

Объединение строк

1. Использование операции +.

```
Console.WriteLine("I am" + 18 + ".")
```

```
Console.WriteLine("I am" + 18.ToString() + ".")
```

Является простейшим методом конкатенации строк. Операция + эквивалента методу `Concat` и транслирует операцию + в вызов `Concat`. Формат написания кода через + является более читабельным, чем прямое использование `Concat`.

В нашем данных случаях конкатенирует 3 указанных экземпляра `System.String`. (`string(I am)`, `string(18/18.ToString())`, `string (.)`).

2. `String.Format` метод.

```
Console.WriteLine($"I am {18}.")
```

`String.Format` более сложный, чем обычная конкатенация. Для читабельности лучше использовать интерполяцию строк (исп. `$" { }."`). Метод `String.Format()` форматирует строки в желаемом формате, вставляя объекты и переменные с заданными пробелами и выравниваниями в другие строки и литералы. Используется для форматирования строк в определенные форматы. В `String.Format` есть заполнители для объектов, начиная с 0-го индекса. Индекс заменяется объектом, переданным в метод.

`String.Concat` просто объединяет строки вместе. Он не обеспечивает преобразований, кроме вызова `ToString`, без форматирования и т. Д.

`String.Format` гораздо богаче, позволяет использовать шаблоны форматирования и т. д.

На небольшом объеме строки разница по времени выполнения не велика.

```

class Program
{
    static void Main(string[] args)
    {
        Stopwatch stopwatch1 = new Stopwatch();
        stopwatch1.Start();
        Console.WriteLine("I am" + 18 + ".");
        stopwatch1.Stop();
        Console.WriteLine(stopwatch1.Elapsed);

        Stopwatch stopwatch2 = new Stopwatch();
        stopwatch2.Start();
        Console.WriteLine("I am" + 18.ToString() + ".");
        stopwatch2.Stop();
        Console.WriteLine(stopwatch2.Elapsed);

        Stopwatch stopwatch3 = new Stopwatch();
        stopwatch3.Start();
        Console.WriteLine($"I am {18}.");
        stopwatch3.Stop();
        Console.WriteLine(stopwatch3.Elapsed);
    }
}

```

Microsoft Visual Studio

```

I am18.
00:00:00.0174003
I am18.
00:00:00.0000335
I am 18.
00:00:00.0001631

C:\Users\vovab\source
0.
To automatically cl
le when debugging s
Press any key to cl

```

```

[9 13 - 9 50]
_0001: ldstr      "I am"
_0006: ldc.i4.s    18 // 0x12
_0008: stloc.0     // V_0
_0009: ldloc.s     V_0
_000b: call        instance string [System.Runtime]System.Int32::ToString()
_0010: ldstr      "."
_0015: call        string [System.Runtime]System.String::Concat(string, string, string)
_001a: call        void [System.Console]System.Console::WriteLine(string)
_001f: nop

[10 13 - 10 61]
_0020: ldstr      "I am"
_0025: ldc.i4.s    18 // 0x12
_0027: stloc.0     // V_0
_0028: ldloc.s     V_0
_002a: call        instance string [System.Runtime]System.Int32::ToString()
_002f: ldstr      "."
_0034: call        string [System.Runtime]System.String::Concat(string, string, string)
_0039: call        void [System.Console]System.Console::WriteLine(string)
_003e: nop

[11 13 - 11 46]
_003f: ldstr      "I am {0}."
_0044: ldc.i4.s    18 // 0x12
_0046: box         [System.Runtime]System.Int32
_004b: call        string [System.Runtime]System.String::Format(string, object)
_0050: call        void [System.Console]System.Console::WriteLine(string)
_0055: nop

[12 9 - 12 10]
_0056: ret

end of method Program::Main

mod public hidebysig specialname rtspecialname instance void
:or() cil managed

ixstack 8

_0000: ldarg.0     // this
_0001: call        instance void [System.Runtime]System.Object::.ctor()
_0006: nop
_0007: ret

end of method Program::.ctor

```