

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

```
// [10 13 - 10 47]
IL_0001: ldstr      "I am"
IL_0006: ldc.i4.s    18 // 0x12
IL_0008: stloc.0      // num
IL_0009: ldloc.s      num
IL_000b: call         instance string [System.Runtime]System.Int32::ToString()
IL_0010: ldstr      "."
IL_0015: call         string [System.Runtime]System.String::Concat(string, string, string)
IL_001a: call         void [System.Console]System.Console::WriteLine(string)
IL_001f: nop
```

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

```
// [19 17 - 19 18]
IL_0020: ldstr      "I am"
IL_0025: ldc.i4.s    18 // 0x12
IL_0027: stloc.0      // num
IL_0028: ldloc.s      num
IL_002a: call         instance string [System.Runtime]System.Int32::ToString()
IL_002f: ldstr      "."
IL_0034: call         string [System.Runtime]System.String::Concat(string, string, string)
```

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

```
IL_003f: ldstr      "I am {0}."

// [24 21 - 24 38]
IL_0044: ldc.i4.s    18 // 0x12
IL_0046: box         [System.Runtime]System.Int32

// [25 21 - 25 40]
IL_004b: call         string [System.Runtime]System.String::Format(string, object)
IL_0050: call         void [System.Console]System.Console::WriteLine(string)
IL_0055: nop
```

Видим, что упаковка производится в третьем варианте. При использовании интерполяции строк. Особенность данного использования, каждое выражение интерполяции должно быть строкой и должно быть временной константой времени компиляции. Т.к. наша строка string формата то интерполируемая строка преобразуется в вызов String.Format, а компилятор в свою очередь может заменить на String.Concat. В других примерах изначально вызывается String.Concat. Также не затрачивает время на конвертацию "18" из Int32 в string в отличие от двух первых.