Nizi

`Tip String`

String

- `String ime = "Matija";`
- Zaporedje znakov med "
- Pozor: `String` in NE `string`
- Tehnično je `String` objekt in ne običajni tip, kot npr. `int` ali `boolean`.

Pozdrav

```
import java.applet.*;
import java.awt.*;
import javax.swing.JOptionPane;

public class Pozdrav extends Applet
{
   public void init()
   {
     String ime;
     String pozdrav;

     ime = JOptionPane.showInputDialog("Vnesi svoje ime");
     pozdrav = "Pozdravljen " + ime + "!";
     JOptionPane.showMessageDialog(null, pozdrav);
   }
}
```

Pozdrav.java

Stikanje nizov

- + stakne dva niza
- Brez "dodatnih" presledkov
- "Matija" + "Lokar" → "MatijaLokar"
- Če en izraz NI niz, se ta pretvori v niz!

- Branje nizov
  - `import javax.swing.*;`
  - metoda `JOptionPane.showInputDialog`

Ime in priimek

```
import javax.swing.JOptionPane;

public class ImePriimek
{
   public static void main(String[] qay)
   {
     String ime, priimek, ime_priimek;
     ime = JOptionPane.showInputDialog("Vnesi svoje ime");
     priimek = JOptionPane.showInputDialog("Vnesi priimek");
     ime_priimek = ime + " " + priimek;
     JOptionPane.showMessageDialog(null, ime_priimek);
   }
}
```

ImePriimek.java

Stik

```
import java.applet.*;
import javax.swing.JOptionPane;

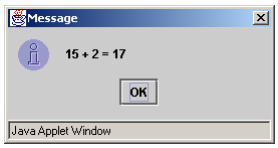
public class Stik extends Applet
{
   public void init()
   {
     String rezultat;
     rezultat = "15 + 2 = " + 15 + 2;
     JOptionPane.showMessageDialog(null, rezultat);
   }
}
```

Stik.java

Stik

```
import java.applet.*;
import javax.swing.JOptionPane;

public class Stik1 extends Applet
{
   public void init()
   {
     String rezultat;
     rezultat = "15 + 2 = " + (15 + 2);
     JOptionPane.showMessageDialog(null, rezultat);
   }
}
```



Stik1.java

Dolžina niza

- Dolžina niza: metoda `length()`
  - `priimek.length()`
  - `"matija".length()` → `6`
- Znak na i-tem mestu `charAt(i)`
  - `ime.charAt(1)`
  - `"Blabla".charAt(3)`
    - `b`
- Znake štejemo od 0 dalje!
  - `String ime = "Matija";`
  - `ime.charAt(1)` → `a`

Obrni niz

- Preberi niz in ga izpiši obrnjeno!
- `Matija` → `ajitaM`
- Zanka
  - Pregledamo vse znake v nizu (dolžina niza)
    - `while (i < niz.length())`
  - Dodajamo na začetek
    - `ob_niz = niz.charAt(i) + ob_niz;`
- `Obrni.java`
- (obrnemo in še enkrat obrnemo)
  - `Obrni2.java`

Premik niza

- Sestavi programček, ki niz premakne za 3 znake v levo.
- Namig - sestavi nov niz, ki ga dobiš tako, da praznemu nizu zaporedoma dodajaš drugi, tretji, ..., zadnji in prvi znak.

Tipične napake pri reševanju

- `niz.charAt(2)` NE obstaja, če je niz krajši kot 3 znake!
- Če je niz dolžine 0, 1 ali 3 je rezultat kar prvotni niz
- Če je niz dolžine 2, se znaka zamenjata

Rešitev

```
import javax.swing.*;

public class Nizi_dn
{
   public static void main(String[] atr)
   {
      String izpis, niz, prem_niz = "";
      int dol_niza, i;

      niz = JOptionPane.showInputDialog("Vnesi niz");
      dol_niza = niz.length();
      if ((dol_niza < 2) || (dol_niza == 3))
      { // niz ima le 0, 1 ali 3 znake
        prem_niz = niz;
      };
      if (dol_niza == 2)
      { // ce sta dva znaka, se pri premiku za 3 zamenjata
        prem_niz = "" + niz.charAt(1) + niz.charAt(0);
      }
```

Rešitev

```
if (dol_niza > 3)
{
   i = 3; // od cetrtega znaka dalje prepisemo
   while (i < dol_niza)
   {
      prem_niz = prem_niz + niz.charAt(i); // dodamo i-ti znak
      i = i + 1;
   }
   // dodamo še prve tri znake
   prem_niz = prem_niz + niz.charAt(0) + niz.charAt(1) + niz.charAt(2);
}
// Izpis rezultata
izpis =  niz + " = prvotni niz" + "\n" +
         prem_niz + " = premaknjeni niz" ;
JOptionPane.showMessageDialog(null, izpis,
   "Premik niza za 3 v levo", JOptionPane.INFORMATION_MESSAGE);
}
```

Nizi_dn.java

Rešitev (2) / substring

```
if (dol_niza > 3)
{ // resitev s pomocjo podniza
   prem_niz = niz.substring(3) +
              niz.substring(0,3);
}
// Izpis rezultata
izpis =  niz + " = prvotni niz" + "\n" +
         prem_niz + " = premaknjeni niz" ;
JOptionPane.showMessageDialog(null, izpis,
   "Premik niza za 3 v levo",
JOptionPane.INFORMATION_MESSAGE);
}
}
```

Nizi_dn_2.java

Podniz

- `substring(od, do)`:
  - `"Matija".substring(2,4)` → `"ti"`
  - vrne podniz od znaka z indeksom `od` do znaka z indeksom `do - 1`
- `substring(od)`:
  - `"Matija".substring(2)` → `"tija"`
  - vrne podniz od znaka z indeksom `od` do konca niza

Znaki (`char`)

- Tip `char`
  - `ker (character)`
  - Ne `čar`
- `char znak;`
- Enojni narekovaji
  - `znak = 'm';`
- Primerjanje
  - Ali je mala črka?
  - `((znak >= 'a') && (znak <= 'z'))`

Preštej številke v nizu

- Preberemo niz
- Pregledamo vsak znak
  - `znak = niz.charAt(i); // tekoci znak`
- Če je številka, povečamo števec za 1
  - ```
    if ((znak >= '0') && (znak <= '9')) // ce je stevka
    {
       koliko_stevk++; // povečanje za 1
    }
    ```
- Izpišemo rezultat
- Stevke.java

Preštej številke v nizu

```
import java.applet.*;
import javax.swing.JOptionPane;

public class Stevke extends Applet
{
   public void init()
   {
      char znak;
      String niz, rezultat;
      int i, dol_niza, koliko_stevk = 0;

      niz = JOptionPane.showInputDialog("Vnesi niz");
      dol_niza = niz.length();
```
```

Preštej številke v nizu

```
        i = 0;
        while (i < dol_niza)
        {
           znak = niz.charAt(i); // tekoci znak
           if ((znak >= '0') && (znak <= '9')) // ce je stevka
           {
              koliko_stevk++; // povečanje za 1
           }
           i++; // enako kot: i = i + 1;
        }
        rezultat = "V nizu " + niz + " je " +
                   koliko_stevk + " stevk.";
        JOptionPane.showMessageDialog(null, rezultat,
           "Spremenjeni niz", JOptionPane.INFORMATION_MESSAGE);
    }
}
```


Primerjanje nizov

- Če nize primerjamo neposredno (==), rezultat NI pravilen (pričakovan)
- Metodi
  - `equals`
  - `compareTo`
- `s1.equals("bla")` : ali je niz shranjen v `s1` enak nizu `bla` – rezultat `true` ali `false`
- `s1.equals(s2)` : ali je niz shranjen v `s1` enak nizu shranjenemu v `s2`

Primerjanje nizov

- Nize primerjamo leksikografsko (po abecedi)
  - Niz "Matija" je manjši kot niz "Mojca", ker sta prva znaka enaka, drugi znak pa je v prvem nizu ('a') manjši kot v drugem nizu ('o').
- `s1.compareTo("bla")` : vrne 0, če je niz shranjen v `s1` enak nizu `bla`, neg. število , če je niz v `s1` manjši od niza `"bla"` in poz. število, če je večji.
- `"matija".compareTo("mojca")` : vrne negativno število
- `equalsIgnoreCase(niz)`: enako kot `equals`, le da ne upošteva male/velike črke
- `"Matija".equalsIgnoreCase("matija")` vrne `true`, `"Matija".equals("matija")` pa je `false`.

Koristne metode v razredu String

- `toUpperCase`
  - pretvori vse male črke v velike
  - `"Mojca_21_b".toUpperCase` → `"MOJCA_21_B"`
  - Originalni niz se NE spremeni
- `toLowerCase`
- `indexOf`
  - Kje se v nizu začne podniz

indexOf

- `"Mojca je rekla joj".indexOf("oj")`
- Dobimo `1` – podniz `"oj"` se v nizu `"Mojca je rekla joj"` začne na mestu z indeksom `1`
- Če podniza ni, je rezultat `-1`
- `"Mojca je rekla joj".indexOf("aj")` je torej `-1`

indexOf

- Druga različica
- `"Mojca je rekla joj".indexOf("oj", 3)`
- Dobimo `16` – podniz začnemo iskati od indeksa 3 dalje!

Preštej samoglasnike

- Za vsak znak ugotovimo ali je samogasnik
- Sestavljeni pogoj
  - `(znak == 'a') || (znak = 'A') || (znak == 'e') || (znak = 'E') || (znak == 'I') || (znak = 'i') || (znak == 'o') || (znak = 'O') || (znak == 'u') || (znak = 'U')`
- Lahko z `indexOf`
  - `String samoglasniki = "AEIOUaeiou";`
  - `samoglasniki.indexOf(znak) != -1`

Preštej samoglasnike

```
String samoglasniki = "AEIOUaeiou";
niz = JOptionPane.showInputDialog("Vnesi poljubno besedilo!");
dol_niza = niz.length(); // dolzina niza
i = 0;
while(i < dol_niza) // pregledamo vse znake
{
  znak=niz.charAt(i);
  if (samoglasniki.indexOf(znak) != -1) // znak je samoglasnik
  { st_samo++;
  }
  i++;
}
rezultat = " V nizu " + niz + "je " + st_samo + " samoglasnikov ";
JOptionPane.showMessageDialog(null, rezultat, "Stevilo
   samoglasnikov v nizu", JOptionPane.PLAIN_MESSAGE);
```

Samoglasniki.java

Odstranjevanje presledkov

- V nizu zamenjaj vse večkratne presledke z enojnimi

Odstranjevanje presledkov

- Ponavljamo, dokler najdemo dvojne presledke
- Pogledamo, kje je dvojni presledek
- Če ga ni, končamo.
- Drugače sestavimo nov niz iz dela do dvojnega presledka, presledka in preostanka niza.
- Z iskanjem dvojnega presledka nadaljujemo dve mesti za položajem, kjer smo našli prejšnji dvojni presledek.

Odstrani presledke

```
int od_kje_iscem = 0, kje_je_dvojni;
String orig_niz, niz, rezultat;
boolean konec = false; // ali smo koncali z
                          dvojnimi presledki
niz = JOptionPane.showInputDialog(
       "Vnesi poljubno besedilo!");
orig_niz = niz; // kopija originalnega niza
```


Odstrani presledke

```
while(!konec) // dokler ni konec
 {
   kje_je_dvojni = niz.indexOf("  ", od_kje_iscem);
   if (kje_je_dvojni != -1)   // nasli smo dvojni presledek
   {
     niz = niz.substring(0, kje_je_dvojni) + " " +
           niz.substring(kje_je_dvojni + 2, niz.length());
           // iz niza odstranimo dvojni presledek
           // in ga nadomestimo z enojnim
     od_kje_iscem = kje_je_dvojni + 2;
           // kje nadaljujemo z iskanjem
   }
   else
   { konec = true;
   }
 }
```

Presledki.java

Preverjanje pravilnosti

- Tipično
  - Vsi smo srečni, če stvar deluje na testnem primeru
  - "xx⌂⌂xxxx⌂⌂xxxx" - "xx⌂xxxx⌂xxxx"
  - Videti je OK
  - "xx⌂⌂⌂xxxx⌂⌂xxxx" - "xx⌂xxxx⌂xxxx"
- Zadnji izpis je malček sumljiv
  - ostro oko
- Presledke zamenjamo z #
- PresledkiA.java

Preverjanje pravilnosti

- Preden nadaljujemo
- Premislimo, kakšni testi bi bili smiselni
- Niz z nekaj dvojnimi presledki(⌂)
  - "xx⌂⌂xxxx⌂⌂xxxx" - "xx⌂xxxx⌂xxxx"
- Niz s trojnimi presledki
  - "xx⌂⌂⌂xxxx⌂⌂xxxx" - "xx⌂xxxx⌂xxxx"
- Niz s sodim (> 2) številom presledkov
  - "xx⌂⌂⌂⌂⌂xxxx" - "xx⌂xxxx"
- Niz s presledki na začetku (liho/sodo)
  - "⌂⌂⌂xxxx⌂⌂xxxx" - "⌂xxxx⌂xxxx"
  - "⌂⌂⌂⌂xxxx⌂⌂xxxx" - "⌂xxxx⌂xxxx"
- Niz s presledki na koncu
  - "xxxx⌂⌂xxxx⌂⌂⌂" - "xxxx⌂xxxx⌂"
- Niz brez presledkov
  - "xxxx" - "xxxx"
- Niz z enojnimi presledki
  - "xxxx⌂" - "xxxx⌂xxx"
- Prazen niz
  - "" _ ""

Odstrani presledke (2)

- Pregledujemo znake
- Če naletimo na presledek
  - Če je tudi naslednji znak presledek
    - Ga odstranimo
    - Kaj, če smo že na koncu?
    - Preverimo prej!
      - Vrstni red vrednotenja logičnega izraza

Odstrani presledke

- `dol = niz.length();`
- `while (kje < dol)`
- `znak = niz.charAt(kje);`
- `if (znak == " ") && (kje + 1 < dol) && (niz.charAt(kje + 1) == " ")`
- `presledkiC.java`