МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра «Интеллектуальные и информационные технологии»

**РЕФЕРАТ**

**По дисциплине**

**ОАСОИ**

на тему: «Язык программирования C#»

Выполнил студент 1 курса

Группы АС-59

Левоцкий Н.Д.

Проверил

Анфилец С.В.

Брест 2020

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Содержание** |  |
| Введение | | | 3 |
| 1. | История развития | | 4 |
| 1.1 | | Предки С# | 4 |
| 1.2 | | Создание .NET и C# | 5 |
| 2. | Особенности языка С# | | 6 |
| 3. | Плюсы и минусы языка | | 7 |
| 4. | | Платформа .NET | 8 |
| 5. | | Основные технологии платформы .NET | 12 |
| 6. | | Пример программы на С# | 15 |
| Заключение | | |  |
| Список используемых источников | | |  |

**Введение**

Что же вообще такое язык программирования?

Язык программирования – это набор правил, которые определяют, как написанная компьютерная программа выглядит и что компьютер может сделать под ее управлением. Программа представляет собой код, написанный по правилам конкретного языка программирования. Язык программирования – это средство общения человека и компьютера. При этом код на языке программирования пишется так, чтобы он был понятен человеку. Один из критериев хорошего кода – программист читает его и понимает, что это, для чего нужно и как будет работать.

Ядро из двух десятков наиболее популярных языков состоит из: Java, C, Python, C++, Visual Basic .NET, JavaScript, C#, PHP, SQL, Objective-C, Assembly language, MATLAB, Perl, Delphi/Object Pascal, R, Ruby, Visual Basic, Go, Groovy, Swift. Есть еще 2-3 десятка довольно широко используемых языков, таких как Scala, Lua, Fortran, COBOL, Lisp, Prolog, Rust, Kotlin, Eglang, Haskell и др. Со временем разработчик знает и сочетает несколько языков для разных задач и платформ.

Каждый язык программирования используется для своей конкретной задачи: какие-то широко используемы для веб разработки, какие-то в создании программ для операционных систем, а какие-то для создания приложений для мобильных систем, например Android, iOS и т.д.

Созданная компанией Microsoft платформа .NET содержит в себе множество технологий, позволяющих выполнять подобные задачи, а основным языком программирования данной платформы является C#.

Сторонники C# относят его к самым продвинутым, универсальным, мультипарадигменным и удобным в применении языкам программирования. Принимая во внимание тот факт, что за этим языком стоит платформа Microsoft.NET, количество таких сторонников достаточно большое

3

1. **История развития**

**1.1 Предки С#**

Еще в 60-х годах появились далекие предки C#. А началось всё с появлением языка B, созданного в 1969 году коллективом исследователей из Технологического института Массачусетса (MIT). Кен Томпсон является главным автором B. В то время команда разрабатывала операционную систему UNIX. Язык PL/I, уже существовавший и применявшийся тогда для мэйнфреймов, изготовленных компанией IBM, меньше подходил для решения поставленной задачи, кроме того был достаточно громоздким. Исходя из этого, ученые решили поработать над созданием нового языка, получившего название B. Это типичный представитель ранних императивных языков программирования.

Как это ни странно, последователем B стал изобретенный в 1972 году язык программирования C. В его основу положен сам B.

Язык C создали Денис Ритчи и Кен Томпсон, работники исследовательской лаборатории компании под названием AT&T (AT&T Bell Telephone Laboratories). Над созданием расширенной версии B Ритчи начал свою работу в 1971 году. Сначала разработчик назвал её NB (New B), но после того как язык получил большие отличия от B, название заменили на C. B получил расширение за счет явного применения структур, типов и ряда новых операций.

Бьёрн Страуструп (Bell Labs) в 1984 году выступил с демонстрацией проекта языка С++. В период занятий исследованиями в фирме, Страуструп написал несколько имитационных программ, которые требовались для моделирования распределенных вычислений.

В это время был создан С++, - язык программирования, которому первоначально дали название С with classes («Си с классами»). А придумал название «С++» Рик Мэсчитти. "++" является оператором инкремента в С, словно намекающий на то обстоятельство, что язык С++, что-то большее, чем просто С.

4

**1.2 Создание .NET и C#**

Начало нового тысячелетия Microsoft решила отметить выпуском новоявленных программных продуктов. К 2000-му году компанией были подготовлены промышленные модификации новых решений и компонентных технологий в сфере обмена сообщениями и информацией, а также изготовления Internet-приложений (ASP+, COM+, SOAP, ADO+, Biztalk Framework). С целью поддержки этих новшеств компания Microsoft выпустила платформу .NET - инструментарий для разработки приложений. Эта платформа совместила «под общей крышей» несколько языков программирования, что для того времени было в новинку.

Технология активных серверных страниц (Active Server Page) ASP.NET также стала ещё одним нововведением платформы. Посредством неё можно было довольно быстро разработать взаимодействующие с базами данных веб-приложения.

C# - язык программирования созданный специально для ASP.NET. Следует отметить, что и сама ASP.NET также была полностью написана на этом языке.

Название «Си шарп» (в пер. с англ. sharp — диез) заключает в себе «сакральный» смысл. В музыкальной нотации знак «#» читается как «диез» и обозначает повышение на полтона высоты звука. Можно рассмотреть образование названия «C#» и с другой стороны, как производное следующей череды трансформаций: C → C++ → C++++(C#), поскольку символ «#» является совокупностью 4-х знаков «+».

Авторы этого языка программирования – Вилтамут Скотт и Хейльсберг Андерс — создатель Дельфи и Турбо Паскаля, в 1996 году перешедший в Microsoft.

Языком C# поддерживаются все три основных составляющих объектно-ориентированного программирования: полиморфизм, наследование и инкапсуляция. Кроме того, в нем реализован замысел автоматической «сборки мусора», динамического связывания и обработки исключений.

5

Началом работы над C# стал декабрь 1998 года. COOL – вот такое закодированное название дали данному проекту (C-style Object Oriented Language).

Первую бета-версию C# 1.0 выпустили летом 2000 года, а окончательную версию языка совместно с Microsoft Visual Studio мир увидел в феврале 2002. Так как в C# сочетаются лучшие стороны предыдущих популярных языков программирования, таких как Java, C и C++, программисты с легкостью могут переходить на C#, руководствуясь знаниями любого из указанных языков. Крайней на данный момент версией языка является 8.0.

**2. Особенности языка С#**

Си Шарп – действительно интересный инструмент, достойный внимания. Он уверенно занимает высокие позиции в рейтингах востребованных ЯП на рынке труда. Потому имеет смысл изучить его возможности подробнее и понять, для чего и где стоит применять C#. На нем пишут практически все, что угодно, от небольших веб-приложений до мощных программных систем, объединяющих в себе веб-структуры, приложения для десктопов и мобильных устройств. Все это стало возможным благодаря удобному Си-подобному синтаксису, строгому структурированию, огромному количеству фреймворков и библиотек (их число достигает нескольких сотен).

Компания Microsoft уделяет значительное внимание поддержке языка разработки, а потому регулярно появляются обновления и дополнения, исправляются выявленные баги в компиляторе, расширяются библиотеки. Разработчики заинтересованы в популяризации инструмента и прилагают к этому массу усилий

Перечень возможностей разработки практически не имеет ограничений благодаря широчайшему набору инструментов и средств. Конечно, все это можно реализовать при помощи других языков, но некоторых из них узкоспециализированные, в других придется использовать дополнительные инструменты сторонних разработчиков.

6

Инструментарий C# позволяет решать широкий круг задач, язык действительно очень мощный и универсальный. На нем разрабатывают:

* Приложения для WEB.
* Различные игровые программы.
* Приложения платформ Андроид или iOS.
* Программы для Windows

В Си Шарп присутствует, т.н. **сборщик мусора**. Он позволяет в автоматическом режиме очистить память от объектов, которые не используются. В Си и С++ такого инструмента нет и приходится очищать память вручную.

В языке принята общая система работы с типами, начиная от примитивов и заканчивая сложными, в том числе, пользовательскими наборами. Применяется единый набор операций для обработки и хранения значений типизации. Также можно использовать ссылочные типы пользователя, что позволит динамически выделить память под объект или хранить упрощенную структуру в сети.

Язык программирования запрещает обращение к переменным, которые не были инициализированы, что исключает возможность выполнения бесконтрольного приведения типов или выхода за пределы определенного массива данных.

7

**3. Плюсы и минусы языка**

**Плюсы:**

1. Поддержка подавляющего большинства продуктов Microsoft.
2. Бесплатность ряда инструментов для небольших компаний и некоторых индивидуальных разработчиков: Visual Studio, облако Azure, Windows Server, Parallels Desktop для Mac Pro, VS Code и др.
3. Типы данных имеют фиксированный размер (32-битный int и 64-битный long), что повышает “мобильность” языка и упрощает программирование, так как вы всегда знаете точно, с чем вы имеете дело.
4. Большое количество “синтаксического сахара” – специальных конструкций, разработанных для понимая и написания кода. Они не имеют значения при компиляции.
5. Низкий порог вхождения. Синтаксис C# имеет много схожего с другими языками программирования, благодаря чему облегчается переход для программистов. Язык C# часто признают наиболее понятным и подходящим для новичков
6. С помощью Xamarin на C# можно писать программы и приложения для мобильных систем (Android, iOS) и таких систем как MacOS и Linux.
7. Количество вакансий достаточно велико.

**Минусы:**

1. Приоритетная ориентированность на платформу Windows
2. Язык бесплатен только для небольших фирм, индивидуальных программистов, стартапов и учащихся. Крупной компании покупка лицензии обойдётся в круглую сумму.

8

1. **Платформа .NET**

Программа на языке С# выполняется в среде .NET Framework. Когда говорят C#, нередко имеют в виду технологии платформы .NET (Windows Forms, WPF, ASP.NET, Xamarin). И, наоборот, когда говорят .NET, нередко имеют в виду C#. Однако, хотя эти понятия связаны, отождествлять их неверно. Язык C# был создан специально для работы с фреймворком .NET, однако само понятие .NET несколько шире.

Как-то Билл Гейтс сказал, что платформа .NET - это лучшее, что создала компания Microsoft. Возможно, он был прав. Фреймворк .NET представляет мощную платформу для создания пр иложений. Можно выделить следующие ее основные черты:

* **Поддержка нескольких языков**. Основой платформы является общеязыковая среда исполнения Common Language Runtime (CLR), благодаря чему .NET поддерживает несколько языков: наряду с C# это также VB.NET, C++, F#, а также различные диалекты других языков, привязанные к .NET, например, Delphi.NET. При компиляции код на любом из этих языков компилируется в сборку на общем языке CIL (Common Intermediate Language) - своего рода ассемблер платформы .NET. Поэтому при определенных условиях мы можем сделать отдельные модули одного приложения на отдельных языках.
* **Кроссплатформенность**. .NET является переносимой платформой (с некоторыми ограничениями). Например, последняя версия платформы на данный момент - .NET 5 поддерживается на большинстве современных ОС Windows, MacOS, Linux.
* **Мощная библиотека классов**. .NET представляет единую для всех поддерживаемых языков библиотеку классов. И какое бы приложение мы не собирались писать на C# - текстовый редактор, чат или сложный веб-сайт - так или иначе мы задействуем библиотеку классов .NET.
* **Разнообразие технологий**. Общеязыковая среда исполнения CLR и базовая библиотека классов являются основой для целого стека технологий, которые разработчики могут задействовать при построении тех или иных приложений. Например, для работы с базами данных в этом стеке технологий предназначена технология ADO.NET и Entity Framework Core. Для построения графических приложений с богатым насыщенным интерфейсом - технология WPF и UWP, для создания более простых графических приложений - Windows Forms. Для разработки мобильных приложений - Xamarin. Для создания веб-сайтов и веб-приложений - ASP.NET и т.д.

К этому стоит добавить активной развивающийся и набирающий популяность Blazor - фреймворк, который работает поверх .NET и который позволяет создавать веб-приложения как на стороне сервера, так и на стороне клиента. А в будущем будет поддерживать создание мобильных приложений и, возможно, десктоп-приложений.

* **Производительность**. Согласно ряду тестов веб-приложения на .NET 5 в ряде категорий сильно опережают веб-приложения, построенные с помощью других технологий. Приложения на .NET 5 в принципе отличаются высокой производительностью.

**Как это всё работает?**

Вспомним, что такое процесс компиляции – это перевод вашего кода, понятного человеку, в бинарный код, понятный компьютеру.  
В программировании на .NET компиляция и запуск приложений происходит следующим образом:  
Код из любого языка преобразовывается в код, написанный на общем языке (Common intermediate language или CIL). Этот язык является языком низшего уровня, похожего по синтаксису на язык ассемблер.  
После, этот код передаётся так называемой исполняющей среде (Common language runtime или CLR), которая берёт функции и методы из .NET Framework

После этого конечный результат передаётся на процессор и выполняется программа.

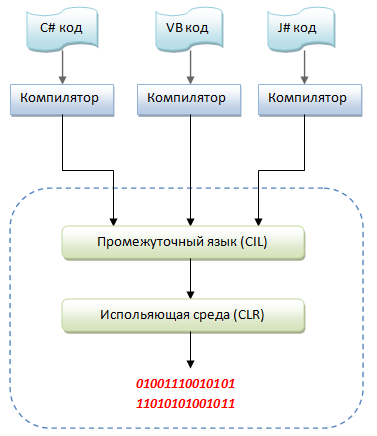


Рис 1. Компиляция и запуск приложения на .NET

CLR – это некая «виртуальная машина», которая собственно и управляет нашими приложениями, написанными для .NET. Такая схема сборки приложений очень удобна и тем, что происходит «компиляция на лету». То есть не компилируя программу, среда разработки может указать Вам на Ваши ошибки, а это заметно ускоряет процесс разработки.

**Что же выделяет .NET программирование среди всех?**

Первое – это конечно-же относительная лёгкость освоения и работы с языками, так как большинство .Net языков – языки высшего уровня.  
Быстрота выполнения конечных исполняемых файлов.  
Приложения, написанные на .net сами чистят себя в оперативной памяти, благодаря автоматическому сборщику мусора.  
Приложение нужно «собрать» всего один раз, и оно будет работать на всех платформах процессоров и операционных системах семейства Windows. В это же время приложение будет показывать весь свой потенциал скорости, который возможен на определённых процессорах.

1. **Основные технологии .NET Framework**

**ASP.NET(**Active Server Pages для .NET) — платформа разработки [веб-приложений](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5), в состав которой входит: [веб-сервисы](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B5%D1%80%D0%B2%D0%B8%D1%81), программная инфраструктура, модель программирования, от компании [Майкрософт](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D0%B9%D0%BA%D1%80%D0%BE%D1%81%D0%BE%D1%84%D1%82). ASP.NET входит в состав платформы [**.NET Framework**](https://ru.wikipedia.org/wiki/Microsoft_.NET)[[2]](https://ru.wikipedia.org/wiki/ASP.NET#cite_note-2) и является развитием более старой технологии [Microsoft ASP](https://ru.wikipedia.org/wiki/Active_Server_Pages" \o "Active Server Pages).

ASP.NET внешне во многом сохраняет схожесть с более старой технологией ASP, что позволяет разработчикам относительно легко перейти на ASP.NET. В то же время внутреннее устройство ASP.NET существенно отличается от ASP, поскольку она основана на платформе [.NET](https://ru.wikipedia.org/wiki/Microsoft_.NET) и, следовательно, использует все новые возможности, предоставляемые этой платформой.

На данный момент является одной из ведущих технологий для крупной веб-разработки.

**Windows Forms** — [интерфейс программирования приложений](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F_%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B9) (API), отвечающий за [графический интерфейс пользователя](https://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8F) и являющийся частью [Microsoft](https://ru.wikipedia.org/wiki/Microsoft" \o "Microsoft) [.NET Framework](https://ru.wikipedia.org/wiki/.NET_Framework). Данный интерфейс упрощает доступ к элементам интерфейса [Microsoft Windows](https://ru.wikipedia.org/wiki/Microsoft_Windows" \o "Microsoft Windows) за счет создания обёртки для существующего [Win32 API](https://ru.wikipedia.org/wiki/Windows_API) в [управляемом коде](https://ru.wikipedia.org/wiki/%D0%A3%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D1%8F%D0%B5%D0%BC%D1%8B%D0%B9_%D0%BA%D0%BE%D0%B4). Причём управляемый код — классы, реализующие [API](https://ru.wikipedia.org/wiki/API) для Windows Forms, не зависят от языка разработки. То есть программист одинаково может использовать Windows Forms как при написании ПО на C#, С++, так и на VB.Net, J# и др.

**WPF (Windows Presentation Foundation) -** аналог WinForms, система для построения клиентских приложений Windows с визуально привлекательными возможностями взаимодействия с пользователем, графическая (презентационная) подсистема в составе . NET Framework (начиная с версии 3.0), использующая язык XAML.

**Преимущества WPF**

* Она более новая и, соответственно, отвечает современным стандартам разработки
* Microsoft использует ее во многих своих приложениях, например Visual Studio
* Это более гибкая система, Вы можете сделать больше, без написания или покупки готовых элементов управления
* Если вдруг Вы решите воспользоваться готовыми решениями из сообщества, скорее всего сторонние разработчики будут сфокусированы именно на WPF, поскольку это более новая система
* С помощью XAML можно легко создавать и редактировать GUI, позволяя разделить работу дизайнера (XAML) и программиста (C#, VB.NET и др.)
* Привязка данных позволяет Вам еще лучше разделить данные и GUI
* Для лучшей производительности, можно использовать аппаратное ускорение при отрисовке GUI
* Позволяет создавать GUI как для Windows приложений, так и для приложений Web (Silverlight/XBAP)

**Преимущества WinForms**

* Технология более старая и, соответственно, лучше испытанная и протестированная
* На данный момент существует огромное множество готовых элементов управления, которые можно купить либо использовать бесплатно
* С точки зрения написания, дизайнер Visual Studio лучше приспособлен к WinForms, так как в WPF больше необходимо делать самому

**Xamarin** — это фреймворк для кроссплатформенной разработки мобильных приложений (iOS, Android, Windows Phone) с использованием языка C#. Идея очень простая. Вы пишете код на своем любимом языке, с применением всех привычных для вас языковых фич типо LINQ, лямбда-выражений, Generic`ов и async`ов. При этом вы имеете полный доступ ко всем возможностям SDK платформы и родному механизму создания UI, получая на выходе приложение, которое, строго говоря, ничем не отличается от нативных и (по крайней мере по заверениям) не уступает им в производительности.  
  
Фреймворк состоит из нескольких основных частей:

* Xamarin.IOS — библиотека классов для C#, предоставляющая разработчику доступ к iOS SDK;
* Xamarin.Android — библиотека классов для C#, предоставляющая разработчику доступ к Android SDK;
* Компиляторы для iOS и Android;
* IDE Xamarin Studio;
* Плагин для Visual Studio.

**Entity Framework** представляет специальную объектно-ориентированную технологию на базе фреймворка .NET для работы с данными. Если традиционные средства ADO.NET позволяют создавать подключения, команды и прочие объекты для взаимодействия с базами данных, то Entity Framework представляет собой более высокий уровень абстракции, который позволяет абстрагироваться от самой базы данных и работать с данными независимо от типа хранилища. Если на физическом уровне мы оперируем таблицами, индексами, первичными и внешними ключами, но на концептуальном уровне, который нам предлагает Entity Framework, мы уже работает с объектами.

1. **Пример программы на C#**

Данная программа будет выводить на консоль n-й член последовательности чисел Фибоначчи. N вводится пользователем.

Листинг - Программа « Вывод n-го числа последовательности Фибоначчи»

using System;

namespace usernsp

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Введите номер числа последовательности Фибоначчи: ");

int Border = Convert.ToInt32(Console.ReadLine());

Console.WriteLine($"{Program.fibonacci(Border)}");

}

static int fibonacci(int N)

{

if (N==0)

{

return 0;

}

else if(N==1)

{

return 1;

}

else

{

return fibonacci(N - 1) + fibonacci(N - 2);

}

}

}

}

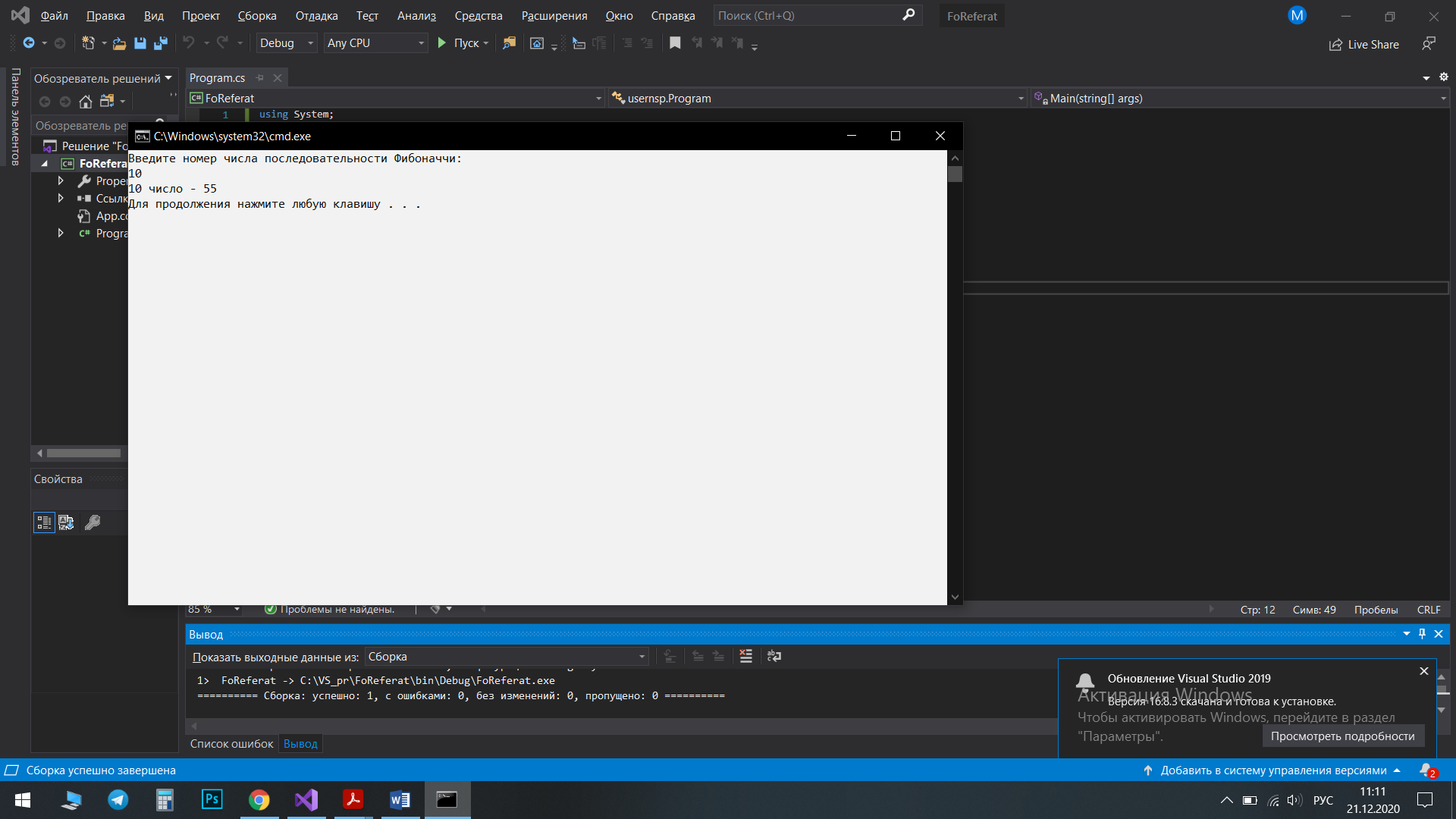


Рис. 2 Результат программы и вывод на консоль

**Заключение**

C# на протяжении долгого времени уверенно лидирует в рейтинге лучших и наиболее востребованных на рынке разработки языков. Сначала им заинтересовались только разработчики, которые пишут приложения под Windows. Но в процессе развития C# «научился» работать на Mac, Linux, IoS и Android. А после того, как код платформы открыли для всех желающих, были сняты практически все возможные ограничения в применении C#. В результате язык активно развивается, применяется все шире. Рекомендован к изучению в качестве одного из базовых для разработчиков любого профиля.

17

**Список используемых источников**

# 1. Язык программирования C#: история от создания до наших дней. URL:

https://appsstudio.ru/blog/application/yazyk-programmirovaniya-c-istoriya-ot-sozdaniya-do-nashih-dney. Дата доступа: 01.12.2020

# 2. Язык программирования C#: краткий обзор URL:

https://techrocks.ru/2019/02/16/c-sharp-programming-language-overview/

Дата доступа: 05.12.2020

# 3. Что такое среда .NET и как она работает? URL: https://habr.com/ru/sandbox/13849/ Дата доступа: 06.12.2020

4. Технологии .NET Framework URL: https://metanit.com/sharp/

Дата доступа: 12.12.2020