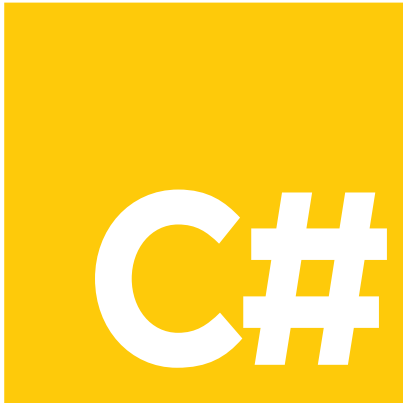




Майк МакГрат



- ЦВЕТНОЕ ИЗДАНИЕ
- ЧЕТКИЕ ПОШАГОВЫЕ ИНСТРУКЦИИ
- СОЗДАЙТЕ ПРИЛОЖЕНИЕ С НУЛЯ
- ОПЫТ В РАЗРАБОТКЕ НЕ НУЖЕН

С#

ДЛЯ НАЧИНАЮЩИХ



издание **3**

 **БОМБОРА**
ИЗДАТЕЛЬСТВО

Москва

УДК 004.43
ББК 32.973.2-018.1
М15

C# PROGRAMMING IN EASY STEPS, 3RD EDITION:
Modern coding with C# 10 and .NET 6. Updated for Visual Studio 2022
by Mike McGrath

Copyright © 2022 by In Easy Steps Limited.

Translated and reprinted under a licence agreement from the Publisher: In Easy Steps, 16 Hamilton
Terrace, Holly Walk, Leamington Spa, Warwickshire, U.K. CV32 4LY



МакГрат, Майк.

М15 С# для начинающих / Майк МакГрат ; [перевод с английского
К. А. Вантуха]. — Издание 3-е. — Москва : Эксмо, 2026. — 288 с. —
(Программирование для начинающих).

ISBN 978-5-04-213828-7

Цветное руководство по С# для начинающих позволит в короткое время освоить этот язык программирования. Следуйте понятным объяснениям и выполняйте интересные практические задания. Внутри есть вся необходимая информация, рассмотрены основы языка (синтаксис, переменные, ключевые слова, операторы и выражения) и подобраны наглядные цветные примеры, что позволит вам погрузиться в программирование даже при отсутствии опыта. Вы познакомитесь с основами работы в Visual Studio, разберетесь с ключевыми принципами объектно-ориентированного программирования, рассмотрите создание приложений для Windows и научитесь использовать платформу .NET. Превратите мечты в код!

УДК 004.43
ББК 32.973.2-018.1

ISBN 978-5-04-213828-7

© Вантух К.А., перевод на русский язык, 2025
© Оформление. ООО «Издательство «Эксмо», 2026

1

Введение

9

Введение в C#	10
Установка Visual Studio	13
Знакомство с IDE	16
Создание проекта Console	20
Запуск проекта Console	23
Пишем по правилам	26
Выводы	29

2

Хранение значений

31

Создание переменных	32
Чтение пользовательского ввода	36
Знакомство с массивами	39
Преобразование типов данных	42
Объявление констант	45
Выводы	49

3

Выполнение операций

51

Выполнение арифметических операций	52
Присвоение значений	55
Сравнение значений	58
Логические операции	61
Проверка условий	64
Приоритет операторов	67
Выводы	70

4**Составление операторов****73**

Ветвление при помощи if	74
Ветвление с помощью switch	77
Цикл for	80
Цикл while	83
Цикл for each	86
Выводы	89

5**Написание методов****91**

Создание функции	92
Передача аргументов	95
Перегрузка методов	98
Рефакторинг кода	101
Выводы	104

6**Работа со строками****107**

Общее представление о возможностях работы со строками	108
Изменение строк	111
Сцепление и сравнение строк	114
Копирование и замена	117
Поиск подстрок	120
Форматирование строк	123
Форматирование строк, содержащих дату	126
Выводы	129

7**Доступ к файлам****131**

Запись в файл	132
Добавление в файл	135
Чтение текста целиком и строк по отдельности	138
Потоковый вывод строк	141
Изменение ввода и вывода	144
Выводы	147

8**Решение проблем****149**

Распознавание ошибок реального времени	150
Исправление ошибок компиляции	153
Отладка кода	156
Создание точек останова	159
Перехват ошибок времени выполнения	162
Получение справки	165
Выводы	168

9**Создание объектов****171**

Инкапсуляция данных	172
Создание нескольких объектов	175
Инициализация объектов класса	178
Наследование свойств класса	181
Вызов конструкторов базового класса	184
Скрытие методов базового класса	187
Управление вызовом методов	190
Создание классов возможностей	193
Использование неполных классов	196
Выводы	199

10**Управление событиями****201**

Создание проекта Forms	202
Добавление визуальных средств управления	205
Написание функционального кода	208
Получение текстового ввода	211
Работа с полями опций	214
Работа с окном сообщений для пользователя	217
Вызов системных диалогов	220
Создание меню приложения	223
Связывание пунктов меню с обработчиками	226
Импорт аудиоресурсов	229
Выводы	232

11

Построение приложения

235

Планирование программы	236
Присвоение статических свойств	239
Проектирование внешнего вида	242
Присвоение динамических свойств	245
Добавление функции времени выполнения	248
Тестирование программы	251
Публикация приложения	254
Выводы	257

12

Работа с разными целевыми устройствами

259

Создание проекта Universal	260
Вставка компонентов страницы	263
Импорт активов программы	266
Проектирование внешнего вида	269
Добавление функции времени выполнения	272
Тестирование программы	275
Доработка интерфейса	278
Публикация приложения	281
Выводы	284

1 Введение

В этой главе вы познакомитесь с интегрированной средой разработки Visual Studio и узнаете, как создать настоящее приложение для Windows.

10	Введение в C#
13	Установка Visual Studio
16	Знакомство с IDE
20	Создание проекта Console
23	Запуск проекта Console
26	Пишем по правилам
29	Выводы

C#

Введение в C#

Объявив о создании фреймворка.NET на Конференции профессиональных разработчиков в 2000 году, компания Microsoft одновременно с ней представила и язык программирования под названием C# (произносится «си-шарп»). Идея названия была навеяна нотной записью, где знак диеза (#), или по-английски «шарп», означает повышение ноты на полтона. Сходным образом образовано название языка C++, поскольку оператор ++ означает приращение на единицу.

- C# спроектирован таким образом, чтобы быть простым, современным, универсальным, объектно-ориентированным языком программирования. При этом он заимствует ряд концепций из других языков, в первую очередь из Java. Таким образом, все в C# — это т. н. «объекты», принадлежащие классам и обладающие «свойствами» и «методами». Именно эти объекты и задействуются в программе.
- C# — элегантный типобезопасный язык, позволяющий создавать надежные и безопасные приложения. На нем можно писать клиентские приложения Windows, веб-службы XML, распределенные компоненты, приложения для баз данных и многое, многое другое.
- C# специально создан таким образом, чтобы использовать испытанную на практике функциональность, предоставляемую «библиотеками классов» фреймворка.NET. Таким образом, он должен быть установлен на компьютере, прежде чем на нем можно будет запускать приложения



на C#. Как правило, он уже входит в вашу систему в качестве одного из компонентов.

Фреймворк Microsoft.NET

Каждая версия Microsoft.NET включает в себя стандартный набор библиотек классов и виртуальную систему исполнения, которая называется CLR (Common Language Runtime, «общезыковая исполняющая среда»). CLR обеспечивает свободное взаимодействие между языком C# и библиотеками классов.

Чтобы создать исполняемую программу, исходный код на C# компилируется в код на языке Common Intermediate Language (IL), «высокоуровневый ассемблер». Этот текст сохраняется на диске и вместе с другими ресурсами программы, например изображениями, образует так называемую «сборку». Как правило, у файла сборки будет расширение .exe или .dll. Каждая сборка содержит «манифест», в котором приводится информация о ее требованиях в отношении безопасности.

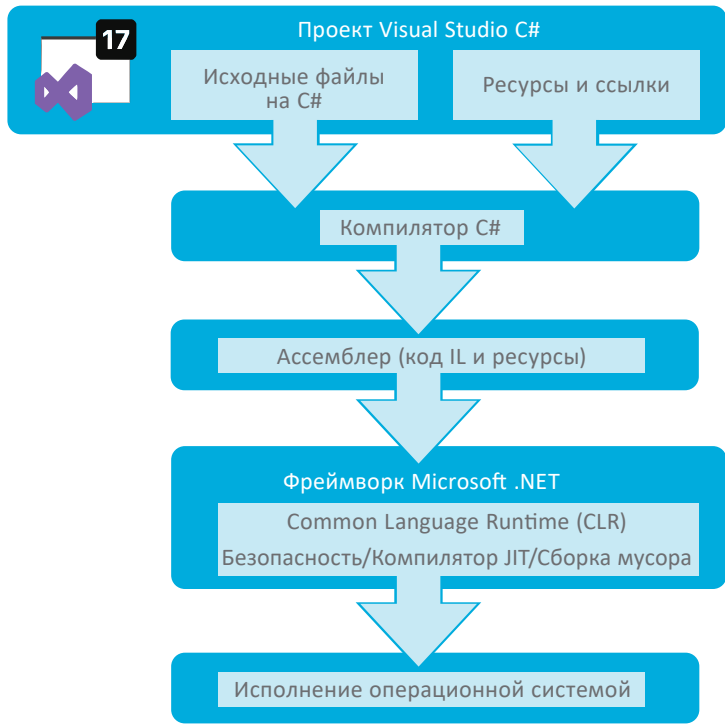
Во время исполнения программы на C# сборка загружается в CLR и проверяются требования безопасности, указанные в манифесте. Когда требования к безопасности удовлетворены, CLR производит JIT-компиляцию кода на IL в машинный код. Затем CLR выполняет задачи так называемой «сборки мусора», обработки исключений и управления ресурсами, и только после того передает программу для исполнения операционной системе:



JIT-компиляция также известна как «динамическая трансляция».



JIT-компиляция происходит во время исполнения программы, а не перед ним.



Поскольку возможность работы с разными языками программирования — важнейшее свойство Microsoft.NET., то код IL, сгенерированный компилятором C#, может взаимодействовать с кодом, сгенерированным NET-версиями компиляторов других языков, таких как Visual Basic или Visual C++. На протяжении всей книги примеры даются на языке Visual C#.

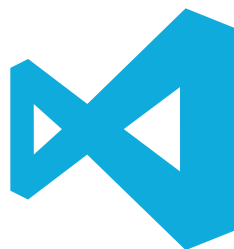
Установка Visual Studio

Чтобы создавать приложения на C#, начать следует с установки интегрированной среды разработки (Integrated Development Environment, IDE) Visual Studio.

Microsoft Visual Studio — профессиональный инструмент разработки, предоставляющий интегрированную среду для языков Visual Basic, Visual C++, Visual J# и Visual C#. В ее рамках кодировать приложения для Windows можно на таких языках, как Visual Basic, C++, J# и C#.

В том числе она существует в усеченном варианте Visual Studio Community, специально предназначенном для тех, кто изучает программирование. Ее пользовательский интерфейс упрощен, а некоторые возможности профессиональной версии исключены, чтобы не путать начинающих. Писать код Windows-приложений на C# можно в редакторе кода Code Editor любой версии Visual Studio.

Как Visual Studio, так и Visual Studio Community служат интегрированной средой разработки для C#, но в отличие от полнофункционального продукта, каким является Visual Studio, вариант Visual Studio Community полностью бесплатен и может быть установлен на любой компьютер, удовлетворяющий следующим требованиям:

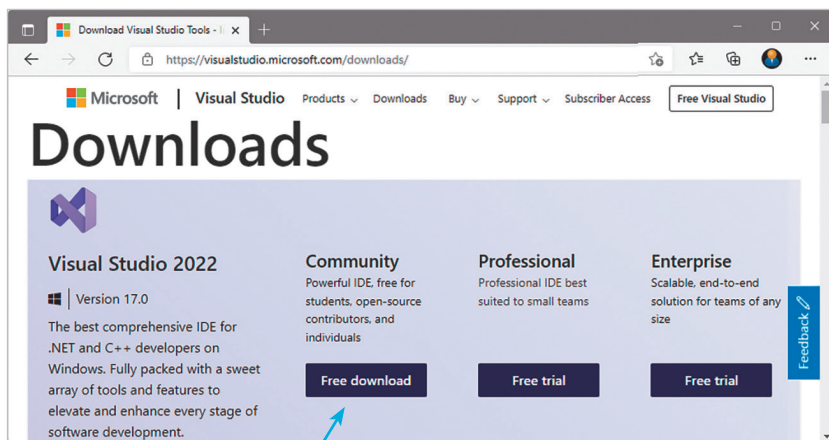



Если выбрать другое место установки, позже может понадобиться менять и другие пути в системе, поэтому проще согласиться с предложенным вариантом.

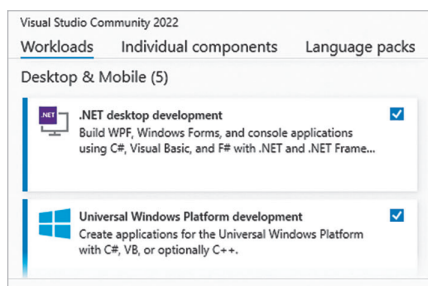
Компонент	Требование
Операционная система	Windows 11
	Windows 10 (версии 1909 или старше) Windows Server 2016 или 2019
	*Все вышеперечисленные операционные системы должны быть в 64-битной версии
Процессор	64-битный процессор с тактовой частотой от 1,8 ГГц
Оперативная память	4 Гб (рекомендуется 16 Гб)
Жесткий диск	До 210 Гб свободного места
Видеокарта	Минимальное разрешение 1366×768
	Оптимальное разрешение 1920×1080

В данной книге для демонстрации программирования на C# используется Visual Studio Community, но все примеры можно воспроизвести и в Visual Studio. Для установки Visual Studio Community следуйте перечисленным ниже шагам.

- 1 Откройте браузер и перейдите на страницу загрузки Visual Studio. На момент написания этой книги она находилась по адресу visualstudio.microsoft.com/downloads.



- 2 Нажмите кнопку под вариантом Community, чтобы скачать файл **VisualStudioSetup.exe**
- 3 Нажмите на значок , чтобы начать установку и запустить программу-установщик **Visual Studio Installer**.
- 4 Примите предложенное место установки, а затем нажмите **Next** (Дальше).
- 5 Поставьте галочки в двух полях **Installer**, относящихся к C#.



- 6 Нажмите кнопку **Install** в нижнем правом углу экрана установщика, чтобы таким образом начать процесс скачивания и установки.

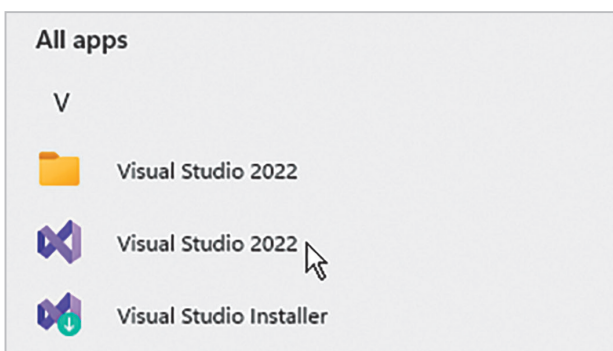


Как **Visual Studio**, так и **Visual Studio Installer** попадут в меню **All apps** вашей системы. Установщик всегда можно будет запустить повторно, чтобы добавить или удалить функции.



Знакомство с IDE

- 1 Перейдите в меню **All apps** (Все приложения) и выберите пункт Visual Studio 2022, который был добавлен туда установщиком.



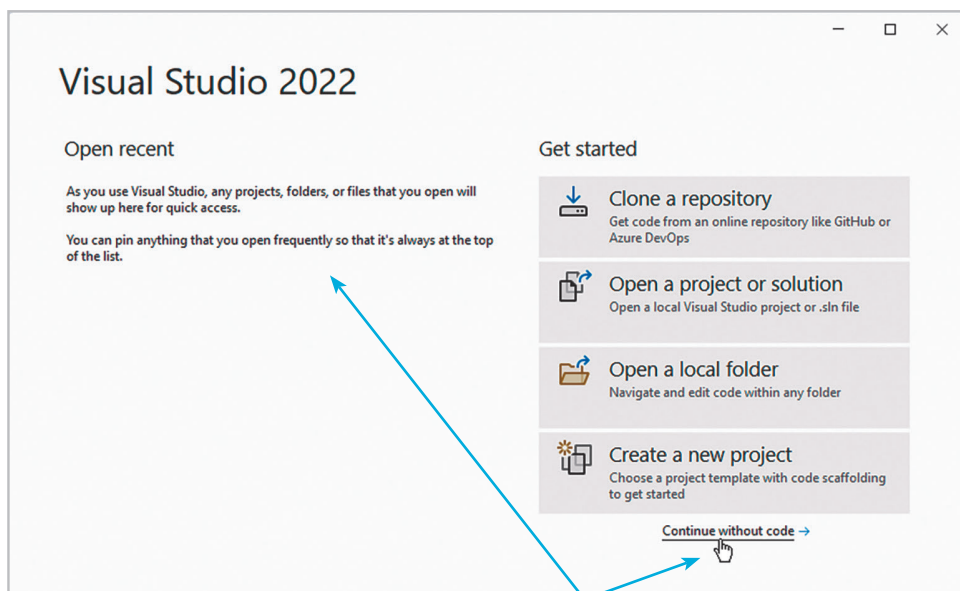
При первом запуске Visual Studio выполняет конфигурационные задачи, поэтому придется подождать несколько минут.

- 2 Войдите со своей учетной записью Microsoft или создайте ее, а потом войдите. Это необходимо для продолжения работы.

- 3 Посмотрите на появившуюся форму **Start Page** (Начальная страница), где будут перечислены проекты, с которыми вы работали в последнее время, а также несколько вариантов действий в разделе «Get Started».



Появляется интегрированная среда разработки Visual Studio, в которой у вас сразу есть доступ ко всему необходимому для создания полноценных приложений Windows: из этой точки можно создавать изящные пользовательские интерфейсы, вводить код, компилировать и исполнять приложения, отлаживать ошибки и многое другое.



4

Пока что просто щелкните по ссылке **Continue without code** (Продолжить без кода), и запустите таким образом приложение Visual Studio.



В будущем здесь будут появляться проекты, над которыми вы недавно работали, поэтому открывать их будет удобно.