

Увод у стрингове

У пракси се често упоребљава обрада текста: читање текст фајлова, претрага података према траженој речи и њихова замена, валидација улаза од стране корисника...

Стринг је секвенца карактера смештена на некој меморијској адреси.

У .NET Framework сваки карактер има серијски број у Уникод табели која садржи различите типове текстуалних података.

Класа System.String

Класа System.String омогућава рад са стринговима у C#.

Приликом декларације стрингова, користиће се службена реч string, која је алијас у C# за класу System.String из .NET Framework.

Пример: `string mojString = "Tekst string";`

На овај начин је декларисана промењива `mojString` као тип стринг са вредности "Teskt string".

Унутрашње представљање класе види вредност као низ карактера.

Могуће је избећи коришћење класе помоћу декларације промењиве као типа `char[]` и попунити елементе низа карактер по карактер, али то са собом носи и неке мање ефикасне технике рада са стринговима.

Тип података стринг је специјалан пошто је он класа и његови подаци су унутар динамичке меморије (heap) а промењиве типа стринг имају референцу на објекат запамћен у хипу.

Посебна особина класе стринг је то што секвенце знакова смештене у промењиву класе се не могу мењати, стрингови су непромењиви (immutable).

Ниске знакова и стрингови

Стрингови су слични низовима знакова (ниска знакова, `char[]`) који се могу мењати по потреби.

Стрингови имају функције `Length` (враћа број знакова у стрингу), омогућавају индексирање позиција знакова у стрингу:

```
string str = "abcde";
char ch = str[1]; // ch == 'b'
str[1] = 'a'; // Compilation error!
ch = str[50]; // IndexOutOfRangeException
```

Пример: Штампање стринга, употреба добијања дужине стринга, штампање карактера из стринга.

```
using System;
namespace Proba
{
    class Program
    {
        static void Main()
        {
            string poruka = "Ovo je mala poruka.";
            Console.WriteLine($"poruka = {poruka}");
            Console.WriteLine($"Duzina poruke je {poruka.Length}");
            for (int i = 0; i < poruka.Length; i++)
            {
```

```

        Console.WriteLine($"poruka[{i}] = {poruka[i]}");
    }
}
}
}

```

Ескејп карактери у стрингу

Да би се користили наводници као део стринга, испред њих се мора ставити \, чиме се указује да желимо наводнике као знак и да не користимо на том месту наводнике за крај стринга:

```
string citat = "Naslov knjige je \"Uvod u C#\"";
```

Декларација стринга

Синтакса за декларацију промењиве типа стринг:

```
string <nazivPromenjive>;
```

Овај начин декларације не значи постављање промењиве и обезбеђивање меморије нити је промењива доступна за обраду (тренутна њена вредност је null).

Овом декларацијом информишемо компајлер да ће се промењива користити и да је очекивани тип података за њу стринг.

Типови иницијализације

Постављање стринг литерала значи доделу предефинисаног текстуалног контекста промењивој типа стринг.

Овај тип иницијализације се користи када знамо вредност која се мора сместити у промењиву:

```
string vebсайт = "http://www.wikipedia.org";
```

Додела вредности је идентична са усмеравањем стринг вредности или промењиве према промењивој типа стринг:

```
string prvi = "Prvi tekst";
string drugi = prvi;
```

Прво се декларише и иницијализује промењива prvi а затим промењива drugi добија вредност промењиве prvi.

Пошто је класа string референца типа, текст "Prvi tekst" је смештен у динамичку меморију (хип) на адресу дефинисану првом промењивом, тј обе промењиве упућују на исту адресу на хипу.

Промена на било којој промењивој ће се одразити само на тој промењивој, пошто је тип стринг непормењив, а када се деси промена, копија измењеног стринга ће се креирати.

Ово се не дешава са осталим типовима података пошто се код њих промене дешавају директно на меморијској адреси и све референце су усмерене на ту адресу.

Трећа врста иницијализације је додавање вредности стринг израза или операције, чиме се добија стринг као резултат:

```
string mejl = "neki@gmail.com";
string info = "Moj mejl je: " + mejl;
```

Читање и штампање стринга

Употребом класе System.Console:

```
string ime = Console.ReadLine();
```

Овако се чита са конзоле улаз података чекајући кориснички унос и притисак на ЕНТЕР.

Штампање садржаја стринга употребом класе System.Console:

```
Console.WriteLine("Tvoje ime je: " + ime);
```