Петље и низови

Једна од лоших страна коришћења класичне for петље за појединачан приступ елементима низа јесте могућност прекорачења дозвољеног индекса низа чиме се појављује грешка и прекид програма.

Зато се препоручује коришћење стабилнијег начин приступа елементима низа, помоћу команде foreach.

Команда foreach

Петља foreach омогућава приступ сваком елементу низа преко синтаксе:

```
foreach (<tipPodataka> <nazivPromenjive> in <nazivNiza>)
{
      ..
}
```

Петља ће проћи преко сваког елемента, смештајући их један по један у промењиву <nazivPromenjive>, без бојазни да ће се петљом приступити индексу непостојећег елемента у низу.

Овиме се не мора бринути ни о броју елемената у низу нити о томе да ли ће неки елемент бити прескочен у обради.

Пример:

```
using System;
namespace Proba
{
    internal class Program
    {
        static void Main(string[] args)
        {
            string[] imena = { "Miki", "Ana", "Dragana" };
            Console.WriteLine($"Ovo su {imena.Length} imena:");
            foreach (string ime in imena)
            {
                Console.WriteLine(ime);
            }
        }
    }
}
```

Једина разлика између употребе foreach и класичне for петље је у томе што fopeach даје само приступ читања елемената низа, и не допушта промену вредности тих елемената.

Команда switch и низови

Команда switch се може користити за поређење шема (match patterns) на основу типа промењиве, нпр string или int низа промењивих.

Када се зна тип промењиве, може се прићи методима и особинама које се користе у том типу података.

Пример:

```
switch (<testPromenjiva>)
{
     case int vrednost:
          <kod ako je <testPromenjiva> tipa int>
          break;
     case string s when s.Length == 0:
          <kod ako je <testPromenjiva> string duzine = 0>
          break;
          ...
     case null:
          <kod ako je <testPromenjiva> = null>
          break;
     default:
          <kod ako je <testPromenjiva> != svih proverenih vrednosti >
          break;
}
```

У примеру тест промењива може бити и било који низ који садржи и различите типове података.
Одмах после службене речи case је тип промењиве који се жели проверити.
Вредност тог типа када постоји поклапање је смештена у декларисану промењиву.
Нпр, ако је тестирана промењива типа целобројне вредности, целобројна вредност је смештена у промењиву са именом vrednost.
Употреба when модификатора службене речи омогућава додавање услова потребних за извршење кода који је унутар case исказа.