

Одстрањивање знакова са почетка и краја стринга

Понекад се дешава да се приликом корисничког уноса текста појаве непотребни карактери на почетку или на крају текста.

За одстрањивање ових карактера се користи метода Trim() (празна места, табови, прекиди линија...):

```
string tekst = " \n\n Miki ";
string ispravljenTekst = tekst.Trim();
Console.WriteLine(ispravljenTekst);
```

Одстрањивање знакова из низа знакова

Метода Trim() може да прихвати низ карактера које је потребно одстранити из стринга:

```
string fajl = " 111 $ % Miki ### s ";
char[] znaciZaOdstranjivanje = new char[] { ' ', '1', '$', '%', '#',
                                             's' };
string ispravljeniFajl = fajl.Trim(znaciZaOdstranjivanje);
```

Са методама TrimStart() и TrimEnd() знаци се одстрањују са предвиђеног дела стринга.

StringBuilder класа

Приликом придруживања стрингова троши се доста меморијских локација што може да успори извршење програма.

Успоравање се највише односи на процес сакупљања ђубрета (garbage collector) којим се чисти динамичка меморија од непотребних података.

Употреба петљи у раду са стринговима вишеструко успорава обраду стрингова.

У сваком циклусу итерације се дешавају следећи процеси: дефинише се меморија (бафер) за смештање следећег податка, стари стринг се смешта у нови бафер, следећи податак се повезује са бафером, бафер се конвертује у стринг а непотребни подаци се уклањају сакупљањем ђубрета. Најефикаснији начин повезивања стрингова путем петљи је употреба StringBuilder класе.

Објекти класе StringBuilder нију непромењиви нити се мора током њихових измена креирати копија објекта у меморији.

За поступке модификације се кристи посебан бафер, дужине која се може динамички мењати, којем се без посебних захтева може модификовати и мењати садржај, што чини операције над стринговима бржим и једноставнијим.

метода	опис
StringBuilder(int kapacitet)	конструктор, користи се за креирање иницијалног објекта
Length	враћа дужину стринга у баферу
Append	придодаје стринг после последњег унетог карактера
Clear	брише све карактере из бафера
Remove(pozicija, duzina)	брише стринг из бафера од дате позиције са датом дужином
Insert(pozicija)	убацује стринг на дату позицију

Форматирање стринга

У .NET сваки објекат се може приказати као текст и то помоћу методе ToString(), и он се аутоматски позива када треба одштампати објекте различитих типова на конзоли.

```
DateTime trenutniDatum = DateTime.Now;
Console.WriteLine(trenutniDatum); //isto kao trenutniDatum.ToString()
```

String.Format() је статичка метода којом се може форматирати текст и други подаци кроз темплејт. Темплејт садржи текст и декларисане параметре (placeholders) и користе се за добијање форматираног текста када се параметри замене са конкретним вредностима.

```
using System;
namespace Proba
{
    class Program
    {
        static void Main()
        {
            DateTime datum = DateTime.Now;
            string ime = "Mica";
            string knjiga = "knjigu 'Uvod u C#'";
            string mesto = "skoli";
            string formatizovanTekst = String.Format($"Danas je
                {datum} i {ime} cita {knjiga} u {mesto}.");
            Console.WriteLine(formatizovanTekst);
        }
    }
}
```

У примеру String.Format() користи параметре ({datum}, {ime}...) и реализује форматизовани стринг.

Парсовање података

Супротан процес од форматизовања јесте парсовање података, добијање вредности датог типа из текстуалне репрезентације те вредности у датом формату, тј конверзији из текста у неки други тип података.

```
string tekst = "11/12/2001";
string format = "MM/dd/yyyy";
DateTime parsovaniDatum = DateTime.ParseExact(tekst, format,
    CultureInfo.InvariantCulture);
Console.WriteLine($"Day: {parsovaniDatum.Day}\nMonth:
    {parsovaniDatum.Month}\nYear: {parsovaniDatum.Year}");
```