УДК 004.43
ББК 32.973.26-018.1
В19

Васильев, Алексей.

В19 Программирование на C# для начинающих. Основные сведения / Алексей Васильев. — Москва : Эксмо, 2018. — 592 с. — (Российский компьютерный бестселлер).

ISBN 978-5-04-092519-3

В этой книге Алексей Васильев, доктор физико-математических наук и автор популярных российских самоучителей по программированию, приглашает читателей ознакомиться с основами языка C#. Прочитав ее, вы узнаете историю языка, его структуру, ознакомитесь с типами данных и переменными, операторами, циклами и множеством другой полезной информации, необходимой для работы с этим языком.

УДК 004.43
ББК 32.973.26-018.1

ISBN 978-5-04-092519-3

ОГЛАВЛЕНИЕ

Введение

ЯЗЫК C# И ТЕХНОЛОГИЯ .NET FRAMEWORK

Русская речь не сложнее других. Вон Маргадон — дикий человек — и то выучил.

из к/ф «Формула любви»

Язык C# уже многие годы неизменно входит в список языков программирования, самых востребованных среди разработчиков программного обеспечения. Язык является базовым для технологии .Net Framework, разработанной и поддерживаемой корпорацией Microsoft. Именно языку C# посвящена книга, которую читатель держит в руках.

История создания языка C#

История, леденящая кровь. Под маской овцы скрывался лев.

из к/ф «Покровские ворота»

Язык C# создан инженерами компании Microsoft в 1998–2001 годах. Руководил группой разработчиков Андерс Хейлсберг, который до того трудился в фирме Borland над созданием компилятора для языка Pascal и участвовал в создании интегрированной среды разработки Delphi. Язык C# появился после языков программирования C++ и Java. Богатый опыт их использования был во многом учтен разработчиками C#.

НА ЗАМЕТКУ

Синтаксис языка C# похож на синтаксис языков C++ и Java. Но сходство внешнее. У языка C# своя уникальная концепция. Вместе с тем многие управляющие инструкции в языке C# будут знакомы тем, кто программирует в C++ и Java.

Вообще же из трех языков программирования C++, Java и C# исторически первым появился язык C++. Затем на сцену вышел язык Java. И уже после этого появился язык программирования C#.

Для понимания места и роли языка C# на современном рынке программных технологий разумно начать с языка программирования C, который в свое время стал мощным стимулом для развития программных технологий как таковых. Именно из языка C обычно выводят генеалогию языка C#. Мы тоже будем придерживаться классического подхода.

Язык программирования C появился в 1972 году, его разработал Денис Ритчи. Язык C постепенно набирал популярность и в итоге стал одним из самых востребованных языков программирования. Этому способствовал ряд обстоятельств. В первую очередь, конечно, сыграл роль лаконичный и простой синтаксис языка. Да и общая концепция языка C оказалась исключительно удачной и живучей. Поэтому когда встал вопрос о разработке нового языка, который бы поддерживал парадигму объектно ориентированного программирования (ООП), то выбор естественным образом пал на язык C: язык программирования C++, появившийся в 1983 году, представлял собой расширенную версию языка C, адаптированную для написания программ с привлечением классов, объектов и сопутствующих им технологий. В свою очередь, при создании языка программирования Java отправной точкой стал язык C++. Идеология языка Java отличается от идеологии языка C++, но при всем этом базовые управляющие инструкции и операторы в обоих языках схожи.

НА ЗАМЕТКУ

Язык программирования Java официально появился в 1995 году и стал популярным благодаря универсальности программ, написанных на этом языке. Технология, используемая в Java, позволяет писать переносимые программные коды, что исключительно важно при разработке приложений для использования в Internet.

Нет ничего удивительного, что при создании языка программирования C# традиция была сохранена: синтаксис языка C# во многих моментах будет знаком тем, кто уже программирует на C++ и Java. Хотя такое сходство — внешнее. Языки очень разные, поэтому расслабляться не стоит. Да и базовые синтаксические конструкции в языке C# имеют свои особенности. Все это мы обсудим в книге.

Особенности языка C#

> Обо мне придумано столько небылиц, что я устаю их опровергать.
>
> *из к/ф «Формула любви»*

Язык программирования C# — простой, красивый, эффективный и гибкий. С помощью программ на языке C# решают самые разные задачи. На C# можно создавать как небольшие консольные программы, так и программы с графическим интерфейсом. Код, написанный на языке C#, лаконичен и понятен (хотя здесь, конечно, многое зависит от программиста). В этом отношении язык программирования C# практически не имеет конкурентов.

НА ЗАМЕТКУ

Язык C# создавался после появления языков C++ и Java. В C# были учтены и по возможности устранены «недостатки» и «недоработки», которые есть в C++ и Java. Иногда язык C# упоминается как усовершенствованная версия языков C++ и Java, хотя концепции у них совершенно разные, так что утверждение довольно поверхностно.

Кроме собственно преимуществ языка C#, немаловажно и то, что язык поддерживается компанией Microsoft. Поэтому он идеально подходит, чтобы писать программы для выполнения под управлением операционной системы Windows.

ПОДРОБНОСТИ

Операционной системой Windows и технологией .Net Framework компании Microsoft область применения языка C# не ограничивается. Существуют альтернативные проекты (такие, например, как Mono), позволяющие выполнять программы, написанные на языке C#, под управлением операционных систем, отличных от Windows. Вместе с тем мы в книге будем исходить из того, что используются «родные» средства разработки и операционная система Windows.

Как отмечалось выше, язык C# является неотъемлемой частью технологии (или платформы) .Net Framework. Основу платформы .Net Framework составляет среда исполнения CLR (сокращение от Common Language Runtime) и библиотека классов, которая используется при программировании на языке C#.

НА ЗАМЕТКУ

Платформа .Net Framework позволяет использовать и иные языки программирования, а не только C# — например, C++ или Visual Basic. Возможности платформы .Net Framework позволяют объединять «в одно целое» программные коды, написанные на разных языках программирования. Это очень мощная технология, но для нас интерес представляет написание программных кодов на языке C#. Именно особенности и возможности языка C# мы будем обсуждать в книге.

При компилировании программного кода, написанного на языке C#, создается промежуточный код. Это промежуточный код реализован на языке MSIL (сокращение от Microsoft Intermediate Language). Промежуточный код выполняется под управлением системы CLR. Система CLR запускает JIT-компилятор (сокращение от Just In Time), который, собственно, и переводит промежуточный код в исполняемые инструкции.

ПОДРОБНОСТИ

Файл с программным кодом на языке C# сохраняется с расширением `.cs`. После компиляции программы создается файл с расширением `.exe`. Но выполнить этот файл можно только на компьютере, где установлена система .Net Framework. Такой код называется управляемым (поскольку он выполняется под управлением системы CLR).

Описанная нетривиальная схема компилирования программ с привлечением промежуточного кода служит важной цели. Дело в том, что технология .Net Framework ориентирована на совместное использование программных кодов, написанных на разных языках программирования. Базируется эта технология на том, что программные коды с разных языков программирования «переводятся» (в процессе компиляции) в промежуточный код на общем универсальном языке. Проще говоря, коды на разных языках программирования приводятся «к общему знаменателю», которым является промежуточный язык MSIL.

ПОДРОБНОСТИ

Программы, написанные на Java, тоже компилируются в промежуточный байт-код. Байт-код выполняется под управлением виртуальной машины Java. Но по сравнению с языком C# имеется принципиальное отличие. Байт-код, в который переводится при компиляции Java-программа, имеет привязку к одному языку программирования — языку

Java. И в Java схема с промежуточным кодом нужна для обеспечения универсальности программ, поскольку промежуточный байт-код не зависит от типа операционной системы (и поэтому переносим). Особенность операционной системы учитывается той виртуальной машиной Java, которая установлена на компьютере и выполняет промежуточный байт-код. Промежуточный код, который используется в технологии .Net Framework, не привязан к конкретному языку программирования. Например, что при компиляции программы на языке C#, что при компиляции программы на языке Visual Basic получаются наборы инструкций на одном и том же промежуточном языке MSIL. И нужно это, еще раз подчеркнем, для обеспечения совместимости разных программных кодов, реализованных на разных языках.

Весь этот процесс для нас важен исключительно с познавательной точки зрения. Более принципиальными являются вопросы, касающиеся особенностей собственно языка C#.

Первое, что стоит отметить: язык C# полностью объектно ориентированный. Это означает, что даже самая маленькая программа на языке C# должна содержать описание хотя бы одного класса.

НА ЗАМЕТКУ

Здесь есть методологическая проблема. Состоит она в том, что неподготовленному читателю сложно воспринимать концепцию ООП сразу, без предварительной подготовки. Мы найдем выход в том, чтобы использовать определенный шаблон при написании программ. Затем, по мере знакомства с языком C#, многие моменты станут простыми и понятными.

Как отмечалось выше, базовые синтаксические конструкции языка C# напоминают (или просто совпадают) соответствующие конструкции в языках C++ и/или Java. Но, кроме них, язык C# содержит множество интересных и полезных особенностей, с которыми мы познакомимся в книге.

ПОДРОБНОСТИ

Знакомым с языками программирования C++ и/или Java будет полезно узнать, что в языке C#, как и в языке C++, используются пространства имен, указатели, существует переопределение операторов. Также в C#, как и в Java, имеются интерфейсы, объекты реализуются через ссылки, используется система автоматической

«уборки мусора». А еще в C# используются делегаты, концепция которых идеологически близка к концепции указателей на функции в C++. Массивы в C# больше напоминают массивы Java, но вообще в C# они достаточно специфичные. Индексаторы в C# позволяют индексировать объекты — подобный механизм, основанный на переопределении оператора «квадратные скобки», есть в C++. Свойства, которые используются в C#, представляют собой нечто среднее между полем и методом, хотя, конечно, больше напоминают поле с особым режимом доступа.

Есть в языке C# и «классические» механизмы, характерные для большинства языков, поддерживающих парадигму ООП. Мы познакомимся с тем, как на основе классов и объектов в языке C# реализуется инкапсуляция, узнаем, что такое наследование и как в C# используется полиморфизм. Будет много других тем и вопросов, которые мы изучим. Например, мы узнаем (во второй части книги), как в C# создаются приложения с графическим интерфейсом, и познакомимся с основными подходами, используемыми при создании многопоточных приложений. Но обо всем этом — по порядку, тема за темой.

Программное обеспечение

Показывай свою гравицапу. Если фирменная вещь — возьмем!

из к/ф «Кин-дза-дза»

Для работы с языком программирования C# на компьютере должна быть установлена платформа .Net Framework. Компилятор языка C# входит в стандартный набор этой платформы. Поэтому, если на компьютере установлена платформа .Net Framework, этого достаточно для начала программирования в C#. Обычно дополнительно еще устанавливают среду разработки, благодаря которой процесс программирования на C# становится простым и приятным.

Если мы хотим написать программу (на языке C#), сначала нам нужно набрать соответствующий программный код. Теоретически сделать это можно в обычном текстовом редакторе. В таком случае набираем в текстовом редакторе код программы и сохраняем файл с расширением `.cs` (расширение для файлов с программами, написанными на языке C#).

После того как код программы набран и файл с программой сохранен, ее следует откомпилировать. Для этого используют программу-компилятор `csc.exe`, которая устанавливается, как отмечено выше, при установке платформы .Net Framework.

ПОДРОБНОСТИ

Алгоритм действий такой: в командной строке указывается название программы-компилятора `csc.exe`, а затем через пробел указывается название файла с программой на языке C#. Допустим, мы записали код программы в файл `MyProgram.cs`. Тогда для компиляции программы в командной строке используем инструкцию `csc.exe MyProgram.cs` или `csc MyProgram.cs` (расширение `exe`-файла можно не указывать). Если компиляция проходит нормально, то в результате получаем файл с расширением `.exe`, а название файла совпадает с названием исходного файла с программой (в нашем случае это `MyProgram.exe`). Полученный в результате компиляции `exe`-файл запускают на выполнение.

Файл `csc.exe` по умолчанию находится в каталоге `C:\Windows\Microsoft.NET\Framework` внутри папки с номером версии — например, `v3.5` или `v4.0`. Также для компилирования программы из командной строки придется, скорее всего, выполнить некоторые дополнительные настройки — например, в переменных среды задать путь для поиска компилятора `csc.exe`.

Хотя такая консольная компиляция вполне рабочая, прибегают к ней редко. Причина в том, что не очень удобно набирать код в текстовом редакторе, затем компилировать программу через командную строку и запускать вручную исполнительный файл. Намного проще воспользоваться специальной программой — *интегрированной средой разработки* (IDE от Integrated Development Environment). Среда разработки содержит в себе все наиболее важные «ингредиенты», необходимые для «приготовления» такого «блюда», как программа на языке C#. При работе со средой разработки пользователь получает в одном комплекте редактор кодов, средства отладки, компилятор и ряд других эффективных утилит, значительно облегчающих работу с программными кодами. Мы тоже прибегнем к помощи среды разработки. Нам понадобится приложение Microsoft Visual Studio.

Загрузить установочные файлы можно с сайта компании Microsoft `www.microsoft.com` (на сайте следует найти страницу загрузок).

После установки среды разработки мы получаем все, что нужно для успешной разработки приложений на языке C#.

ПОДРОБНОСТИ

Приложение Microsoft Visual Studio является коммерческим. Однако у него есть некоммерческая «упрощенная» версия Visual Studio Express, которая вполне подойдет для изучения языка программирования C#.

НА ЗАМЕТКУ

Кроме языка программирования C# среда разработки Visual Studio позволяет создавать программы на языках Visual Basic и C++. Часть настроек, влияющих на функциональность среды разработки, определяется в процессе установки.

Как выглядит окно приложения Visual Studio Express (версия 2015), показано на рис. В.1.

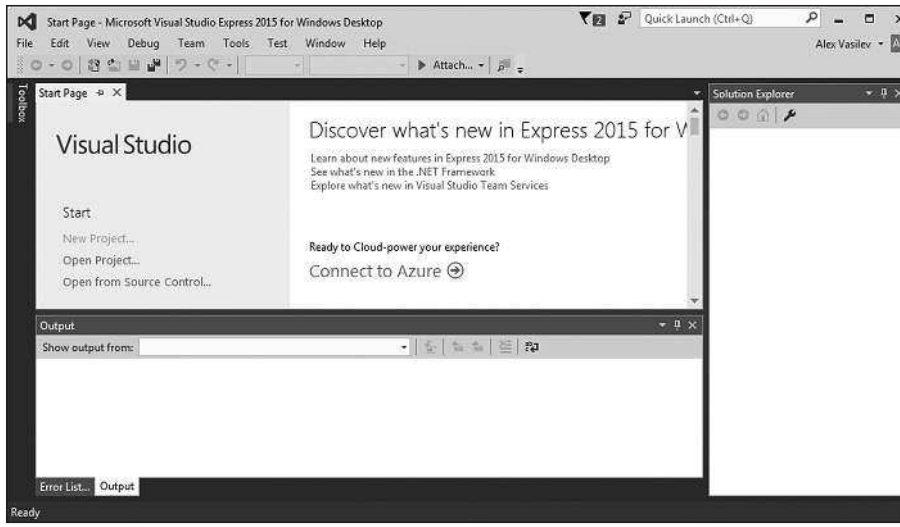

Рис. В.1. Окно приложения Visual Studio Express 2015

Основные операции, выполняемые при создании приложений с помощью среды разработки Visual Studio Express, мы кратко рассмотрим в основной части книги, когда будем обсуждать программные коды.

Среда разработки Visual Studio хотя и предпочтительная, но далеко не единственная. Существуют другие приложения, используемые при создании программ на языке C#. Причем масштабы использования альтернативных средств разработки постоянно расширяются. Но, в любом случае, имеет смысл знать, какие есть варианты помимо Visual Studio. Нас интересуют в первую очередь среды разработки для программирования на C#, распространяемые на некоммерческой основе.

Среда разработки SharpDevelop является приемлемым выбором и позволяет создавать приложения разного уровня сложности. По сути, данная среда разработки является интегрированным отладчиком, взаимодействующим со средой .Net Framework. Окно среды разработки SharpDevelop показано на рис. В.2.

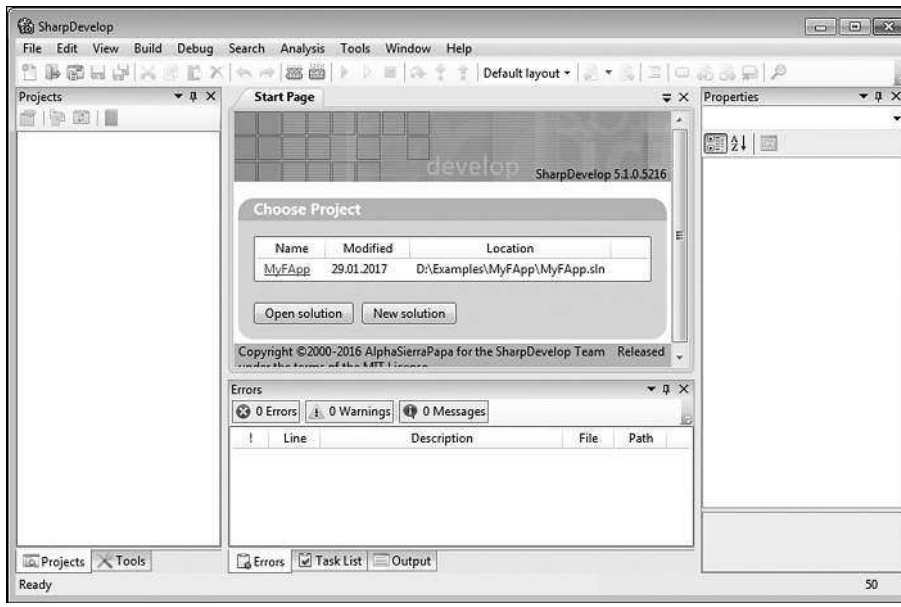

Рис. В.2. Окно среды разработки SharpDevelop

Загрузить все необходимые файлы для установки среды разработки SharpDevelop можно на сайте проекта `www.icsharpcode.net`.

Еще один проект, заслуживающий внимания, называется Mono (сайт поддержки проекта `www.mono-project.com`). Проект развивается в основном благодаря поддержке компании Xamarin (сайт `www.xamarin.com`). В рамках этого проекта была создана среда

разработки MonoDevelop (сайт `www.monodevelop.com`), предназначенная, кроме прочего, для создания приложений на языке C#. На данный момент разработчикам для установки предлагается среда разработки Xamarin Studio, которая позиционируется как продолжение проекта MonoDevelop. Окно среды разработки Xamarin Studio показано на рис. В.3.

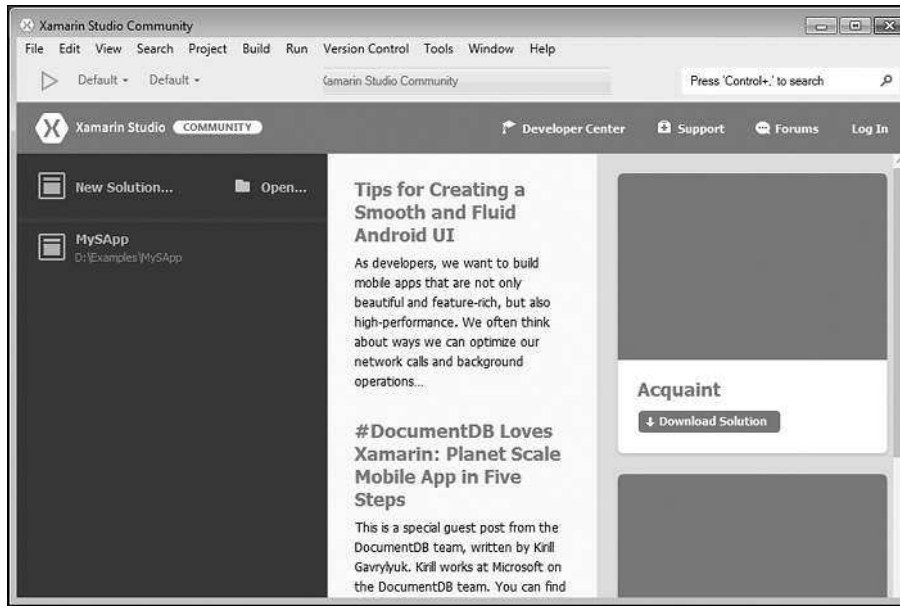

Рис. В.3. Окно среды разработки Xamarin Studio

Методы работы с перечисленными выше средами разработки не являются предметом книги. Отметим лишь следующие обстоятельства:

- У читателя есть альтернатива в плане использования среды разработки.
- Среды разработки не являются эквивалентными по своим функциональным возможностям. Наиболее надежный вариант связан с использованием среды разработки Microsoft Visual Studio. Все примеры из книги тестировались именно в этой среде разработки.
- Нужно иметь в виду, что в силу специфики сред разработки SharpDevelop и Xamarin Studio некоторые программные коды, выполняемые в Microsoft Visual Studio, могут не выполняться в SharpDevelop и Xamarin Studio (и наоборот).