tekst preuzet sa http://web.zpr.fer.hr/ergonomija/2004/bosnjakm/literatura.html

Stringovi

Niz znakova se deklaše kao **`string`** (string je alijas za String tip). Za razliku od drugih referencnih tipova, ovde se može zaobići new:

```
string s = "Hello, World!";
```

Spajanje dva stringa:

```
string s1 = "orange";
string s2 = "red";

s1 += s2;
System.Console.WriteLine(s1);  // outputs "orangered"
```

Izdvajanje podstringa:

```
s1 = s1.Substring(2, 5);
System.Console.WriteLine(s1);  // outputs "anger"
```

String objekti su nepromenljivi! To znaci da se ne mogu promeniti nakon što su stvoreni. Metode koje rade sa stringovima zapravo vracaju novi string objekt.

U prethodnom primeru, kad se **`s1`** i **`s2`** konkateniraju u jedan string, ta dva stringa koji sadrže vrednosti **`"orange"`** i **`"red"`** su nepromenjeni. Operator **`+=`** stvara novi string koji sadrži kombinovani rezultat. Nakon toga s1 ima referencu na drugi string. String koji sadrži **`"orange"`** još uvijek postoji, ali više nema reference na njega.

Zbog toga, radi poboljšanja performansi, ako radimo veliki broj manipulacija nad stringovima, trebali bi koristiti klasu **`StringBuilder`**, kao u primeru:

```
System.Text.StringBuilder sb = new System.Text.StringBuilder();
sb.Append("one ");
sb.Append("two ");
sb.Append("three");
string str = sb.ToString();
```

StringBuilder

`StringBuilder` klasa stvara string buffer koji nudi bolje performanse od običnog stringa ako se u programu obavlja puno manipulacija sa string objektima. StringBuilder osim toga omogućuje i naknadnu promenu pojedinih znakova. Primer prikazuje kako promieniti sadržaj stringa bez da stvorimo novi objekt:

```
System.Text.StringBuilder sb = new System.Text.StringBuilder("Rat: the ideal
pet");
sb[0] = 'C';
System.Console.WriteLine(sb.ToString());
System.Console.ReadLine();
```

U ovom primjeru se StringBuilder objekt koristi da bi se stvorio string iz seta numeričkih tipova:

```
class TestStringBuilder
{
    static void Main()
    {
        System.Text.StringBuilder sb = new System.Text.StringBuilder();

        // Create a string composed of numbers 0 - 9
        for (int i = 0; i < 10; i++)
        {
            sb.Append(i.ToString());
        }
        System.Console.WriteLine(sb);  // displays 0123456789

        // Copy one character of the string
        // (not possible with a System.String)
        sb[0] = sb[9];

        System.Console.WriteLine(sb);  // displays 9123456789
    }
}
```

Rad sa stringovima

Isto kao i u C++ jeziku, ako želimo ubaciti posebne znakove u string, koristimo *escape* znakove, poput `"\n"` za novi red ili `"\t"` za tab.

```
string hello = "Hello\nWorld!";
```

Prethodna naredba ispise rijeci Hello i World u dva odvojena retka. Ako želimo ubaciti backslash u string, onda napišemo dvije kose crte:

```
string filePath = "\\\\My Documents\\";
```

Simbol @

Simbol @ govori konstruktoru stringa da ignorira *escape* znakove. Sledeca dva stringa su identična:

```
string p1 = "\\\\My Documents\\My Files\\";
string p2 = @"\\My Documents\My Files\";
```

ToString()

Kao i svi objekti izvedeni iz klase Objekt, string ima `ToString()` metodu koja pretvara vrednost u string. Ova metoda se koristi da se npr. numericke vrednosti pretvore u stringove:

```
int year = 1999;
string msg = "Eve was born in " + year.ToString();
System.Console.WriteLine(msg);  // outputs "Eve was born in 1999"
```

Pristup pojedinim znakovima

Pojedini znakovi koji su sadržani u stringu se mogu dobiti pomoću metoda **SubString()**, **Replace()**, **Split()** i **Trim()**.

```
using System;
string s3 = "Visual C#";
Console.WriteLine(s3.Substring(7, 2));         // outputs "C#"
Console.WriteLine(s3.Replace("C#", "Basic")); // outputs "Visual Basic"
```

Moguće je i kopiranje znakova u niz znakova, poput:

```
string s4 = "Hello, World";
char[] arr = s4.ToCharArray(0, s4.Length);

foreach (char c in arr)
{
    System.Console.Write(c);  // outputs "Hello, World"
}
```

Pojedini znakovi iz stringa se mogu dobiti i pomoću indeksa:

```
string s5 = "Backwards";

for (int i = 0; i < s5.Length; i++)
{
    System.Console.Write(s5[s5.Length - i - 1]);  // outputs "sdrawkcaB"
}
```

Velika i mala slova

Da promijenimo slova u stringu u mala ili velika, možemo koristiti ugrađene metode **ToUpper()** i **ToLower()**:

```
string s6 = "Battle of Hastings, 1066";

System.Console.WriteLine(s6.ToUpper());  // outputs "BATTLE OF HASTINGS
1066"
System.Console.WriteLine(s6.ToLower());  // outputs "battle of hastings
1066"
```

Usporedba

Najjednostavniji način da se uporede dva stringa je pomocu **==** i **!=** operatora, koji obavljaju proveru uzimajuci u obzir i velika i mala slova (*case sensitive*).

```
string color1 = "red";
string color2 = "green";
string color3 = "red";

if (color1 == color3)
{
    System.Console.WriteLine("Equal");
}
if (color1 != color2)
{
```

```
    System.Console.WriteLine("Not equal");
}
```

String objekti imaju i **`CompareTo()`** metodu koja vraca integer vriednost na osnovu toga da li je jedan string manji ili veći od drugog. Pri tome se misli na Unicode vrednost znakova u stringu.

```
string s7 = "ABC";
string s8 = "abc";

if (s7.CompareTo(s8) > 0)
{
    System.Console.WriteLine("Greater-than");
}
else
{
    System.Console.WriteLine("Less-than");
}
```

Da bi pronašli neki string unutar stringa, koristimo **`IndexOf()`** metodu koja vraća indeks znaka u stringu gde se prvi put pojavljuje zadani string. Ako ne pronađe zadani string, vraća vrednost -1.

```
string s9 = "Battle of Hastings, 1066";

System.Console.WriteLine(s9.IndexOf("Hastings"));  // outputs 10
System.Console.WriteLine(s9.IndexOf("1967"));      // outputs -1
```

Podjela stringa na podstringove

Podela stringa na podstringove, poput recimo podela recenice u pojedine rijeci, je čest programski zadatak. **`Split()`** metoda uzima kao parametar niz znakova u kojem se nalaze delimiteri po kojima ce se izvršavati podjela (npr. razmak), a vraća niz podstringova.

```
char[] delimit = new char[] { ' ' };
string s10 = "The cat sat on the mat.";
foreach (string substr in s10.Split(delimit))
{
    System.Console.WriteLine(substr);
}
```