Класс String переопределяет метод Object.Equals, в результате чего сравнение происходит не по ссылке, а по значению.

Для предопределенных типов значений оператор равенства (==) возвращает значение true, если значения его операндов равны, и false в любых остальных случаях. Для ссылочных типов (кроме string) оператор == возвращает true, если оба его операнда ссылаются на один и тот же объект. Для типа string оператор == сравнивает значения строк.

При использовании строк в C#, CLR делает что-то хитрое и это что-то называется интернирование строк. Это способ хранения одной копии любой строки. Если вы храните в ста или, что еще хуже, в миллионе строковых переменных одинаковое значение получится, что память для хранения значений строк будет выделяться снова и снова. Интернирование строки это способ обойти эту проблему. Среда CLR поддерживает таблицу называемую пул интернирования. Эта таблица содержит одну уникальную ссылку на каждую строку, которая либо объявлена, либо создана программно во время выполнения вашей программы.

При помощи оператора + для конкатенации получаем поведение, как если бы вы использовали литерал «abccba», но String.Format() выполняется во время выполнения, а это значит что мы получаем совсем другие инструкции языка IL:

Код:

using System;

namespace ConsoleApp1

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("abc" + "cba");

Console.WriteLine("abccba");

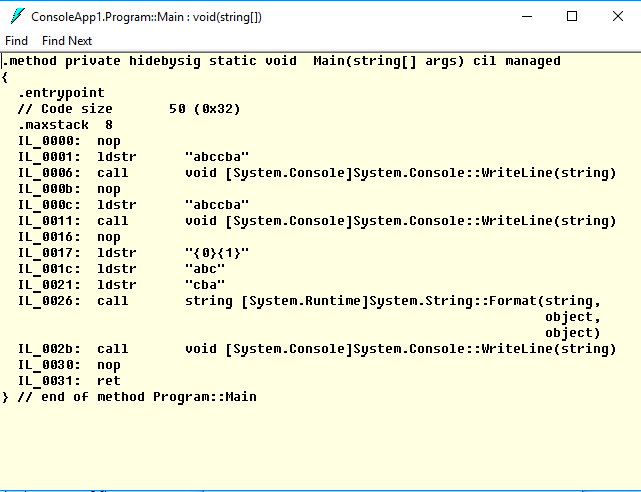
Console.WriteLine(String.Format("{0}{1}", "abc", "cba"));

}

}

}

Инструкции языка IL:

[](https://i.stack.imgur.com/vi2eU.png)

Как видите, мы получили две одинаковые строки для первой и второй команды, и отличную от них строку для 3 команды. Соответственно в пуле интернирования они будут представляться разными строками.