

Упоређивање једнакости стрингова

Ако се стрингови пореде према еквиваленцији њихових вредности, најједноставнији начин за то је метода Equals().

Ова метода враћа Булову вредност (true или false) у зависности да ли два стринга имају исте вредности или не.

Метода пореди сваки знак у стрингу на идентичним позицијама и узима у обзир велика и мала слова која нису идентична.

```
string rec1 = "C#";
string rec2 = "c#";
Console.WriteLine(rec1.Equals("C#"));    //True
Console.WriteLine(rec1.Equals(rec2));    //False
Console.WriteLine(rec1 == "C#");        //True
Console.WriteLine(rec1 == rec2);        //False
```

У случају да нам нису битна мала и велика слова већ само садржај стрингова:

```
Console.WriteLine(word1.Equals(word2,
    StringComparison.CurrentCultureIgnoreCase)); //True
```

Поређење стрингова по алфабетном редоследу

Да би се упоредили стрингови који је пре другог стринга по алфабетском редоследу, користи се метода CompareTo().

Да би у овом случају два стринга исте вредности морају бити исте дужине и сви карактери морају бити на истим местима идентични.

Метода CompareTo() се налази у класи String и враћа негативну вредност, нулу или позитивну вредност.

Негативна вредност значи да је први стринг по редоследу пре другог, нула значи да су једнаки, позитивна вредност значи да је први стринг после другог.

Пример:

```
string miki = "Miki";
string mica = "Mica";
Console.WriteLine(miki.CompareTo(mica)); //1
Console.WriteLine(mica.CompareTo(miki)); //-1
Console.WriteLine(miki.CompareTo(miki)); //0
```

У случају да се тражи игнорисање величине слова у стринговима:

```
string alfa = "alfa";
string skor1 = "sKor";
string skor2 = "skor";
Console.WriteLine(string.Compare(alfa, skor1, false)); //-1
Console.WriteLine(string.Compare(skor1, skor2, false)); //1
Console.WriteLine(string.Compare(skor1, skor2, true)); //0
Console.WriteLine(string.Compare(skor1, skor2,
    StringComparison.CurrentCultureIgnoreCase)); //0
```

Види се да се може користити метода: `string.Compare(string str1, string str2, bool ignoreCase)` која има опцију игнорисања величине слова која се може поставити на тачно или нетачно.

#### Оптимизација меморије за стрингове

У коду:

```
string ime1 = "Miki";  
string ime2 = "Miki";
```

У случајевима када С# компајлер препознаје да треба да се ради са дуплицираним вредностима у различитим стринговима, компајлер смањује потрошњу меморије оптимизацијом која се зове интернисања стрингова, па две променљиве ће указивати на идентичан садржај у меморији.

#### Придруживање стрингова

Пример:

```
string a = "Miki ";  
string b = "Mikic";  
string c = string.Concat(a, b); //Miki Mikic
```

Могу се придруживати подаци других типова са стринговима:

```
string poruka = "Imam ";  
int godine = 10;  
string recenica = poruka + godine;
```

Употреба придруживања стрингова не мења постојеће стрингове већ враћа нови стринг као резултат.