Выполнение программы

Весь код программы на языке C# помещается в файлы с расширением **.cs**. По умолчанию в проекте, который создается в Visual Studio (а также при использовании .NET CLI) уже есть один файл с кодом C# - файл **Program.cs** со следующим содержимым:

```
1   // See https://aka.ms/new-console-template for more information
2   Console.WriteLine("Hello, World!");
```

Именно код файла **Program.cs** выполняется по умолчанию, если мы запустим проект на выполнение. Но при необходимости мы также можем добавлять другие файлы с кодом C#.

Инструкции

Базовым строительным блоком программы являются **инструкции** (statement). Инструкция представляет некоторое действие, например, арифметическую операцию, вызов метода, объявление переменной и присвоение ей значения. В конце каждой инструкции в C# ставится точка с запятой (;). Данный знак указывает компилятору на конец инструкции. Например, в проекте консольного приложения, который создается по умолчанию, есть такая строка:

```
1   Console.WriteLine("Hello, World!");
```

Данная строка представляет вызов метода `Console.WriteLine`, который выводит на консоль строку. В данном случае вызов метода является инструкцией и поэтому завершается точкой с запятой.

Набор инструкций может объединяться в блок кода. Блок кода заключается в фигурные скобки, а инструкции помещаются между открывающей и закрывающей фигурными скобками. Например, изменим код файла **Program.cs** на следующий:

```
1   {
2       Console.WriteLine("Привет");
3       Console.WriteLine("Добро пожаловать в C#");
4   }
```

Здесь блок кода содержит две инструкции. И при выполении этого кода, консоль выведет две строки

```
Привет
Добро пожаловать в C#
```

В данном блоке кода две инструкции, которые выводят на консоль определенную строку.

Одни блоки кода могут содержать другие блоки:

```
{
    Console.WriteLine("Первый блок");
    {
        Console.WriteLine("Второй блок");
    }
}
```

Регистрозависимость

C# является регистрозависимым языком. Это значит, что в зависимости от регистра символов какие-то определенные названия могут представлять разные классы, методы, переменные и т.д. Например, для вывода на консоль используется метод **WriteLine** - его имя начинается именно с большой буквы: "WriteLine". Если мы вместо "Console.WriteLine" напишем "Console.writeline", то программа не скомпилируется, так как данный метод обязательно должен называться "WriteLine", а не "writeline" или "WRITELINE" или как-то иначе.

Комментарии

Важной частью программного кода являются комментарии. Они не являются собственно частью программы, при компиляции они игнорируются. Тем не менее комментарии делают код программы более понятным, помогая понять те или иные его части.

есть два типа комментариев: однострочный и многострочный. Однострочный комментарий размещается на одной строке после двойного слеша //. А многострочный комментарий заключается между символами /* текст комментария */. Он может размещаться на нескольких строках. Например:

```
/*
        первая программа на C#,
        которая выводит приветствие на консоль
*/
Console.WriteLine("Привет");                // Выводим строку "Привет"
Console.WriteLine("Добро пожаловать в C#"); // Выводим строку "Добро пожаловать в
```

Файл проекта

В каждом проекте проекте C# есть файл, который отвечает за общую конфгурацию проекта. По умолчанию этот файл называется **Название_проекта.csproj**. Итак, откроем данный файл. Для этого либо двойным кликом левой кнопкой мыши нажмем на название проекта, либо нажмем на название проекта правой кнопкой мыши и в появившемся меню выберем пункт **Edit Project File**

После этого Visual Studio откроет нам файл проекта, который будет выглядеть следующим образом:

```
<Project Sdk="Microsoft.NET.Sdk">

  <PropertyGroup>
    <OutputType>Exe</OutputType>
    <TargetFramework>net6.0</TargetFramework>
    <ImplicitUsings>enable</ImplicitUsings>
    <Nullable>enable</Nullable>
  </PropertyGroup>

</Project>
```

Этот файл в виде кода xml определяет конфгурацию проекта и он может содержать множество элементов. Остановлюсь только на двух основных:

- **OutputType**: определяет выходной тип проекта. Это может быть выполняемое приложение в виде файла с расширением **exe**, которое запускается по нажатию. И также это может быть файл с расширением **.dll** - некоторый набор функциональностей, который используется другими проектами. По умолчанию здесь установлено значение "Exe", что значит, что мы создаем исполняемое приложение.
- **TargetFramework**: определяет применяемую для компиляции версию фреймворка .NET. В данном случае это значение "net6.0", то есть применяется .NET 6.0.

На самых ранних этапах этот файл может не понадобиться, однако впоследствии может потребоваться внести некоторые изменения в конфигурацию, и тогда может возникнуть потребность в обращении к этому файлу.