

*Марлен, продолжающей вдохновлять меня и поддерживать все,
что мы делаем вместе.*

Введение

Сообщество C# в 2016 году сильно отличается от того, что было в 2004 году, когда вышло первое издание книги *Effective C#*. Разработчиков, использующих C#, стало намного больше. Крупная доля участников сообщества C# теперь считает C# своим главным профессиональным языком. Они не переходили на C# с набором укоренившихся привычек, которые сформировались в результате работы с каким-то другим языком. Сообщество обладает гораздо более широким опытом. Язык C# применяют как выпускники учебных заведений, так и профессионалы с десятками лет стажа. Теперь язык C# доступен для множества платформ. Вы можете строить на C# серверные приложения, веб-сайты, настольные приложения и мобильные приложения для многочисленных платформ.

Я организовал третье издание книги с учетом изменений в языке, а также изменений в сообществе C#. Эта книга не является историческим экскурсом в изменения языка. Взамен приводятся рекомендации о том, как использовать текущую версию языка C#. Советы, которые были изъяты из настоящего издания, не имеют отношения к современному языку C# или к теперешним приложениям. Новые советы охватывают новые возможности языка и инфраструктуры, а также отражают опыт, который сообщество извлекло из строительства нескольких версий программных продуктов на C#. Читатели предыдущих изданий заметят, что в это издание включен материал первого издания книги *More Effective C#*, а большое количество советов было удалено. Начиная с данного издания, я реорганизирую обе книги, и новое издание *More Effective C#* будет раскрывать другие концепции. В целом предложенные 50 советов представляют собой набор рекомендаций, которые помогут вам как профессиональному разработчику применять язык C# более эффективно.

В книге предполагается использование версии C# 6.0, но книга не является исчерпывающим изложением новых языковых средств. Подобно всем книгам серии, посвященной эффективной разработке программного обеспечения, она предлагает практические рекомендации о том, как применять эти средства для решения задач, с которыми вы сталкиваетесь ежедневно. Я специально раскрываю новые языковые возможности C# 6.0, которые вводят новые и улучшенные способы написания общих идиом. Поиск в Интернете может по-прежнему давать ссылки на более ранние решения, имеющие многолетнюю историю. Я особо обращаю внимание на старые рекомендации и объясняю, почему усовершенствования языка делают возможными лучшие методы.

Многие рекомендации в этой книге могут быть проверены анализаторами и исправлениями кода, основанными на Roslyn. Хранилище кода доступно по адресу <https://github.com/BillWagner/EffectiveCSharp2016>. Если у вас есть какие-то идеи или вы хотите принять участие, тогда напишите мне запрос.

Кто должен читать эту книгу?

Книга написана для профессиональных разработчиков, использующих C# в качестве части повседневного набора инструментов. В ней предполагается, что вы хорошо знакомы с синтаксисом C# и возможностями языка. Книга не включает в себя обучающие руководства по языковым средствам. Взамен в ней обсуждается, каким образом задействовать все возможности текущей версии языка C# при ежедневной разработке.

В дополнение к средствам языка предполагается наличие определенных знаний общезыковой исполняющей среды (Common Language Runtime – CLR) и оперативного (Just-In-Time – JIT) компилятора.

О чем эта книга?

Существуют языковые конструкции, которые вы будете применять повседневно почти в каждой разрабатываемой программе на C#. В главе 1 раскрываются идиомы языка, которые вы используете настолько часто, что они должны выглядеть подобно поношенным инструментам в ваших руках. Они являются строительными блоками каждого создаваемого типа и каждого реализуемого алгоритма.

Работа в управляемой среде вовсе не означает, что среда освобождает вас от всех обязанностей. Вы по-прежнему должны взаимодействовать со средой для создания корректных программ, которые удовлетворяют установленным требованиям по производительности. Речь идет не только о тестировании и настройке производительности. В главе 2 вы изучите идиомы проектирования, которые позволяют работать со средой для достижения этих целей до того, как начнется обстоятельная оптимизация.

Обобщения являются обеспечивающей технологией для всего остального, что было добавлено в язык C#, начиная с версии C# 2.0. В главе 3 обобщения раскрываются как замена применению `System.Object` и приведений. Кроме того, обсуждаются расширенные приемы, такие как ограничения, обобщенная специализация, ограничения методов и обратная совместимость. Вы научитесь нескольким приемам, в которых обобщения облегчают выражение проектных намерений.

В главе 4 объясняется язык LINQ, синтаксис запросов и связанные средства. Вы узнаете, когда использовать расширяющие методы для отделения контрактов от реализации, как эффективно применять замыкания C# и каким образом программировать с использованием анонимных типов. Вы увидите, как компилятор сопоставляет ключевые слова запросов с вызовами методов, каким образом проводить различие между делегатами и деревьями выражений (а также при необходимости преобразовывать их друг в друга) и как обходиться с запросами, когда нужны скалярные результаты.

В главе 5 предлагается руководство по управлению исключениями и ошибками в современных программах C#. Вы узнаете, каким образом гарантировать надлежащее сообщение об ошибках, а также как сохранять состояние программы согласованным и в идеальном случае неизменным, когда ошибки возникают. Вы научитесь предоставлять лучшую практику отладки разработчикам, применяющим ваш код.

Соглашения, используемые в коде

Демонстрация кода в книге по-прежнему требует соблюдения компромиссов между пространством и ясностью. В коде примеров я старался оставлять только важные строки, чтобы четко отражать их цель. Часто это означает, что другие порции класса или метода не показаны. Иногда ради экономии пространства опускается код восстановления после ошибок. Открытые методы должны проверять достоверность своих параметров и других входных данных, но такой код обычно не приводится, чтобы экономить пространство. По аналогичным соображениям не показывается код проверки вызовов методов и конструкции `try/finally`, которые часто встречались бы в сложных алгоритмах.

Я также обычно допускаю, что большинство разработчиков могут найти распространенные пространства имен, когда они применяются в примерах. Вы можете благополучно полагать, что каждый пример неявно включает следующие операторы `using`:

```
using System;
using static System.Console;
using System.Collections.Generic;
using System.Linq;
using System.Text;
```

Благодарности

Есть немало людей, которых я обязан поблагодарить за их вклад в эту книгу. На протяжении многих лет я был удостоен чести входить в состав удивительного сообщества C#. Каждый участник списка рассылки C# Insiders (как внутри компании Microsoft, так и за ее пределами) содействовал улучшению данной книги своими идеями и обсуждениями.

Я должен специально выделить нескольких членов сообщества C#, которые напрямую помогли мне с идеями и их превращением в конкретные рекомендации. Беседы с Джоном Скитом, Дастином Кемпбеллом, Кевином Пилч-Биссоном, Джаредом Парсонсом, Скоттом Алленом и, что важнее всего, с Мэдсом Торгерсеном стали основой для многих новых идей в этом издании.

При подготовке настоящего издания мне довелось поработать с замечательной командой технических рецензентов. Джейсон Бок, Марк Михаэлис и Эрик Липперт сосредоточенно изучали текст и примеры, чтобы гарантировать качество книги, которую вы теперь держите в руках. Их рецензии были всесторонними и полными, как только можно было в принципе надеяться. Кроме того, они добавили рекомендации, которые помогли мне лучше объяснить многие темы.

Команда в Addison-Wesley оказалась просто мечтой для работы с ней. Трина Макдональд была фантастическим редактором, постановщиком задач и движущей силой всего, что удалось сделать. Она сильно опиралась на Марка Ренфро и Оливию Базейджо, да и сам я тоже. Благодаря их вкладу качество завершенной рукописи было обеспечено от корки до корки. Курт Джонсон проделал невероятную работу по оформлению содержимого. Независимо от выбранного формата книги Курт всегда обеспечит ее существование.

И снова это честь быть частью серии Скотта Мейерса. Он просматривает каждую рукопись и выдвигает предложения и замечания по улучшению. Он крайне основателен, а его опыт в разработке программного обеспечения, пусть даже и не на C#, позволяет находить любые области, где я объяснил совет недостаточно четко или обосновал рекомендацию не полностью. Его отзывы, как всегда, неоценимы.

Моя семья дала мне возможность завершить рукопись. Моя жена Марлен ждала нескончаемые часы, пока я занимался текстом и примерами. Без ее поддержки я никогда бы не завершил эту или любую другую книгу. И у меня не было бы столько радости от окончания работы.

Об авторе

Билл Вагнер является одним из выдающихся разработчиков на C# во всем мире и членом комитета по стандартам C# в организации ECMA. Он президент организации Humanitarian Toolbox, в течение 11 лет был награжден званиями Microsoft Regional Director и .NET MVP, а недавно был назначен в консультативный совет .NET Foundation. Вагнер сотрудничал с компаниями в диапазоне от стартапов до предприятий, которые улучшали процесс разработки программного обеспечения и расширяли свои команды разработчиков. В настоящее время он трудится в составе команды содержимого .NET Core внутри Microsoft. Он создает учебные материалы для разработчиков, заинтересованных в языке C# и платформе .NET Core. Билл получил степень бакалавра по вычислительной технике в Иллинойском университете в Урбане-Шампейне.

Ждем ваших отзывов!

Вы, читатель этой книги, и есть главный ее критик. Мы ценим ваше мнение и хотим знать, что было сделано нами правильно, что можно было сделать лучше и что еще вы хотели бы увидеть изданным нами. Нам интересны любые ваши замечания в наш адрес.

Мы ждем ваших комментариев и надеемся на них. Вы можете прислать нам бумажное или электронное письмо либо просто посетить наш веб-сайт и оставить свои замечания там. Одним словом, любым удобным для вас способом дайте нам знать, нравится ли вам эта книга, а также выскажите свое мнение о том, как сделать наши книги более интересными для вас.

Отправляя письмо или сообщение, не забудьте указать название книги и ее авторов, а также свой обратный адрес. Мы внимательно ознакомимся с вашим мнением и обязательно учтем его при отборе и подготовке к изданию новых книг.

Наши электронные адреса:

E-mail: info@dialektika.com

WWW: <http://www.dialektika.com>

Наши почтовые адреса:

в России: 195027, Санкт-Петербург, Магнитогорская ул., д. 30, ящик 116

в Украине: 03150, Киев, а/я 152