

# Введение

В данной книге вы познакомитесь с новым языком программирования C# (произносится как “си-шарп”), так как, вероятно, вы уже решили, зачем вам нужен этот язык. Хотя ответ на вопрос, зачем нужен еще один язык программирования, не является очевидным. Чем плохи уже давно зарекомендовавшие себя языки, такие как C++, VB, Java, на которых можно создавать отличные программы и которыми пользуются миллионы программистов?

Языки программирования в чем-то похожи на рабочие инструменты — они тоже ориентированы на выполнение определенных задач. Например, отверткой удобнее завинчивать шурупы, чем ножом, хотя при отсутствии таковой можно иногда воспользоваться и ножичком. Аналогично компьютерную игру с большим количеством графики можно написать на языке типа LISP, но на C++ это сделать проще, а еще проще воспользоваться возможностями Direct X.

Язык C# является родным языком среды .NET. Более того, интеграция со средой .NET (произносится как “дот-нет”) была одной из целей, преследуемых при его разработке. Хотя для среды .NET уже можно программировать на многих языках, однако в большинстве случаев лучше всего подойдет именно язык C#.

Язык C# является полностью объектно-ориентированным, что значительно облегчает работу программиста, позволяя создавать надежные, устойчивые к ошибкам компоненты, неоднократно используемые во многих практических ситуациях.

Общезыковая исполняющая среда .NET (CLR — Common Language Runtime), с помощью которой выполняются программы, написанные на языке программирования C#, построена на основе *компонентов*. Вполне естественно, что одной из целей разработки языка C# было упрощение создания компонентов. Язык C# является “компонентно-ориентированным”, так как все объекты оформляются в виде компонентов и все события так или иначе связаны с ними.



Компонентом называется отдельный функциональный блок, доступный для использования в программе и обычно реализуемый в статических библиотеках. Компоненты можно группировать в модули.

Основные характеристики компонентов, такие как свойства, методы и события, имеют первостепенное значение и для среды времени выполнения. С компонентами может быть связана декларативная информация (атрибуты), которая передает сведения о компонентах другим элементам программы на стадии разработки или выполнения. Существует возможность размещения документации в компонентах и ее экспорт в формат XML.

Для создания и применения объектов C# не нужны специальные заголовочные файлы, файлы с описанием используемых интерфейсов или библиотеки типов. Компоненты, созданные на C#, содержат всю необходимую информацию о себе и могут прямо применяться без регистрации в системе, как это приходится сейчас делать с другими компонентами.

Программирование на C# отличается простотой, надежностью и логичностью. Механизм обработки ошибок в нем не выглядит каким-то неуклюжим дополнением, а исключения обрабатываются практически на любом уровне. Язык C# безопасен по отношению к типам. Он защищен от использования неинициализированных переменных, небезопасных преобразований типов и других распространенных ошибок программирования.

Система сборки мусора снимает с программиста тяжелое бремя управления памятью, а проблема написания компонентов для разных версий упрощается благодаря продуманной семантике размещения данных и возможности отделения интерфейса от реализации. Числовые операции можно проверять на предмет переполнения, а у массивов поддерживается проверка границ, что отсутствует, например, в широко распространенном языке программирования C.

Написание программ на языке программирования C# оправдано с позиций затрат времени и денег. Среда разработки обладает всеми преимуществами унифицированной среды и имеет достаточно возможностей для создания управляемых и безопасных программ, но при этом предоставляет доступ и к небезопасным средствам (например, указателям), когда этого требует конкретная ситуация.

Существующие объекты COM могут использоваться так, словно они являются объектами среды .NET. Из программного кода C# можно обращаться к коду C, находящемуся в библиотеках DLL.

Наконец, C# использует все возможности среды .NET, обеспечивающей обширную библиотечную поддержку для решения общих и специализированных задач программирования.

Среда .NET представляет новую компьютерную платформу, упрощающую разработку приложений для распределенной сети Интернет. При разработке среды .NET ставились следующие цели:

- формирование согласованной объектно-ориентированной среды программирования как для локального хранения и выполнения кода, так и для локального выполнения кода, находящегося в сети Интернет, либо для удаленного выполнения;
- создание среды выполнения, устраняющей конфликты при развертывании программного обеспечения и отслеживании версий;
- формирование среды, гарантирующей безопасное выполнение кода, который создан неизвестным разработчиком;
- создание среды выполнения кода, исключающей снижение производительности при выполнении сценариев или интерпретируемого кода;
- согласование возможностей разработчиков для разных типов приложений, таких как приложения Windows и приложения для Web;
- взаимодействие на основе промышленных стандартов, которое обеспечивает объединение кода, разработанного для среды .NET, с любым другим кодом.

Двумя основными составляющими среды .NET являются общезыковая среда выполнения (CLR — Common Language Runtime) и библиотека классов (FCL — Framework Class Library). Общезыковая среда выполнения управляет кодом во время работы программы и предоставляет основные службы, такие как управление памятью, управление потоками и удаленное взаимодействие. При этом используются правила строгой типизации и другие виды проверки достоверности кода, обеспечивающие безопасность и надежность. Основным принципом среды выполнения является концепция управления кодом. Код, который выполняется, используя все возможности среды .NET, называют *управляемым кодом*, а код, использующий запрещенные в среде выполнения средства и поэтому обходящий защитные барьеры, называют *неуправляемым кодом*. Другая составляющая среды .NET — *библиотека классов*, представляет полную объектно-ориентированную коллекцию многократно используемых типов, которые применяются для разработки программ в диапазоне от обычных, запускаемых

с командной строки или имеющих интерфейс пользователя приложений, до приложений, применяющих последние возможности ASP.NET, таких как и Web-службы XML.

Среда .NET может использоваться неуправляемыми компонентами, которые загружают ее в собственные процессы и инициируют выполнение управляемого кода, создавая таким образом среду программного обеспечения, позволяющую применять средства как управляемого, так и неуправляемого выполнения. Браузер Internet может служить примером неуправляемого приложения, использующего среду .NET. Размещение общезыковой среды выполнения в браузере позволяет внедрить управляемые компоненты или элементы управления Windows Forms в HTML-документы.

Даже на основе столь краткого обзора возможностей среды .NET для языка программирования C# можно сделать вывод, что это шаг в будущее и очередной этап развития технологий программирования, и поэтому стоит потратить время на знакомство с новыми возможностями.

Книга рассчитана на тех, кто мало знаком с программированием, но имеет начальные знания об операционной системе Windows и умеет работать с ее окнами. Именно поэтому в книге довольно подробно рассматриваются как среда разработки .NET, так и базовые понятия объектно-ориентированного стиля программирования, без знания которых невозможно объяснить построение программы и работу с ней в редакторе кодов или проектировщике форм.

## **От издательства “Диалектика”**

Вы, читатель этой книги, и есть ее главный 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 Web-сервер и оставить свои замечания там. Одним словом, любым удобным для вас способом дайте нам знать, нравится вам эта книга или нет, 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](mailto:info@dialektika.com)

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

Наши почтовые адреса:

в России: 115419, Москва, а/я 783

в Украине: 03150, Киев, а/я 152