

Nizi, znaki

Delo z nizi – tip string

Delo z znaki – tip char

Niz (string) – zaporedje znakov

- ❑ Ustvarjanje spremenljivke tipa nit (string)
`string ime = "Polde";`
- ❑ Niz - zaporedje znakov med znakoma " "
- ❑ Spajanje nizov: operator +
`string imeInPriimek = "Polde " + "Bibič";`
- ❑ Dodatne presledke moramo dodati sami
- ❑ Če en izraz NI niz, se ta pretvori v niz!
`string Naslov = "Cankarjeva ulica " + 23;`

- ❑ Branje nizov
 - metoda `Console.ReadLine()`

Dolžina niza, posamezni znak v nizu, ...

- Dolžina niza: lastnost `Length`
 - `string priimek = "Bibič";`
 - `priimek.Length` → 5
 - `"Cankar".Length` → 6
 - `(priimek + " " + "bla").Length` → 9
 - Dostop do znaka na *i*-tem mestu v nizu [*i*]
 - `ime[i]`
 - Znake štejemo od 0 dalje!
 - `"Blabla"[3]`
 - b
 - `string ime = "Bibič";`
 - `ime[0]` → B
 - `ime[1]` → i
 - `ime[ime.Length-1]` → č
 - `Ime[4]` → č
-

Zgled – obrni niz

- ❑ Preberi niz in ga izpiši obrnjeno!
- ❑ Polde → edloP
- ❑ Zanka
 - Pregledamo vse znake v nizu (dolžina niza)
 - while (i < niz.Length)
 - Dodajamo na začetek
 - obrnjeniNiz = niz[i] + obrnjeniNiz ;
 - Program v C#

```
string niz="Polde";  
string obrnjeniNiz="";//zaenkrat prazen niz
```

```
for (int i=0;i<niz.Length;i++)  
    obrnjeniNiz = niz[i] + obrnjeniNiz ;
```

```
Console.WriteLine(obrnjeniNiz);//izpis obrnjenega niza
```

Znaki – tip char

- ❑ Tip char
 - ker (character)
 - ❑ Ustvarjanje nove spremenljivke tipa znak char znak;
 - ❑ Inicializacija (začetna vrednost) - enojni narekovaji
znak = 'm';
 - ❑ Primerjanje
 - Ali je mala črka?
 - ((znak >= 'a') && (znak <= 'z'))
 - ❑ primerjanje znakov
 - 'a' < 'b' < 'c' < ... < 'z'
 - '0' < '1' < ... < '9'
 - 'A' < 'B' < 'C' < ... < 'Z'
 - v teh treh zgornjih primerih ni med napisanimi znaki nobenih drugih znakov (znaku 'b' sledi znak 'c', znaku 'L' sledi znak 'M', znaku '6' sledi znak '7', ...)
-

Znaki - računanje

- z znaki lahko računamo , saj gre dejansko za kode znakov
 - `(char)('a' + 2) → 'c'`
 - `(char)('A' + 'k' - 'a') → 'K'`
- C# obravnava znake kot "majhna" števila. V spremenljivko tipa *char* pravzaprav shranimo kodo znaka. Ta koda je neko **naravno** število. Na vsako spremenljivko tipa *char* lahko gledamo bodisi kot na znak, bodisi kot na število

```
char znak = 'a';  
Console.WriteLine(znak + znak); //izpis 194
```

- V spremenljivko *znak* smo shranili kodo znaka mali a, ki je neko naravno število (v našem primeru 97). In ker je med dvema "številoma" operator *+*, se izvede operacija seštevanja, kot smo že navajeni. Torej se seštejeta dve številski vrednosti malega znaka a in vsota se izpiše.

Zgled – vsota števk v nizu (algoritem)

- ❑ Preberemo niz
- ❑ Pregledamo vsak znak
 - `znak = niz[i]; // tekoči znak`
- ❑ Če je števka, povečamo števec za 1
 - ```
if (('0' <= znak) && (znak <= '9')) {
 // če je števka
 koliko_stevk++; // povečanje za 1
}
```
- ❑ Izpišemo rezultat

# Zgled – številke v nizu (Program)

---

```
string niz = Console.ReadLine();//branje niza
int i = 0, kolikoStevk = 0; //začetne vrednosti
while (i < niz.Length)
{
 char znak = niz[i]; // tekoči znak niza
 if (('0' <= znak) && (znak <= '9'))
 {
 // ce je številka, povečamo števec
 kolikoStevk = kolikoStevk +1;
 }
 i = i + 1;
}
string rezultat = "Število števk v nizu: "+ kolikoStevk;
Console.WriteLine(rezultat);
```

---



# Metode in lastnosti za delo z nizi

---

- ❑ Dolžina niza – lastnost Length

```
string niz = "Programiranje 1";
int stZnakov = niz.Length;//izračun števila znakov nizu
```

- ❑ Metoda Contains

```
if (niz.Contains("ami"))
 Console.WriteLine("Ta niz vsebuje zaporednje znakov 'ami'");
```

- ❑ Metoda IndexOf: pozicija zaporedja znakov znotraj niza.

```
string niz = "Programiranje 1";
int st=niz.IndexOf("gram");
Console.WriteLine(st);
```

Dobimo izpis: 3

```
int st=niz.IndexOf("C#");
Console.WriteLine(st);
```

Ker v nizu "Programiranje 1" ni podniza C#, dobimo izpis: -1

---

# Metode in lastnosti za delo z nizi

---

- ❑ Primerjanje nizov:
  - Z relacijskim operatorjem ==  
    `s1 == s2`
  - Metodi
    - `Equals`
    - `Compare`
  - Metoda `Equals`:
    - `s1.Equals("bla")` : ali je niz shranjen v `s1` enak nizu `bla` – rezultat `true` ali `false`
      - `string s1="Bla", s2="bla";`
      - `if (s1.Equals("bla"))`
      - `Console.WriteLine("Niza sta enaka");`
      - `else`
      - `Console.WriteLine("Niza NISTA enaka");`
    - `s1.Equals(s2)` : ali je niz shranjen v `s1` enak nizu shranjenemu v `s2`

# Metode in lastnosti za delo z nizi

---

## □ Primerjanje nizov

- Nize primerjamo leksikografsko (po abecedi)
  - Niz "Marjan" je manjši kot niz "Mojca", ker sta prva znaka enaka, drugi znak pa je v prvem nizu ('a') manjši kot v drugem nizu ('o').
- `String.Compare(s1, "bla")` : vrne 0, če je niz shranjen v `s1` enak nizu `bla`, neg. število -1, če je niz v `s1` manjši od niza "bla" in poz. Število 1, če je večji.
- `String.Compare("Marjan", "Mojca")` : vrne negativno število -1

```
if (String.Compare("Marjan", "Mojca") < 0)
 Console.WriteLine("Leksikografsko gledano je prvi
niz manjši");
```

- `bool jeManj = String.Compare(n1, n2) < 0;`
-

# Metode in lastnosti za delo z nizi

---

- ❑ Insert: vstavljanje podniza v niz

```
string niz = "Programiranje 1";
string noviNiz = niz.Insert(0, "C# ");
```

noviNiz dobi vrednost *C# Programiranje 1*

- ❑ Remove: odstranjevanje podniza iz niza

```
string niz = "Programiranje 1";
//odstranimo 8 znakov, od znaka številka 7 naprej
string noviNiz=niz.Remove(7, 8);
```

Spremenljivka noviNiz dobi vrednost *Program*

Če je drugi parameter ( v našem primeru številka 8) prevelik (če torej ne obstaja toliko znakov), program javi napako!

---

# Metode in lastnosti za delo z nizi

---

- ❑ Metoda Replace: zamenjava podniza v nizu

```
string niz = "Programiranje 1";
string noviNiz = niz.Replace("1", "2");
string najnovejsiNiz = niz.Replace("Programiranje", "Vaje");
```

```
Console.WriteLine(noviNiz); //Izpis: Programiranje 2
Console.WriteLine(najnovejsiNiz); //Izpis: Vaje 1
```

- ❑ Metoda Substring: pridobivanje podniza iz niza

```
string niz = "Programiranje 1";
string noviNiz = niz.Substring(6, 3);
```

Iz niza *niz* želimo vzeti 3 znake od 6 znaka naprej!  
Niz *noviNiz* dobi vrednost *mir*

# Metode in lastnosti za delo z nizi

---

- Metodi ToLower in ToUpper: pretvorba črk v nizu v male, oz. velike črke

```
string niz = "Programiranje 1";
```

```
Console.WriteLine(niz.ToLower()); //Izpis "programiranje 1"
Console.WriteLine(niz.ToUpper()); //Izpis PROGRAMIRANJE 1
```

- Metoda Trim: odstranjevanje vodilnih in končnih presledkov iz niza (dodatno še TrimStart, TrimEnd)

```
string niz = " Programiranje 1 ";
string noviNiz = niz.Trim();
```

Novi niz dobi vrednost 'Programiranje 1'

---

# Vaja – štetje samoglasnikov

---

- Preberimo poljuben stavek in izpišimo, koliko znakov vsebuje, koliko je v njem samoglasnikov, koliko števk in koliko ostalih znakov

```
Console.Write("Vnesi poljuben stavek: ");
string stavek = Console.ReadLine();

string samogl = "AaEeIiOoUu";
int stSam = 0, stStevk = 0; //začetno število samoglasnikov in števk
for (int i = 0; i < stavek.Length; i++)
{
 char znak = stavek[i];
 if (samogl.IndexOf(znak) != -1) // znak je samoglasnik
 stSam = stSam + 1;
 if ('0' <= znak && znak <= '9') //znak je števka
 stStevk = stStevk + 1;
}
Console.WriteLine("\nŠtevilo znakov v staku: " + stavek.Length);
Console.WriteLine("Število samoglasnikov : " + stSam);
Console.WriteLine("Število cifer : " + stStevk);
Console.WriteLine("Število ostalih znakov : " + (stavek.Length -
 stSam - stStevk));
```

---

# Vaje

---

- ❑ Preberi poljuben stavek:
    - Koliko je vseh znakov v stavku;
    - Koliko je v stavku samoglasnikov;
    - Odstrani morebitne začetne in končne presledke;
    - Preveri, če prvi znak tega stavka velike začetnica. Če ni, jo naredi;
    - Preveri, če je zadnji znak tega stavka ločilo pika. Če je ni, jo dodaj;
    - Iz niza odstrani prve tri znake;  
`stavek = stavek.Remove(0, 3);`
    - Kolikokrat se v nizu pojavi znak ',' (vejica);
    - Ali sta zadnja dva znaka niza enaka prvima dvema?
-



# Vaje

---

- ❑ Izpiši zadnje tri znake tega niza (uporabi metodo Substring, nato pa še zanko!)

```
Console.Write("Vnesi poljuben stavek: ");
string stavek = Console.ReadLine();
```

```
//metoda substring
Console.WriteLine(stavek.Substring(stavek.Length-
 3, 3));
```

```
//zanka
string zadnjeTri = "";
for (int i = stavek.Length-1; i >= stavek.Length - 3;
 i--)
 zadnjeTri = stavek[i] + zadnjeTri;
Console.WriteLine(zadnjeTri);
```

---

# Vaje

---

- ❑ Preveri, kolikokrat se v tem stavku pojavi veznik 'pa';

```
Console.Write("Vnesi poljuben stavek: ");
string stavek = Console.ReadLine();
string pomocna=stavek;
int st = 0;
while (true)
{
 if (pomozna.Contains("pa"))
 {
 int n = pomocna.IndexOf("pa");
 pomocna=pomozna.Remove(n, "pa".Length);
 st++;
 }
 else break;
}
Console.WriteLine("Besedica \"pa\" se pojavi "+st+"x");
```

---

# Vaje

---

- ❑ Vse veznike 'in' v tem stavku nadomesti z znakom vejica;

```
Console.Write("Vnesi poljuben stavek: ");
string stavek = Console.ReadLine();
string pomocna=stavek; int st = 0;
while (true)
{
 if (pomozna.Contains("in"))
 {
 int n = pomocna.IndexOf("in");
 pomocna=pomozna.Replace("in", ",", "");
 st++;
 }
 else break;
}
Console.WriteLine(pomozna);
```

---

# Vaje

---

- ❑ Verjetno je vsak med nami že kdaj prebral kak strip o Asterixu in Obelixu. Zato vemo, da se imena vseh Galcev zaključijo z "ix", na primer Asterix, Filix, Obelix, Dogmatix, itn. Napišimo program, ki bo prebral ime, nato pa bo izpisal "Galec!", če gre za galsko ime, v nasprotnem primeru pa bo izpisal "Rimljan!". Predpostavimo, da so vnesena imena dolga vsaj tri znake.
- ❑ Preberi kemijsko formulo in jo izpiši tako, da so številke vrstico nižje. Če npr. prebereš formulo H<sub>2</sub>SO<sub>4</sub>, jo izpiši kot

H SO  
2 4

# Vaje

---

- Kaj izpišejo spodnji deli programov :

```
string niz = "Tantadruj";
char y = niz[5];
Console.WriteLine("V nizu je crka " + y);
```

```
string niz1 = "Tantadruj";
int n = niz1.Length, i = 0;
string niz2 = "";
while(i < n)
{
 char x = niz1[i];
 niz2 = x + niz2;
 i = i + 1;
}
Console.WriteLine(niz2);
```

```
string niz1 = "Tantadruj ", niz2 = "";
int n = niz1.Length, i = 3;
while(i < n)
{
 niz2 = niz2 + niz1[i];
 i = i + 1;
}
niz2 = niz2 + niz1[0] + niz1[1] + niz1[2];
Console.WriteLine(niz2);
```

---

# Vaja

---

Kaj izpiše naslednja metoda:

```
string a = "moje bojno polje";
Write(a.IndexOf("oj"));
```

- a) 6
  - b) 7
  - c) 1
  - d) 2
  - e) nič od navedenega, ampak \_\_\_\_\_
-

# Vaja

---

- Napišite program Okvir, ki bo zahteval vnos niza. Nato ga bo izpisal na zaslon v "okvirju".

*Primer:*

Vnesi niz: Mladost je norost, skače čez jarke, kjer je most.

```

|Mladost je norost, skače čez jarke, kjer je most.|

```

- Sestavite program, ki bo zahteval vnos besede. Program naj to besedo izpiše na zaslon tako, da bo beseda izpisana v obliki trikotnika.

- *Primer:* Vnesena beseda je JEZIK

JEZIK

JEZI

JEZ

JE

J

---