

Введение

Что в этой книге

Эта книга о языке C# версии 2 и выше. Я не рассматриваю собственно язык C# 1, а библиотеки инфраструктуры .NET Framework и общезыковая исполняющая среда (CLR) описаны только в связи с языком C#. Это преднамеренное решение, и в результате получилась книга, которая довольно сильно отличается от большинства книг по C# и .NET, которые я видел.

С учетом наличия у читателя некоторого объема знаний по языку C# 1, я не буду тратить сотни страниц, рассматривая материал, который известен большинству. Это обеспечит мне намного больше места для подробного рассмотрения деталей языка C# версий 2, 3 и 4, для чего, я надеюсь, вы и читаете данную книгу.

Я полагаю, что множество разработчиков были бы менее расстроены результатом своей работы, если бы у них были более глубокие отношения с языком, на котором они пишут. Я понимаю, “отношения” с языком программирования — звучит довольно дико, но это наилучшее, что я смог подобрать. Данная книга — моя попытка помочь вам достичь таких отношений или углубить их. Самой этой книги не будет достаточно, но совместно с практическим программированием она направит вас по правильному пути, предложит интересные пути исследования, а также объяснит, почему ваш код ведет себя так или иначе.

Для кого написана эта книга

Я был приятно удивлен реакцией сообщества на первое издание этой книги. Я заметил, что ее рекомендуют в различных местах, особенно на форуме Stack Overflow (где, я признаю, мое присутствие могло несколько поспособствовать этому). Обычно ее рекомендовали тем, кто действительно желает знать то, что происходит в их коде, но иногда ее предлагали также как средство для обучения новичков. Лично я полагаю, что это неправильно. Здесь вы, вероятно, с трудом отыщете концепции, необходимые для программирования на языке C#1, вы извлечете больше пользы, если будете читать эту книгу вместе с другой книгой, где объяснение начинается с самого начала.

Я не отговариваю вас от покупки данной книги, но хочу заметить, что на рынке есть множество других хороших книг по C#, и они, фактически, хороши как дополнение к этой книге. Я видел множество отзывов читателей, которые начали с менее насыщенных книг, а затем постепенно перешли к этой книге. Я хочу рекомендовать книгу Эндрю Троелсена *Язык программирования C# 2010 и платформа .NET 4.0, 5-е изд.*, в которой не только рассматривается язык C# с самого начала, но и затрагивается ядро инфраструктуры .NET Framework.

Я не собираюсь утверждать, что чтение этой книги повысит вашу квалификацию. Для успешной разработки программного обеспечения нужно несколько больше, чем знание синтаксиса языка, который вы собираетесь использовать. Я уделяю некоторое внимание наставлениям (скорее, больше в этом, втором, издании, чем в первом), однако, в конечном счете, для разработки нужно намного больше внутреннего инстинкта, чем большинство из нас хотело бы признать. Мне бы хотелось, чтобы вы, прочитав и поняв эту книгу, почувствовали себя свободно с языком C# и могли следовать своим инстинктам без лишних опасений. Речь не о том, чтобы уметь писать код, который никто больше не поймет, поскольку в нем используются малоизвестные особенности языка, а именно о том, чтобы быть уверенным в знании доступных для вас возможностей и знать путь, по которому поощряют вас следовать идиомы языка C#.

Структура книги

Структура книги проста — четыре части и три приложения. Часть I включает новые темы по C# 1, которые важны для понимания тем по C# 2 и 3 и которые зачастую понимают неправильно. В части II рассматриваются новые возможности C# 2. Часть III посвящена новым возможностям языка C# 3. Выяснение того, что содержится в части IV, я оставлю читателю как самостоятельное упражнение.

Иногда материал организован так, что мы возвращаемся к одной и той же теме несколько раз, в частности делегаты обсуждаются сначала в контексте C# 2, а затем — в контексте C# 3, но в этом есть смысл. Я предполагаю, что многие читатели будут использовать различные версии для разных проектов: например, вы можете использовать язык C# 3 на работе, а дома экспериментировать с C# 4. Следовательно, имеет смысл разъяснить, что, где и в какой версии находится.

В главе 1 на примере простого фрагмента кода C# 1 показывается, как усовершенствование последующих версий языка позволяет сделать исходный код более читабельным и мощным. Мы рассмотрим исторический контекст возникновения языка C# и технический контекст, в котором он работает как часть платформы: язык C# построен на библиотеках инфраструктуры и мощной среде выполнения, превращающих абстракцию в действительность.

В главе 2 возвратимся к языку C# 1 и трем специфическим аспектам: делегатам, характеристикам системы типов и различиям между типами значений и ссылочными типами. Эти темы разработчики C# 1 зачастую понимают лишь “достаточно хорошо”, но поскольку язык C# развился и значительно их усовершенствовал, четкое понимание основ обязательно для максимального использования новых возможностей.

Глава 3 посвящена самому важному преимуществу языка C# 2 и, вероятно, самому трудному для понимания: обобщениям. Методы и типы могут быть написаны обобщенно, их параметры типа будут заменены реальными типами, указанными в вызывающем коде. Вначале это будет казаться таким же запутанным, как и данное описание, но как только разберетесь в обобщениях, зададитесь вопросом, как вы жили раньше без них.

Если вы когда-либо хотели представить значение `null` как целое число, глава 4 для вас. Она знакомит с типами, допускающими значения `null`, средствами на базе обобщений и преимуществами их поддержки языком, средой выполнения и инфраструктурой.

Глава 5 демонстрирует усовершенствования делегатов в языке C# 2. До сих пор вы, возможно, использовали делегаты только для обработки событий, таких как щелчки на кнопках. Язык C# 2 облегчает создание делегатов, а поддержка библиотекой делает их более применимыми в ситуациях, которые отличаются от обработки событий.

22 Введение

В главе 6 мы исследуем итераторы и простой способ их реализации в языке C# 2. Многие разработчики используют блоки итераторов, но поскольку LINQ to Objects основан на итераторах, их важность повышается. “Ленивый” характер их выполнения — также ключевая часть LINQ.

Глава 7 демонстрирует множество средств меньшего масштаба, добавленных в язык C# 2, которые тем не менее существенно упрощают жизнь разработчика. Разработчики языка постарались смягчить несколько острых углов, присущих языку C# 1, обеспечив в последующих версиях более гибкое взаимодействие с генераторами кода, улучшив поддержку вспомогательных классов, увеличив подробность контроля доступа к свойствам и т.д.

В главе 8 еще раз рассматривается несколько относительно простых средств, но на сей раз в языке C# 3. Почти весь новый синтаксис приспособлен к задачам LINQ, но он также полезен и сам по себе. Обладая анонимными типами, автоматически реализованными свойствами, неявно типизированными локальными переменными и существенно улучшенной поддержкой инициализации, язык C# 3 предоставляет намного больше возможностей, при помощи которых ваш код может выразить свое поведение.

Глава 9 посвящена главной теме языка C# 3 — лямбда-выражениям. Не удовлетворенные разумно кратким синтаксисом, который мы видели в главе 5, разработчики языка упростили работу с делегатами, по сравнению с языком C# 2. Лямбда-выражения способны на большее — они могут быть преобразованы в деревья выражений; это мощнейший способ представить код как данные.

В главе 10 исследуются методы расширения, которые позволяют обмануть компилятор и заставить его поверить, что методы, объявленные в одном типе фактически, принадлежат другому. На первый взгляд это проблема удобочитаемости, но на самом деле — чрезвычайно мощная возможность, жизненно важная для LINQ.

Глава 11 объединяет темы трех предыдущих глав в контексте выражений запросов — краткого, но мощного способа запроса данных. Сначала сосредоточимся на LINQ to Objects, а затем посмотрим, как шаблон выражения запроса применяется способом, позволяющим полностью задействовать другие провайдеры данных.

Глава 12 содержит краткий обзор различных способов использования LINQ. Сначала рассмотрим преимущества выражений запросов в комбинации с деревьями выражений: демонстрация того, как LINQ to SQL позволяет преобразовать то, что выглядит нормально на языке C#, в операторы SQL. Затем перейдем к рассмотрению того, как могут быть разработаны библиотеки для LINQ, например для LINQ to XML. Совместно LINQ и Reactive Extensions демонстрируют два альтернативных подхода к внутреннему процессу запроса, а завершает главу обсуждение дополнения LINQ to Objects собственными операторами LINQ.

Обсуждение языка C# 4 начинается в главе 13, где мы рассмотрим именованные аргументы и необязательные параметры, усовершенствование взаимодействия с моделью COM и обобщенную вариантность. До некоторой степени это разные средства, но именованные аргументы и необязательные параметры обеспечивают поддержку взаимодействия с моделью COM, а также других, весьма специфических возможностей, которые доступны только при работе с объектами COM.

В главе 14 описывается одно крупное средство языка C# 4: динамические типы. Способность динамической привязки членов во время выполнения, а не статической во время компиляции, является серьезным отклонением от концепции C#, но применяется оно выборочно: динамически будет выполняться только тот код, в котором участвуют динамические значения.

Глава 15 посвящена вероятному будущему кода C#, а не новым возможностям самого языка. Мы рассмотрим Code Contracts, а также новый набор библиотек и инструмен-

тальных средств от Microsoft, позволяющих намного подробнее выражать то, в чем нуждаются ваши методы и что они должны гарантировать жестче, чем может сама система типов. Пока это все еще очень ново, но, возможно, через несколько лет вы не подумаете писать новый код без определения его контракта.

Завершаем книгу главой 16, размышляя о том, с чего все начиналось, о текущем состоянии отрасли и ее вероятном будущем.

В приложениях содержится справочный материал. В приложении А рассматриваются стандартные операторы запросов LINQ с некоторыми примерами. Приложение Б представляет базовые обобщенные классы коллекций и интерфейсы, включая новые параллельные коллекции .NET 4. Приложение В содержит краткий обзор различных версий .NET, включая такие разновидности, как Compact Framework и Silverlight.

Терминология и оформление

Большая часть терминологии в книге объясняется по мере изложения, но есть несколько определений, которые стоит выделить здесь. Я использую термины C# 1, C# 2, C# 3 и C# 4 вполне очевидным способом, но в других книгах и на веб-сайтах вы можете встретить обозначения C# 1.0, C# 2.0, C# 3.0 и C# 4.0. Дополнительный номер “.0” кажется мне избыточным, поэтому я опустил его, надеясь, что это и так понятно.

Я применил несколько терминов из книги Марка Михаэлиса по C#, чтобы не путать термин *runtime*, который обозначает исполняющую среду (как в “Общезыковой исполняющей среде”), с моментом времени (как в “переопределение происходит во время выполнения”). Марк использует для обозначения последней концепции термин *execution time* (время выполнения) вместо *compile time* (время компиляции). Это мне кажется вполне разумным, и я надеюсь, что это приобретет более широкую популярность в сообществе. Я тоже вношу свою лепту, используя эти термины в данной книге.

Я часто упоминаю термины “спецификация языка” или просто “спецификация”. Если я не указываю иного, то это означает спецификацию языка C#. Тем не менее доступно несколько версий спецификации, частично в связи с различными версиями самого языка, а частично в связи с процессом стандартизации. Все упомянутые номера версий начиная с C# 4.0 относятся к спецификациям языка от Microsoft.

В книге содержится множество фрагментов кода, выделенных моноширинным шрифтом, вывод листингов оформлен таким же образом. Некоторые листинги сопровождаются аннотациями кода, а иногда код выделен **полужирным шрифтом**. Почти весь код представлен в виде фрагментов, чтобы, оставаясь компактным, он был все еще выполняемым в пределах соответствующей среды. Этой средой является специальный инструмент Snipru, описанный в разделе 1.7. Он доступен для загрузки, наряду со всем кодом книги (в форме фрагментов, полных решений Visual Studio или того и другого), на веб-сайте книги по адресу: csharpindepth.com.

Загрузка исходного кода

Исходный код всех примеров книги доступен на веб-сайте издателя по адресу: www.manning.com/CSharpInDepthSecondEdition или csharpindepth.com.

Форум Author Online и веб-сайт книги

Покупая книгу, вы получаете бесплатный доступ к закрытому веб-форуму издательства Manning, где можете оставить комментарии о книге, задать технические вопросы, а также получить помощь от автора и других пользователей. Для доступа к форуму и подписки на

24 Введение

новости обратитесь по адресу: www.manning.com/CsharpinDepthSecondEdition. Эта страница предоставит информацию о том, как перейти на форум (только после регистрации), какая помощь доступна и каковы правила поведения на форуме.

В дополнение к веб-сайту издательства Manning, я создал веб-сайт поддержки для книги по адресу: csharpindepth.com; он содержит информацию, которая не вошла в книгу, исходный код всех листингов книги для загрузки, а также ссылки на другие ресурсы.

Соглашения, принятые в этой книге

При оформлении книги были использованы соглашения, общепринятые в компьютерной литературе.

- Новые термины в тексте выделяются *курсивом*. Чтобы обратить внимание читателя на отдельные фрагменты текста, также применяется *курсив*.
- Текст программ, функций, переменных, URL, веб-страниц и другой код представлены моноширинным шрифтом.
- Все, что придется вводить с клавиатуры, выделено **полужирным моноширинным** шрифтом.
- Знакоместо в описаниях синтаксиса выделено *курсивом*. Это указывает на необходимость заменить его фактическим именем переменной, параметром или другим элементом, который должен находиться на этом месте `BINDSIZE=(максимальная ширина колонки) * (номер колонки)`.
- Пункты меню и названия диалоговых окон представлены следующим образом: **Menu Option** (Пункт меню).
- Разрыв слишком длинных строк кода, не помещающихся на странице, обозначен специальным символом ↵.

От издательства

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 интересно услышать и любые другие замечания, которые вам хотелось бы высказать авторам.

Мы ждем ваших комментариев. Вы можете прислать письмо по электронной почте или просто посетить наш веб-сервер, оставив на нем свои замечания, — 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 подходящими для вас.

Посылая письмо или сообщение, не забудьте указать название книги и ее авторов, а также ваш e-mail. Мы внимательно ознакомимся с вашим мнением и обязательно учтем его при отборе и подготовке к изданию следующих книг. Наши координаты:

E-mail: info@williamspublishing.com
WWW: <http://www.williamspublishing.com>

Информация для писем из:

России: 127055, г. Москва, ул. Лесная, д. 43, стр. 1
Украины: 03150, Киев, а/я 152