выпол-

нения действий. Вы будете использовать концепции *объектно-ориентированного программирования (ООП)*, такие как агрегирование и инкапсуляция. Вы изучите языковые функции, такие как поддержка синтаксиса кортежей и переменные `out`, литералы для значений по умолчанию и автоматически определяемые имена элементов кортежей. Кроме того, вы научитесь определять и работать с неизменяемыми типами с помощью ключевого слова `record`, свойств только для инициализации и выражений, представленных в C# 9.

Глава 6 «Реализация интерфейсов и наследование классов» описывает создание новых типов из существующих с использованием объектно-ориентированного программирования. Вы узнаете, как определять операции и локальные функции, делегаты и события, как реализовывать интерфейсы с базовыми и производными классами, переопределять элементы типа. Кроме того, вы изучите концепции полиморфизма, научитесь создавать методы расширения и поймете, как выполнять приведение классов в иерархии наследования. Вы также узнаете о значительных изменениях в C# 8, вызванных введением ссылочных типов, допускающих значение `null` (nullable).

Глава 7 «Упаковка и распространение типов .NET» представляет версии .NET и содержит таблицы, в которых приведены новые функции, а также типы .NET, соответствующие стандарту .NET, и их отношение к языку C#. Вы научитесь создавать и компилировать код на любой из поддерживаемых операционных систем: Windows, macOS и Linux. Вы также узнаете, как разворачивать и упаковывать собственные приложения и библиотеки.

Глава 8 «Работа с распространенными типами .NET» описывает типы, позволяющие вашему коду выполнять типовые практические задачи, такие как управление числами и текстом, датой и временем, хранение элементов в коллекциях, работа с сетью и изображениями и реализация интернационализации.

Глава 9 «Работа с файлами, потоками и сериализация» касается взаимодействия с файловой системой, чтения и записи в файлы и потоки, кодирования текста и форматов сериализации, таких как JSON и XML, включая улучшенную функциональность и производительность классов `System.Text.Json`.

Глава 10 «Работа с данными с помощью Entity Framework Core» научит читать данные и записывать их в базы данных, такие как Microsoft SQL Server и SQLite, с помощью технологии *объектно-реляционного отображения данных* Entity Framework Core. Вы узнаете, как определить модели сущностей, которые можно сопоставлять с существующими таблицами в базе данных. Мы также расскажем, как определить модели Code First, которые могут создавать таблицы и базы данных во время выполнения программы.

Глава 11 «Создание запросов и управление данными с помощью LINQ» рассказывает о языковых расширениях *Language INtegrated Queries (LINQ)*, которые позволяют

работать с последовательностями элементов с их последующей фильтрацией, сортировкой и проецированием на различные программные выходы. Вы также узнаете о специальных возможностях `Parallel LINQ (PLINQ)` и `LINQ` для XML.

Глава 12 «Улучшение производительности и масштабируемости с помощью многозадачности» рассказывает, как одновременно выполнять несколько действий, чтобы повысить производительность, масштабируемость и продуктивность пользователей. Вы узнаете о функции `async Main` и о том, как с помощью типов в пространстве имен `System.Diagnostics` мониторить код с целью измерения производительности и эффективности.

Глава 13 «Реальные приложения на C# и .NET» знакомит с типами кросс-платформенных приложений, которые можно создавать с помощью C# и .NET. Вы также создадите модель EF Core для представления базы данных Northwind, с которой будете работать в последующих главах этой книги.

Глава 14 «Разработка сайтов с помощью ASP.NET Core Razor Pages» посвящена тому, как создавать сайты с современной HTTP-архитектурой на стороне сервера с помощью ASP.NET Core. Вы научитесь проектировать простые сайты, используя функцию ASP.NET Core, известную как Razor Pages, которая упрощает создание динамических веб-страниц для небольших сайтов. Вы также научитесь создавать запросы и получать ответы HTTP.

Глава 15 «Разработка сайтов с помощью паттерна MVC» посвящена созданию сложных сайтов таким образом, чтобы можно было легко их тестировать и управлять ими с помощью команд, использующих ASP.NET Core MVC. Кроме того, здесь описаны такие темы, как конфигурации запуска, аутентификация, маршруты, модели, представления и контроллеры.

Глава 16 «Разработка и использование веб-сервисов» описывает, как создавать поддерживающие REST веб-сервисы с помощью ASP.NET Core Web API и как правильно потреблять их, задействуя созданные фабрикой HTTP-клиенты.

Глава 17 «Разработка пользовательских веб-интерфейсов с помощью Blazor» посвящена созданию компонентов пользовательского веб-интерфейса с помощью Blazor, которые могут выполняться либо на стороне сервера, либо внутри браузера на стороне клиента. Вы проанализируете различия между Blazor Server и Blazor WebAssembly и узнаете, как создавать компоненты, упрощающие переключение между двумя моделями хостинга.

Эту книгу дополняют три главы, которые вы можете скачать по ссылке <https://clck.ru/32EVw7>.

Глава 18 «Создание и использование специализированных сервисов» рассказывает о создании сервисов с помощью gRPC, а также о том, как реализовывать взаимодей-

ствие между сервером и клиентом в режиме реального времени, используя SignalR. Кроме того, вы научитесь предоставлять данные в качестве веб-сервиса с помощью OData и размещать в облаке функции, реагирующие на триггеры, используя платформу Azure Functions.

Глава 19 «Разработка мобильных и настольных приложений с помощью .NET MAUI» знакомит вас с созданием кросс-платформенных мобильных и настольных приложений для Android, iOS, macOS и Windows. Вы изучите основы XAML, с помощью которых можно определять пользовательский интерфейс для графического приложения.

Глава 20 «Защита данных и приложений» посвящена тому, как с помощью шифрования защищать данные от просмотра злоумышленниками, а также тому, как применять хеширование и цифровые подписи для защиты данных от вмешательства или повреждения. Кроме того, вы узнаете, как с помощью аутентификации и авторизации защищать приложения от неавторизованных пользователей.

Приложение содержит ответы на проверочные вопросы, приведенные в конце каждой главы.

Необходимое программное обеспечение

Вы можете разрабатывать и разворачивать приложения C# и .NET с помощью Visual Studio Code на многих платформах, включая Windows, macOS и большинство модификаций Linux.

Вам необходимы операционная система, поддерживающая Visual Studio Code, и подключение к Интернету (для всех глав, кроме одной).

Вы также можете использовать Visual Studio для Windows или macOS либо сторонний инструмент, например JetBrains Rider.

Понадобится операционная система macOS для разработки приложений, описанных в главе 19, поскольку для компиляции приложений iOS вам потребуются macOS и Xcode.

Скачивание цветных изображений для книги

Вы также можете просмотреть и скачать сверстаный PDF-файл с цветными версиями оригинальных иллюстраций. Полноцветные изображения помогут быстрее разобраться в примерах. Файл доступен по адресу https://static.packt-cdn.com/downloads/9781801077361_ColorImages.pdf.

Условные обозначения

В книге вы увидите текст, оформленный различными стилями. Ниже я привел примеры и объяснил, что означает это форматирование.

Фрагменты кода в тексте, имена таблиц баз данных, файлов, расширения файлов, пути, ввод пользователя, а также учетные записи в Twitter оформляются следующим образом: «Папки **Controllers**, **Models** и **Views** содержат классы ASP.NET Core и файлы `.cshtml` для выполнения на сервере».

Блок кода выглядит так:

```
// хранение элементов в индексированных позициях
names[0] = "Kate";
names[1] = "Jack";
names[2] = "Rebecca";
names[3] = "Tom";
```

Если нужно обратить ваше внимание на определенную часть приведенного кода, то соответствующие строки или элементы будут выделены **полужирным моноширинным** шрифтом:

```
// хранение элементов в индексированных позициях
names[0] = "Kate";
names[1] = "Jack";
names[2] = "Rebecca";
names[3] = "Tom";
```

Весь ввод/вывод в командной строке дается так:

```
dotnet new console
```

URL и имена папок оформляются **шрифтом без засечек**. Слова, которые вы видите на экране, например в меню или диалоговых окнах, отображаются в тексте так: «Нажмите кнопку **Next** (Далее), чтобы перейти к следующему экрану».

Новые термины и важные слова выделены *курсивом*.



Этот рисунок указывает на важные примечания и ссылки на внешние дополнительные источники информации.



Этот рисунок указывает на советы и рекомендации экспертов по разработке.

От издательства

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

Мы будем рады узнать ваше мнение!

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

Рекомендуем посмотреть комментарии автора по адресу <https://github.com/markjprice/cs10dotnet6/blob/main/errata.md>: он поясняет некоторые примеры.