

Форматиране на извежданите данни

1. Форматиран изход

За да се направи конзолното приложение по-удобно за потребителя;

При въвеждане всички данни са във вид на низ и трябва да се преобразуват с метода Parse;

Извеждат се директно константи, променливи и стойности на изрази от всички познати типове;

А) Метод `Console.Write(израз);`

- Извежда на конзолата стойността на израза;

Б) Метод `Console.WriteLine(израз);`

- Извежда на конзолата стойността на израза, след което преминава на нов ред;

Примери

I начин:

```
Console.Write("Виe сте на ");
```

```
Console.Write(years);
```

```
Console.Write(" години");
```

II начин:

```
Console.WriteLine("Виe сте на " + years + " години");
```

2. Общ вид на методите за извеждане

WriteLine(<форматиращ низ>, <списък от изрази>)

Write (<форматиращ низ>, <списък от изрази>)

<списък от изрази> се състои от няколко израза, разделени един от друг със запетаи, номерирани с 0, 1, 2 и т.н. в реда, по който се срещат в списъка;

<форматиращ низ> е константен низ, съдържащ текст, който ще се извежда и който съдържа форматиращи елементи.

Примери

```
Console.WriteLine("Сумата е {0} лева и {1} стотинки", lev, st);
```

На мястото на форматиращият елемент {0} ще се изведе стойността на променливата lev;

На мястото на форматиращият елемент {1} ще се изведе стойността на променливата st;

3. Форматиране в посочен брой позиции

Ако след номера на форматиращият елемент е записано още едно число, разделено със запетая от него – задава се броя позиции, в които да се изведе стойността;

Ако стойността заема по-малко позиции от посочената, тогава в излишните позиции ще се изведат интервали;

Ако броят на позициите е недостатъчен, ще се заделят толкова позиции, колкото трябва.

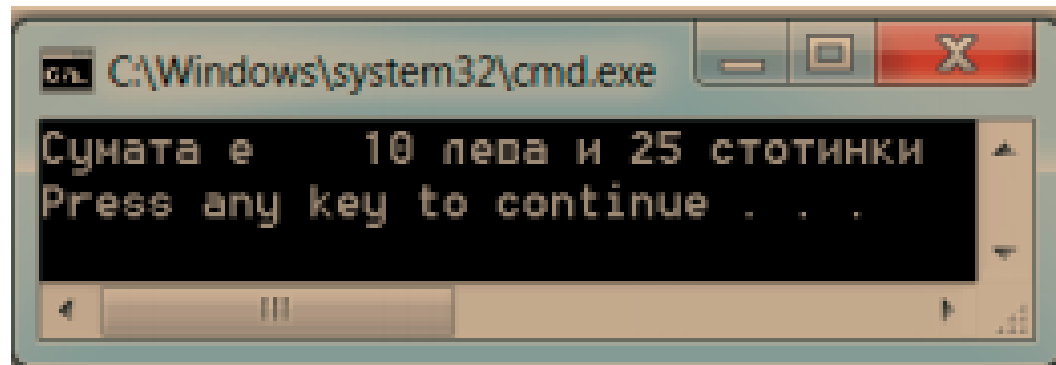
Пример

```
Static void Main(string[] args)
```

```
{ int lev=10, st=25;
```

```
Console.WriteLine(“Сумата е {0, 5} лева и {1, 1} стотинки”, lev, st);
```

```
}
```



4. Форматиране в графично приложение

Чрез метода `String`

Пример:

```
double temp = 20.4;
```

```
string s = String.Format (“Температурата е {0} °C.”, temp);
```

```
Label1.Text = s;
```

5. Управляващи знаци

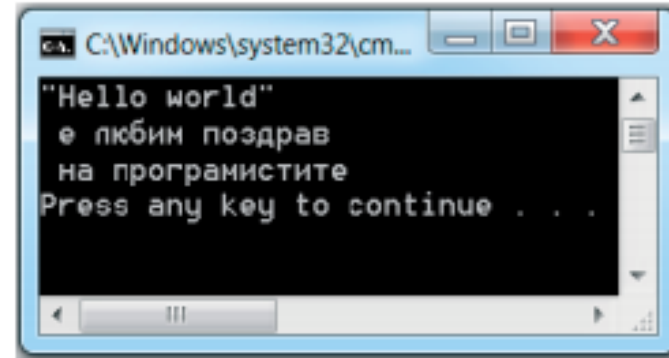
'\n' – премини на нов ред (new line)	'\'' – изведи апостроф
'\a' – издай звук (system beep)	'\"' – изведи кавички
'\b' – изтрий последния знак (backspace)	'\{' – изведи лява фигурна скоба
'\r' – върни в началото на реда (return)	'\}' – изведи дясна фигурна скоба
'\t' – хоризонтална табулация (tab)	'\\' – изведи обратна наклонена черта

Ако пред форматиращия низ се постави знакът "@", то зададеният текст ще се изведе така, както е зададен в редактора, дори и да е на повече от един ред, заедно с включените интервали, знаци за табулация и т.н.

Примери

```
class Program
{ static void Main(string[] a)
  {
    Console.Write("\Hello world"
+ "\n е любим поздрав\n"
+ " на програмистите\n");
  }
}
```

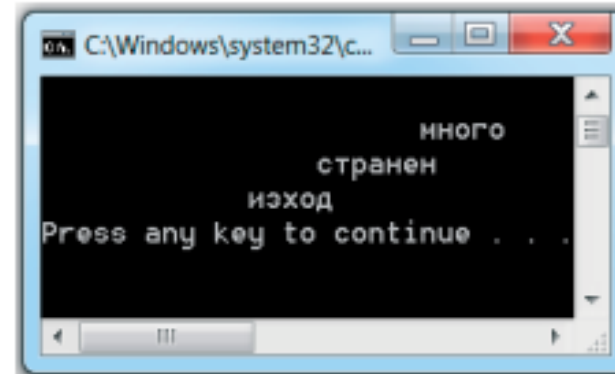
а.



б.

```
static void Main(string[] args)
{
  Console.WriteLine(@"
    МНОГО
    странен
    изход");
}
```

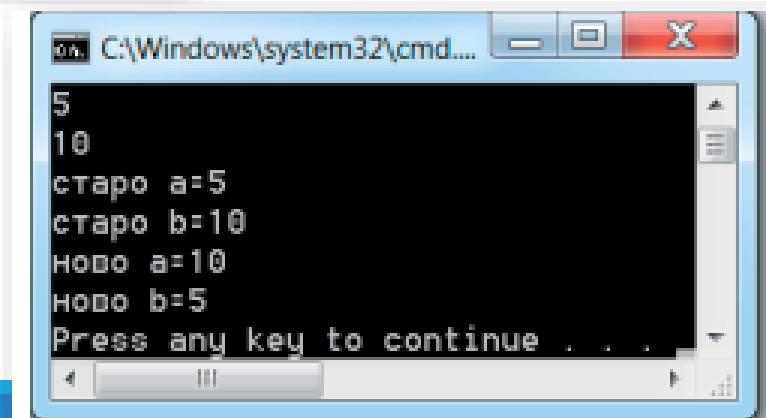
а.



б.

6. Задача: Напишете програма, която разменя стойностите на две променливи.

```
using System;
class Program
{ static void Main(string[] args)
  {
    int a, b, c; string s;
    s=Console.ReadLine(); a = int.Parse(s);
    s=Console.ReadLine(); b = int.Parse(s);
    Console.WriteLine("старо a={0}/n старо b={1}", a, b);
    c = a; a = b; b = c;
    Console.WriteLine("ново a={0}/n ново b={1}", a, b);
  }
}
```



```
C:\Windows\system32\cmd...
5
10
старо a=5
старо b=10
ново a=10
ново b=5
Press any key to continue . . .
```