

Низови

Сви до сада коришћени типови података имају једну заједничку особину: сваки се користи за смештање једне вредности.

У случајевима потребе за баратањем са већом количином података, употреба ових типова података није погодна.

Ако је потребно радити са неколико вредности истог типа истовремено, није погодно радити са више различитих промењивих за сваку од тих вредности.

Пример: Потребно је манипулисати са више имена људи, па је могуће користити више различитих промењивих:

```
string ime1 = "Miki Mikic";
string ime2 = "Ana Anic";
string ime3 = "Jovan Jovanovic";
```

Да би се обрадили овакви подаци, неопходно је писати посебан код за сваки од примера, што усложњава кодовање.

Алтернатива је употреба низова (arrays).

Низови су индексирани листе промењивих смештене у једну промењиву низовног типа.

Нпр, може се са низом имена чувати три имена људи из претходног примера.

Декларација промењивих у низу

Промењива низа се декларише навођењем типа елемента, пара угластих заграда и назива промењиве.

```
<tipPodataka> [] <imeNiza>;
```

Угласе заграде означавају да је та промењива низ.

У синтакси, <tipPodataka> може бити било који тип.

Нпр, да би се декларисао низ промењивих типа int под називом mojNiz:

```
int[] mojNiz;
```

Низови се морају иницијализовати пре него им се приступи.

Следећи код се не може извршити пошто низ није прво иницијализован да би се могло приступити неком елементу низа:

```
int[] mojNiz;
mojNiz[10] = 5;
```

Иницијализација низа

Низови се могу иницијализовати на три начина.

Први начин је специфицирањем потпуног садржаја низа у форми литерала, а то значи коришћење листе елемената одвојених зарезима унутар витичастих заграда:

```
int[] mojNiz = { 5, 9, 10, 2, 99 };
```

У примеру, низ mojNiz има пет елемената, при чему је сваком додељена целобројна вредност.

Други начин је специфицирање величине низа и употреба службене речи new чиме се иницијализују сви елементи низа.

Иницијализација ових елемената се врши са подразумеваним вредностима; 0 за целобројне елементе, false за елементе логичких вредности и null.

Са `int[] mojNiz = new int[5];` се декларише низ `mojNiz` са елементима типа `int` али и иницијализује пет елемената тог низа на вредност 0.

Такође, декларација и иницијализација се не морају одвијати на истој линији кода:

```
int[] mojNiz;
mojNiz = new int[5];
```

Уместо константне вредности за дефинисање величине низа, може се користити неконстантна промењива, која се може израчунавати у тренутку извршавања:

```
int velicinaNiza = Convert.ToInt32(Console.ReadLine());
int[] mojNiz = new int[velicinaNiza];
```

Трећи начин је комбинација прва два начина иницијализације:

```
int[] mojNiz = new int[5]{ 5, 9, 10, 2, 99 };
```

Са овом методом, величина низа и број вредности за елементе низа се морају поклапати.

Када би се за величину низа користила промењива, онда та промењива мора бити константа:

```
const int velicinaNiza = 5;
int[] mojNiz = new int[velicinaNiza] { 5, 9, 10, 2, 99 };
```

Пример: Приказати све вредности елемената низа

```
using System;
namespace Proba
{
    internal class Program
    {
        static void Main(string[] args)
        {
            string[] imena = { "Miki", "Ana", "Dragana" };
            int i;
            Console.WriteLine($"Ovo su {imena.Length} imena:");
            for (i = 0; i < imena.Length; i++)
            {
                Console.WriteLine(imena[i]);
            }
        }
    }
}
```