["Hello World" в стиле C#](javascript:void(0))

Для упорядочения и оформления кода в языке C# используются классы. В действительности весь выполняемый код C# должен содержаться в классе, что справедливо и для короткой программы типа "Hello World!". Ниже приведен полный текст программы, отображающей в окне консоли сообщение "Hello World!".

C#

// A Hello World! program in C#

using System;

namespace HelloWorld

{

class Hello

{

static void Main()

{

System.Console.WriteLine("Hello World!");

// Keep the console window open in debug mode.

System.Console.WriteLine("Press any key to exit.");

System.Console.ReadKey();

}

}

}

[Директивы using и пространства имен](javascript:void(0))

При создании консольного приложения с Visual C#, экспресс-выпуск, в первой линии в редакторе кода содержится директива **using** с перечислением нескольких пространств имен .NET Framework. Пространство имен позволяет, в некотором смысле, сгруппировать вместе классы и структуры, что ограничивает их область действия и позволяет избежать конфликта имен с другими классами и структурами. При создании программы в Visual C# Express пространство имен создается автоматически. Для использования в программе классов из других пространств имен необходимо указать их с [директивой using](http://msdn.microsoft.com/ru-ru/library/sf0df423(v=vs.90).aspx). При создании нового приложения наиболее часто используемые пространства имен .NET Framework включены в список по умолчанию. При использовании классов из других пространств имен в библиотеке классов необходимо добавить директиву **using** для пространства имен к исходному файлу. Дополнительные сведения о пространствах имен см. в разделе [Пространства имен](http://msdn.microsoft.com/ru-ru/library/zz9ayh33(v=vs.90).aspx).

Когда редактор кода определит, что объявленный класс или структура не встречается в пространствах имен из текущего списка директивы **using**, он предложит пространство имен, содержащее этот класс или структуру.

[Примечания](javascript:void(0))

Следующая после оператора **using** строка содержит комментарий. Комментарии являются полезными для включения в них примечаний для себя или других программистов.

C#

// A Hello World! program in C#

Символы // преобразуют остальную часть строки в комментарий. Можно также закомментировать блок текста, поместив его между символами /\* и \*/, как в примере.

C#

/\* A "Hello World!" program in C#.

This program displays the string "Hello World!" on the screen. \*/

Для автоматического комментирования кода можно также использовать параметры форматирования. Дополнительные сведения см. в разделе [Практическое руководство. Преобразование строк кода в комментарий (C# Express)](http://msdn.microsoft.com/ru-ru/library/se4hs38a(v=vs.90).aspx).

[Классы](javascript:void(0))

В C# классы используются для оформления кода: весь выполняемый код C# должен содержаться в классе Дополнительные сведения см. в разделе [Классы](http://msdn.microsoft.com/ru-ru/library/ey4ke239(v=vs.90).aspx).

[Main()](javascript:void(0))

* В программе на C# должен присутствовать метод Main, в котором начинается и заканчивается управление. В методе Main создаются объекты и выполняются другие методы. Метод Main является [статическим](http://msdn.microsoft.com/ru-ru/library/98f28cdx(v=vs.90).aspx) методом, расположенным внутри класса или структуры. В предыдущем примере "Hello World!" он расположен в классе с именем **Program**.

Метод Main можно объявить одним из следующих способов:

* Он возвращает значение [void](http://msdn.microsoft.com/ru-ru/library/yah0tteb(v=vs.90).aspx).

C#

static void Main()

{

//...

}

* Он также может возвращать значение типа [int](http://msdn.microsoft.com/ru-ru/library/5kzh1b5w(v=vs.90).aspx).

C#

static int Main()

{

//...

return 0;

}

* Он может принимать аргументы, что может быть полезно для программ командной строки.

C#

static void Main(string[] args)

{

//...

}

или

C#

static int Main(string[] args)

{

//...

return 0;

}

Параметр метода Main является массивом значений типа [string](http://msdn.microsoft.com/ru-ru/library/362314fe(v=vs.90).aspx), представляющим аргументы командной строки, используемые для вызова программы. Обратите внимание, что в отличие от C++, массив не содержит имя исполняемого (EXE) файла. Дополнительные сведения см. в разделе [Main() и аргументы командной строки (Руководство по программированию в C#)](http://msdn.microsoft.com/ru-ru/library/acy3edy3(v=vs.90).aspx).

[Консольный ввод и вывод](javascript:void(0))

Программы консоли на C# обычно используют службы ввода/вывода, предоставляемые классом .NET Framework [Console](http://msdn.microsoft.com/ru-ru/library/system.console(v=vs.90).aspx). ОператорConsole.WriteLine("Hello, World!"); использует метод **WriteLine**. Он выводит свой строковый параметр в окно командной строки с новой строки. Другие методы **Console** используются для различных операций ввода и вывода. Класс **Console** является членом пространства имен **System**. Если в начало программы поместить директиву using System;, то нужно будет указать классы **System**, как в следующем примере.

C#

System.Console.WriteLine("Hello World!");

Метод **WriteLine** очень полезен и часто используется при создании консольных приложений.

Метод **WriteLine** может отображать строки.

C#

Console.WriteLine("Hello World!");

Метод **WriteLine** может также отображать числа.

C#

int x = 42;

Console.WriteLine(x);

Если необходимо отобразить несколько элементов, используйте {0} для представления первого элемента, {1} — для второго элемента и так далее, следующим образом.

C#

int year = 1066;

string battle = "Battle of Hastings";

Console.WriteLine("The {0} took place in {1}.", battle, year);

Результат будет выглядеть следующим образом.

The Battle of Hastings took place in 1066.