**.NET Framework**

Предназначена для разработки и запуска приложений под Windows. Включает инструменты для разработки, языки программирования и библиотеки, которые используются для создания настольных и веб-приложений.  
**.NET Core**

Кроссплатформенная среда выполнения для создания различных типов приложений — от веб-сервисов до мобильных решений.

Основное отличие между .NET Framework и .NET Core заключается в ориентации на разные платформы. .NET Framework в первую очередь ориентирован на Windows, а .NET Core поддерживает несколько платформ — Windows, macOS и Linux

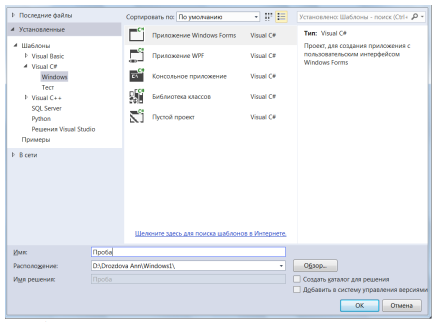
**Visual Studio 2019** — интегрированная среда разработки от Microsoft, которая включает инструменты для разработки, облачные сервисы и расширения. С её помощью можно создавать приложения и игры для настольных компьютеров, интернета, Windows Store, Android и iOS.

**особенности Visual Studio 2019:**

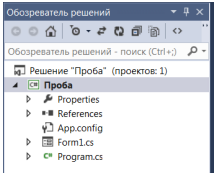
* Улучшенный поиск. Результаты поиска отображаются динамически при вводе. Новая логика нечёткого поиска находит всё необходимое, независимо от опечаток.
* Новый экран приветствия. Также в этой версии появился новый интерфейс для создания проектов и строка заголовка с меню приложения и переработанным поиском.
* Система Live Share. Она работает и в Visual Studio, и в Visual Studio Code, позволяя разработчикам кодить и отлаживать вместе.
* Поддержка различных рантаймов Python. Это позволяет легче переключаться между интерпретаторами и версиями.
* Интеграция с GitHub. В 2019 году появилась поддержка модели пулл-реквестов для управления интеграцией патчей в кодовую базу прямо в IDE.
* Отладчик моментальных снимков. С его помощью можно получить моментальный снимок выполнения приложения в облаке. (Эта функция доступна только в Visual Studio Enterprise).

**Пример создания программы на windows forms:**

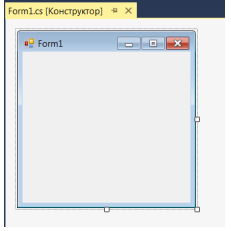
1. Запустите среду программирования VisualStudio. На начальной странице выберите пункт Создать проект. В разделе шаблонов выберите Приложение Windows Forms.



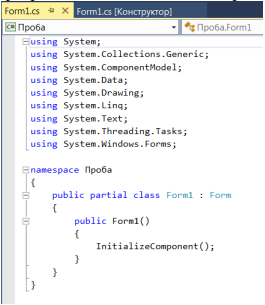
1. В результате будет создана заготовка нового проекта. В окне Обозреватель решений виден состав нашего проекта. Он не пуст – в папках уже имеются компоненты. Именно те, которые нужны для организации Windows-приложения. Каждое решение в C# может содержать несколько проектов. В нашем случае – это один проект под названием Проба. В составе проекта есть два программных файла – Program.cs (такой файл вы уже видели в консольном приложении) и Form1.cs.



1. На экране слева вы видите отображение содержимого файла Form1.cs. Обратите внимание на закладку, соответствующую этому отображению. Закладка помечена, словом, [Конструктор]. Это не случайно. Файл отображается визуально. Показано, какие объекты в нем используются. Правда, пока в нём нет никаких объектов, не считая самого окна, но само оно тоже объект.

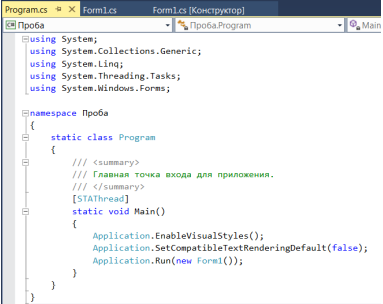


1. Установите мышку на окно и нажмите правую кнопку. Отобразится контекстное меню, связанное с окном. Выберите режим «Перейти к коду». Появится новая закладка с именем Form1, но без пометки Конструктор. Отображено содержимое файла Form1.cs, но уже не в виде объектов, а в виде программного кода на языке C#. Операторов, подключающих к программе системные пространства имён, здесь значительно больше, чем в консольном приложении. Есть уже известное нам пространство имен System. Другие строки подключают подпространства имен пространства System. Есть среди них, в частности, и пространство System.Windows.Forms, которое содержит всё необходимое для разработки окон Windows-приложения.

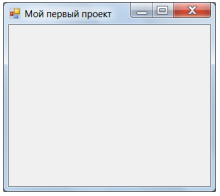


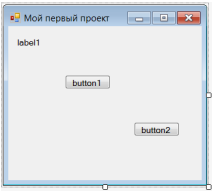
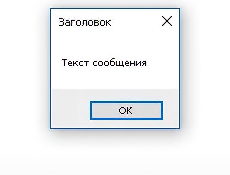
Имеется пространство имен нашей программы – Проба. Внутри него имеется описание класса с именем Form1. Класс – это структурная единица программы, ее отдельный модуль, где описываются какие-либо алгоритмы. Описания алгоритмов оформлены в виде функций (методов). В составе класса Form1 уже имеется метод Form1(). Это особый метод: его называют конструктором объектов класса. Конструктор объектов есть в любом классе. В данном случае – это конструктор окна Form1. Кроме конструктора в составе класса будут и другие методы, которые напишет программист. Все эти методы будут иметь отношение именно к окну Form1.

1. Рассмотрите модуль Program.cs. Выберите его мышкой в окне Обозреватель решения. Структура этого модуля нами уже изучена. В этом модуле есть класс – Program. Он содержит метод Main(). Вы помните, что это главный метод программы. Однако, в отличие от консольного приложения здесь этот метод не пуст – в нем уже есть три оператора. Если последовательно навести указатель мышки на каждый оператор, то можно получить сведения о назначении каждого оператора. В целом смысл операторов таков. Два первых оператора разрешают использовать визуальные стили в процессе выполнения программы и задают некоторые стандартные режимы управления программой. Третий оператор обеспечивает запуск процесса, который связан с окном Form1. Для этого используется метод Run из класса Application. В круглых скобках указано имя того объекта, с которым связан процесс.



1. С помощью закладки Form1.cs [Конструктор] переключитесь в окно визуального отображения формы Form1. Установите курсор мышки на окне, вызовите контекстное меню и выберите Свойства. Отобразится дополнительное окно свойств формы. Свойств много, они различны, знакомиться с ними нужно постепенно, по мере надобности. Для начала отметим свойство Name, которое определяет идентификатор окна. Фактически это имя переменной: Form1. Именно это имя будет использоваться в программном коде при различных обращениях к объекту окно Form1.
2. Свойство Text содержит заголовок окна – сейчас заголовок совпадает с именем переменной. Это свойство можно изменить, что приведёт к изменению названия окна. Измените значение данного свойства на Мой первый проект. При этом на форме изменится заголовок окна.
3. Откомпилируйте приложение, выбрав из главного меню команду Сборка – Собрать решение.
4. Для запуска приложения выберите из главного меню команду Отладка – Начать отладку (F5). Приложение запустится в отладочном режиме и на экране появится разработанное окно.



1. Для закрытия окна щелкните мышью на кнопке закрытия.
2. В окне компонентов(cntr + alt + x) найдите элемент кнопки и перетащите на форму:  
    
3. Выделите кнопку button, перейдите в окно Свойства, найдите свойство Text и измените его на Выход. Найдите свойство Name и измените его на btnExit.(стандарт microsoft устанавливает ‘btn...’ как наименование для компонента кнопка). Выберети свойство color и измените цвет кнопки
4. Нажав двойной клик по компоненту вы поподёте в функцию отработки элемента: 
5. Используйте команду MessageBox.Show(“Текст сообщения”); для вывода текста в бокс  
   
6. Используйте команду Application.Exit(); для выхода из программы (завершение текущего потока)
7. Если вы запустите проект(ctrl + f5) и нажмёте на кнопку, то в зависимости от введённой команды произойдёт обработка этой кнопки.
8. После проделанной работы не забывайте завершить программу.