

# 1

## Привет, C#! Здравствуй, .NET Core!

Эта глава посвящена настройке вашей среды разработки, сходствам и различиям между .NET Core, .NET Framework, .NET Standard и .NET Native, а также использованию различных инструментов для создания простейших приложений с помощью C# 7 и .NET Core.

Многие люди читают сложную документацию, предпочитая имитировать и повторять код за автором вместо глубокого изучения теории. Поэтому я тоже не стану объяснять каждое ключевое слово или шаг. Идея в следующем: дать вам задание написать некий код, собрать приложение и посмотреть, что происходит при запуске. Вам не нужно будет разбираться, как все работает.

Выражаясь словами Сэмюэля Джонсона (Samuel Johnson), составившего в 1755 году толковый словарь английского языка, я, вероятно, допустил «несколько диких промахов и забавных несуразиц, без которых не обходится ни одна из работ подобной сложности». Я принимаю на себя полную ответственность за них и надеюсь, что вы оцените мою попытку «*попасть в струю*» и написать книгу о .NET Core и инструментах командной строки этой среды в процессе ее непростого рождения в 2016–2017 годах.

В этой главе:

- настройка среды разработки;
- знакомство с .NET;
- написание и компиляция кода с помощью интерфейса командной строки .NET Core;
- написание и компиляция кода с применением Microsoft Visual Studio 2017;
- написание и компиляция кода с использованием Microsoft Visual Studio Code;
- написание и компиляция кода с помощью Microsoft Visual Studio для Mac;
- управление исходным кодом с применением GitHub.

## Настройка среды разработки

Прежде чем приступить непосредственно к программированию, вам нужно настроить *интегрированную среду разработки* (Integrated Development Environment, IDE), содержащую редактор кода C#. Корпорация Microsoft выпустила целое семейство IDE:

- ❑ Visual Studio 2017;
- ❑ Visual Studio для Mac;
- ❑ Visual Studio Code.

Самая зрелая и полнофункциональная среда — *Microsoft Visual Studio 2017*, но, к сожалению, ее можно запустить на компьютерах только под управлением операционной системы Windows.

Самой современной и упрощенной кросс-платформенной средой разработки, также созданной корпорацией Microsoft, является *Microsoft Visual Studio Code*. Ее можно запустить во всех распространенных операционных системах, включая Windows, macOS и множество разновидностей Linux, таких как *Red Hat Enterprise Linux (RHEL)* и Ubuntu.



Чтобы решить, подходит ли вам среда Visual Studio Code, я рекомендую посмотреть видеоролик, доступный по ссылке <https://channel9.msdn.com/Blogs/raw-tech/Beginners-Guide-to-VS-Code>.

Для создания мобильных приложений наиболее подходит *Visual Studio для Mac*. Разрабатывать приложения для устройств под управлением операционной системы iOS (iPhone и iPad), tvOS, macOS и watchOS поможет компьютер с установленной системой macOS и среда Xcode. Хотя вы и можете использовать среду Visual Studio 2017 с расширениями Xamarin для *написания* кода кросс-платформенного мобильного приложения, вам все равно понадобится система macOS и среда Xcode для *компиляции* этого приложения.

В табл. 1.1 я перечислил среды разработки и операционные системы, которые могут или должны использоваться для выполнения примеров из каждой главы этой книги.

**Таблица 1.1**

Глава	Среда разработки	Операционная система
1–16	Visual Studio 2017	Windows 7 SP1 или выше
1–16	Visual Studio Code	Windows, macOS или Linux
1–16	Visual Studio для Mac	macOS
17	Visual Studio 2017	Windows 10
18	Visual Studio для Mac	macOS



Если есть такая возможность, я рекомендую попытаться выполнить все упражнения в обеих средах: как в Visual Studio 2017 в операционной системе Windows, так и в Visual Studio Code в macOS, Linux или Windows. Так вы получите прекрасный опыт программирования на С# и .NET Core с помощью различных инструментов разработки в разных операционных системах.

При работе над этой книгой я использовал программное обеспечение следующих версий (рис. 1.1):

- ❑ Visual Studio 2017 в операционной системе Windows 10 (в виртуальной машине);
- ❑ Visual Studio для Mac в операционной системе macOS;
- ❑ Visual Studio Code в операционной системе macOS;
- ❑ Visual Studio Code в операционной системе Red Hat Enterprise Linux (не показано на рисунке).

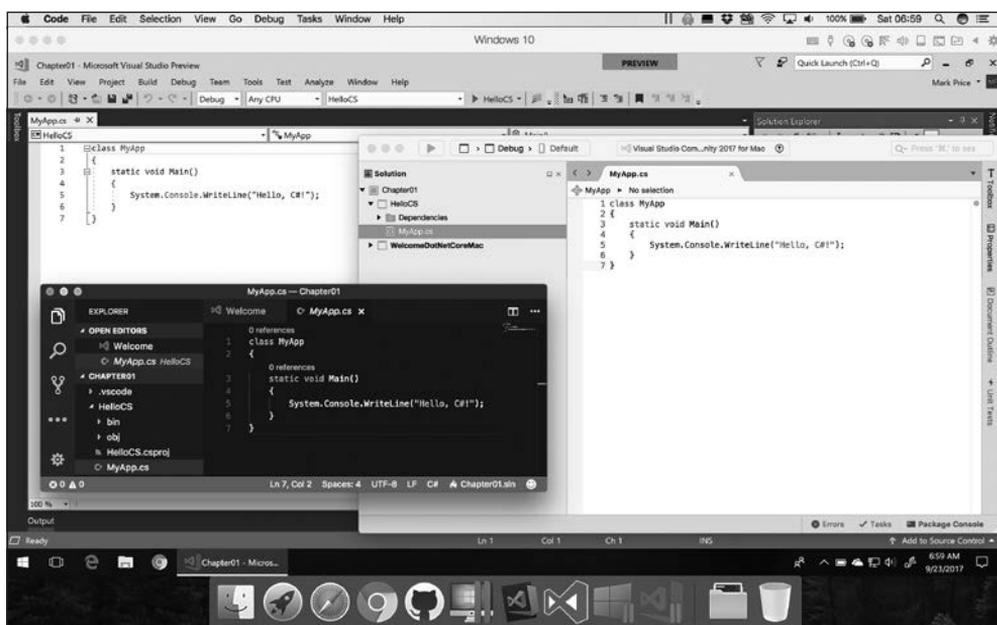


Рис. 1.1

## Альтернативные среды разработки С#

Существуют и альтернативные среды разработки для программирования на языке С#, например *MonoDevelop* и *JetBrains Rider*. Вы можете установить их, воспользовавшись следующими URL.

- ❑ Для установки среды разработки MonoDevelop посетите сайт <http://www.monodevelop.com/>.

- ❑ Для установки среды разработки JetBrains Rider посетите сайт <https://www.jetbrains.com/rider/>.

Кроме того, существует среда разработки *Cloud9*, запускаемая прямо в браузере, так что она еще более кросс-платформенная, чем остальные. Благодаря этой особенности среда набирает популярность. См. дополнительную информацию по адресу <https://c9.io/web/sign-up/free>.

## Кросс-платформенное развертывание

Выбор среды разработки и используемой операционной системы не влияет на место развертывания программы. .NET Core 2.0 поддерживает для развертывания следующие платформы:

- ❑ Windows 7 SP1 или версии выше;
- ❑ Windows Server 2008 R2 SP1 или версии выше;
- ❑ Windows IoT 10 или версии выше;
- ❑ macOS Sierra (версия 10.12) или версии выше;
- ❑ Red Hat Enterprise Linux 7.3 или версии выше;
- ❑ Ubuntu 14.04 LTS или версии выше;
- ❑ Fedora 25 или версии выше;
- ❑ Debian 8.7 или версии выше;
- ❑ OpenSUSE 42.2 или версии выше;
- ❑ Tizen 4 или версии выше.



Операционные системы Linux как платформы для хостинга серверов популярны, поскольку относительно легковесны и более рентабельны в плане масштабируемости, в отличие от операционных систем наподобие Windows и macOS, скорее направленных на конечного пользователя.

В следующем подразделе вы установите среду Microsoft Visual Studio 2017 в операционной системе Windows. Если предпочитаете использовать программу Microsoft Visual Studio Code, то пропустите этот текст и переходите к подразделу «Установка Microsoft Visual Studio Code для Windows, macOS или Linux». Если же предпочитаете Microsoft Visual Studio для Mac, то переходите к подразделу «Установка Microsoft Visual Studio для macOS».

## Установка Microsoft Visual Studio 2017

Для выполнения примеров из большинства глав этой книги можно использовать операционную систему Windows версии 7 SP1 или более поздней, но работать будет удобнее, если применять Windows 10 с установленным обновлением Fall Creators Update.

Начиная с октября 2014 года корпорация Microsoft стала выпускать новую полнофункциональную редакцию среды Visual Studio, доступную всем желающим бесплатно. Эта редакция называется *Community Edition*.

Скачайте пакет Microsoft Visual Studio 2017 версии не ниже 15.4 с сайта <https://www.visualstudio.com/downloads/>.



Вы должны установить пакет Visual Studio 2017 версии не ниже 15.4, чтобы иметь возможность взаимодействовать с .NET Core для универсальной платформы Windows. Версия Visual Studio 2017 не ниже 15.3 требуется для работы с .NET Core 2.0. Более старые версии Visual Studio 2017 поддерживают только .NET Core 1.0 и 1.1.

## Выбор рабочих нагрузок

На вкладке Workloads (Рабочие нагрузки) установите флажки напротив следующих компонентов (рис. 1.2):

- Universal Windows Platform development (Разработка приложений для универсальной платформы Windows);
- .NET desktop development (Разработка классических приложений .NET);
- ASP.NET and web development (ASP.NET и разработка веб-приложений);
- Azure development (Разработка для Azure);
- Node.js development (Разработка Node.js);
- .NET Core cross-platform development (Кросс-платформенная разработка .NET Core).

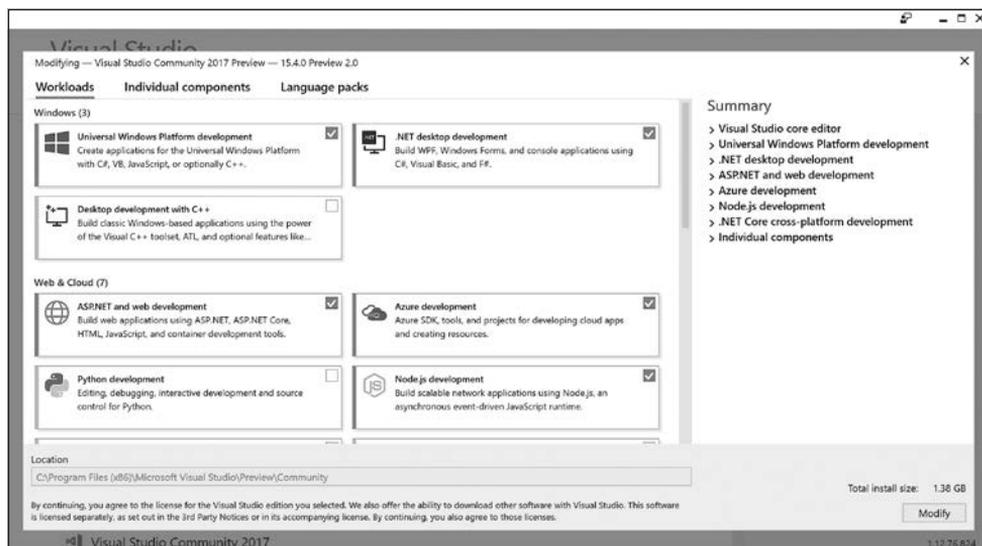


Рис. 1.2

## Выбор дополнительных компонентов

На вкладке **Individual components** (Отдельные компоненты) установите флажки у следующих компонентов:

- Class Designer** (Конструктор классов);
- GitHub extension for Visual Studio** (Расширение GitHub для Visual Studio);
- PowerShell tools** (Инструменты PowerShell).

Нажмите кнопку **Install** (Установить) и подождите, пока выбранные компоненты не будут установлены. После завершения процесса установки нажмите кнопку **Launch** (Запустить).



Пока вы дожидаетесь установки среды разработки Visual Studio 2017, можно перейти к разделу «Знакомство с .NET» далее в этой главе.

Когда вы запустите Visual Studio 2017 в первый раз, увидите окно с запросом авторизации. Если вы ранее зарегистрировали учетную запись Microsoft, то можете воспользоваться регистрационными данными такого аккаунта. В случае отсутствия учетной записи зарегистрируйте новую, перейдя по ссылке <https://signup.live.com/>.

При первом запуске Visual Studio 2017 вы увидите окно с предложением настроить среду под свои предпочтения. В раскрывающемся списке **Development Settings** (Настройки разработки) выберите пункт **Visual C#**. В качестве цветовой схемы я выбрал вариант **Blue** (Голубая), но ваш выбор может быть другим.

После запуска Microsoft Visual Studio вы увидите пользовательский интерфейс программы с открытой начальной страницей в центральной области. Как и большинство других классических Windows-приложений, интерфейс Visual Studio содержит строку меню, панель инструментов с часто вызываемыми командами и строку состояния внизу окна. В правой части окна программы расположена панель **Solution Explorer** (Обозреватель решений) со списком открытых проектов (рис. 1.3).



Для последующего быстрого запуска среды Visual Studio щелкните правой кнопкой мыши на значке этой программы на панели задач Windows и в контекстном меню выберите пункт **Pin this program to taskbar** (Закрепить на панели задач).

Для работы с главами 14, 15 и 16 необходимо установить платформу Node.js и менеджер пакетов NPM.

Скачайте дистрибутив Node.js для операционной системы Windows с сайта <https://nodejs.org/en/download/>.

Запустите мастер установки Node.js (рис. 1.4).

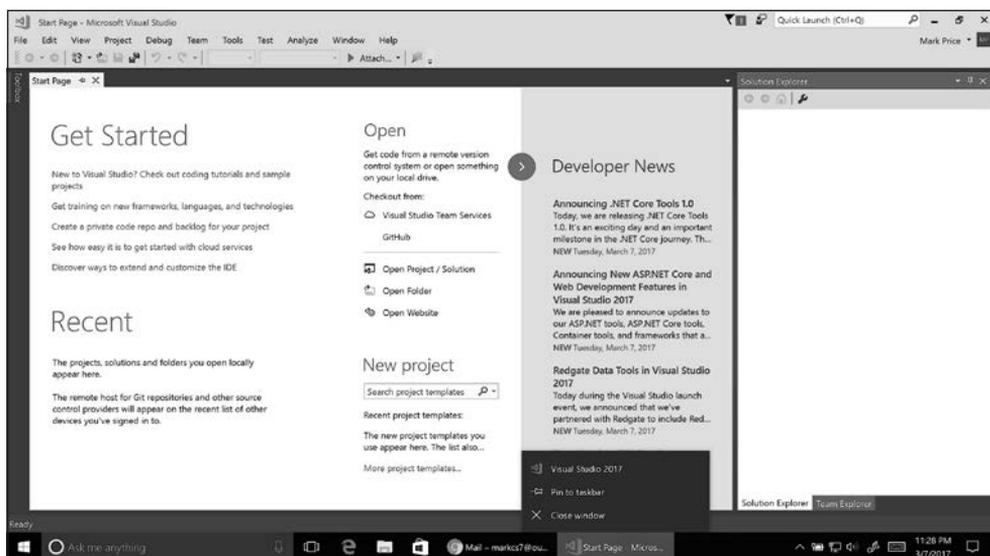


Рис. 1.3

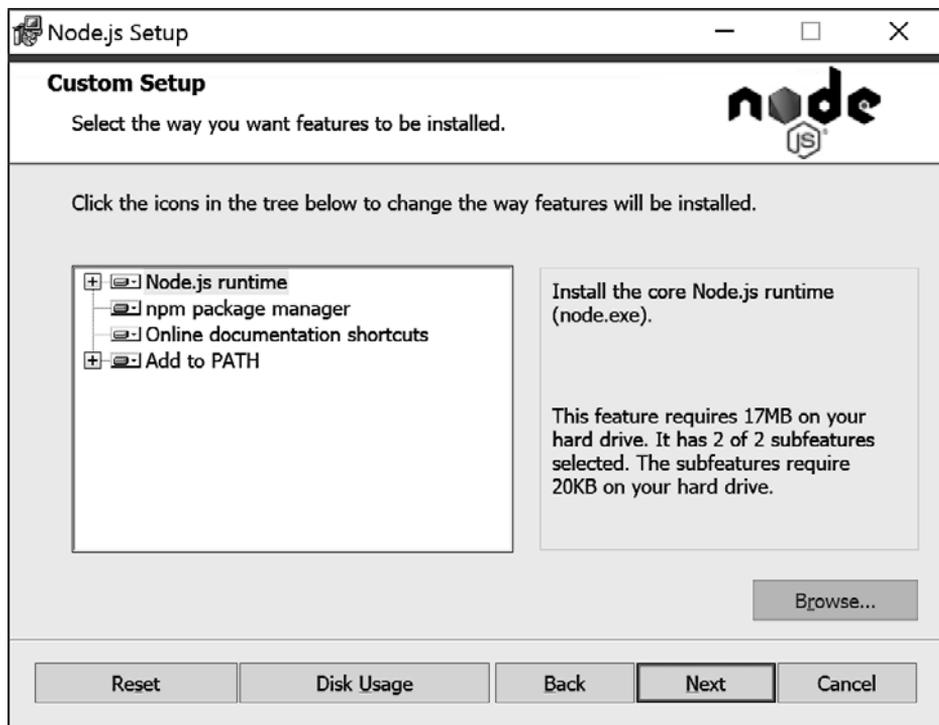


Рис. 1.4

## Установка Microsoft Visual Studio Code

В период с июня 2015-го по сентябрь 2017 года корпорация Microsoft выпускала новую версию среды Visual Studio Code ежемесячно. Эта среда стала быстро развиваться и поразила сотрудников корпорации Microsoft растущей популярностью. Даже если вы планируете пользоваться Visual Studio 2017 или Visual Studio для Mac как основным инструментом разработки, я рекомендую научиться применять среду Visual Studio Code и средства командной строки .NET Core.

Скачать дистрибутив Visual Studio Code можно на сайте <https://code.visualstudio.com/>.



Ознакомиться с планами корпорации Microsoft для Visual Studio Code на 2018 год можно, перейдя по ссылке <https://github.com/Microsoft/vscode/wiki/Roadmap>.

### Установка Microsoft Visual Studio Code для Windows, macOS или Linux

Выполняя примеры и создавая снимки рабочего процесса, я использовал версию Visual Studio Code для операционной системы macOS. Шаги для версий Visual Studio Code для операционной системы Windows и различных модификаций Linux практически неотличимы, поэтому я не буду приводить инструкции для каждой из платформ.



Инструкции по выбору языка интерфейса среды разработки Visual Studio Code приведены на странице [code.visualstudio.com/docs/getstarted/locales](https://code.visualstudio.com/docs/getstarted/locales).

После загрузки дистрибутива Visual Studio Code для операционной системы macOS перетащите его значок на папку Applications (Приложения) (рис. 1.5).

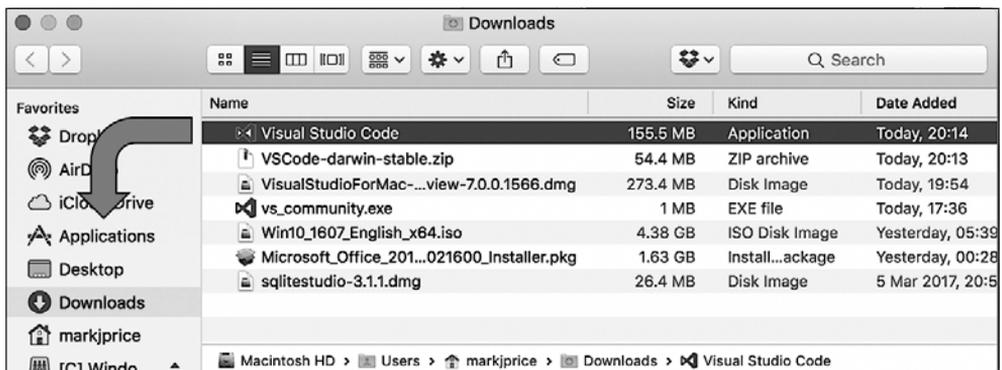


Рис. 1.5

Теперь нужно установить пакет инструментов .NET Core SDK для операционной системы macOS. Полная инструкция вместе с обучающим видеороликом доступна

по ссылке <https://www.microsoft.com/net/core#macos>, а в книге я приведу лишь базовые инструкции, как мы и договаривались.

Сначала требуется установить менеджер пакетов Homebrew (если, конечно, он еще не установлен).

Запустите встроенное в операционную систему macOS приложение Terminal (Терминал) и введите следующую команду в окне консоли:

```
/usr/bin/ruby -e "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

Приложение Terminal (Терминал) выведет запрос о необходимости нажатия клавиши Enter для продолжения, а затем предложит ввести пароль администратора.



Если вы пользуетесь версией .NET Core 1.0 или 1.1, то на данном этапе Homebrew понадобится для установки криптографического пакета OpenSSL, необходимого для работы старых версий .NET Core в операционной системе macOS.

## Установка .NET Core SDK в macOS

Следующий шаг — загрузка и установка дистрибутива .NET Core SDK для операционной системы macOS (x64). Скачать этот пакет можно по ссылке <https://www.microsoft.com/net/download/core>.

Запустите установочный пакет `dotnet-sdk-2.0.0-sdk-osx-x64.pkg`, чтобы перейти к его установке (рис. 1.6).

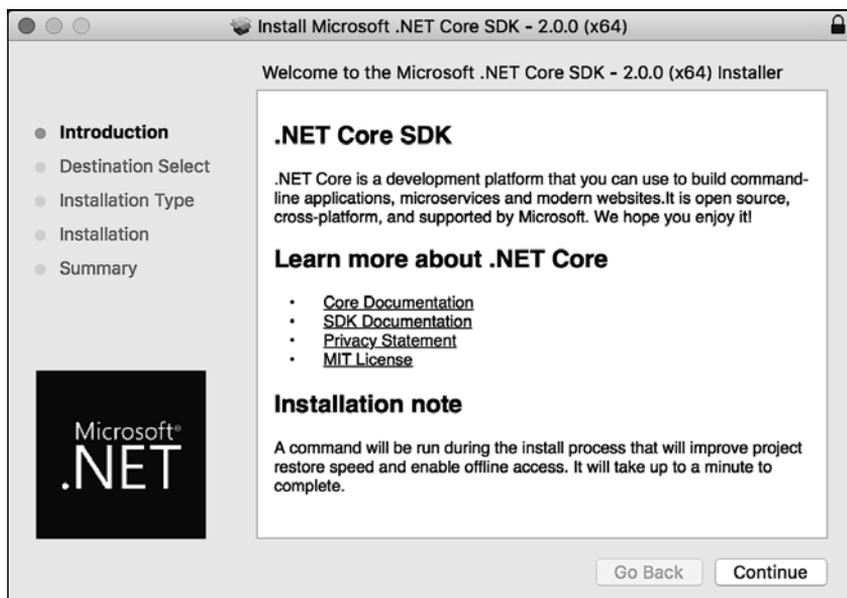


Рис. 1.6