

Uvod v programiranje

5. vaje

Kenny Štorgel

Fakulteta za informacijske študije Novo mesto

Novo mesto, februar 2019

Še dve funkciji nad nizi (**String**-i)

- ▶ Znake v spremenljivki `niz` tipa `String` spremenimo v male oziroma velike s klicem funkcij `toLowerCase()` oziroma `toUpperCase()`:

```
String maliNiz = niz.toLowerCase();  
String velikiNiz = niz.toUpperCase();
```

- ▶ Spremenljivko `crka` tipa `char` spremenimo v malo oziroma veliko s pomočjo statične funkcije razreda `Character`:

```
char malaCrka = Character.toLowerCase(crka);  
char velikaCrka = Character.toUpperCase(crka);
```

Diagram poteka

- ▶ Grafični prikaz algoritma
- ▶ Simboli so naslednji:

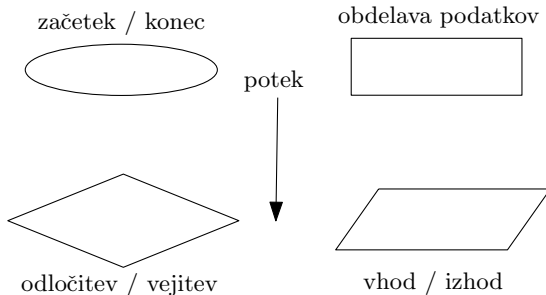
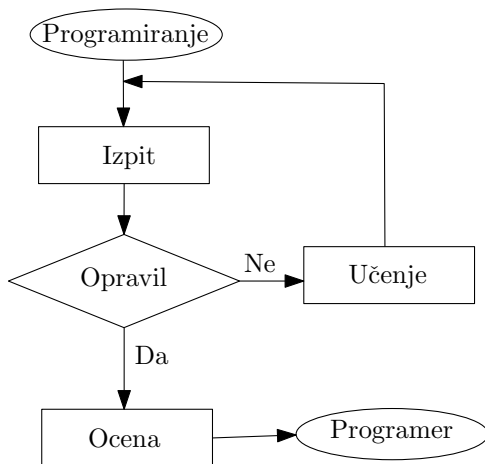


Diagram poteka - primer



Seštej števila od 0 do 11

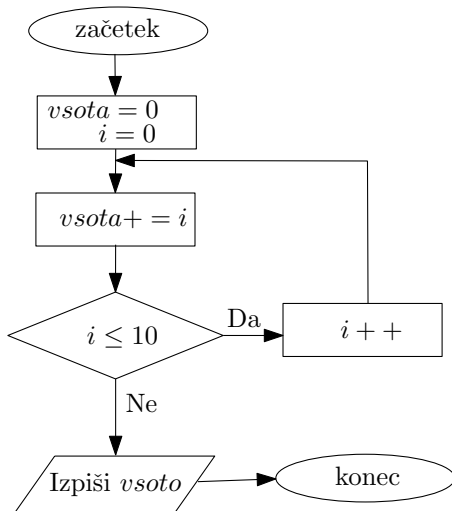


Diagram poteka - naloga

- ▶ **Naloga 1:** Izberite eno izmed nalog iz dosedanjih predavanj in narišite diagram poteka zanjo.

- **Naloga 4:** Na bankomatu lahko dvignemo največ 500 evrov v bankovcih po 10 ali 50 evrov. Narišite diagram poteka za postopek dviga določenega zneska. V primeru, ko je znesek previsok, naj bo odgovor: "Žnesek je previsok!", če znesek ni deljiv z 10, izpišemo "Žnesek ni ustrezen!", sicer pa se izpišeta števili izplačanih bankovcev po 10 in po 50 evrov. Bankomat od uporabnika zahteva vpis zneska, dokler ta ni izplačan. Pozor: števili bankovcev morate izračunati, število bankovcev po 50 evrov pa mora biti največje možno!

Na koncu s pomočjo diagrama poteka napišite še program.

- ▶ Samostojen del programa
- ▶ Dobi množico parametrov (lahko prazno)
- ▶ Vrne vrednost (lahko nič)

- ▶ Sintaksa:

```
vrstaMetode tipRezultata imeMetode(parametri) {  
    teloMetode  
}
```

- ▶ Vrsta metode: npr. `public`, `static`,...
- ▶ Tip rezultata: npr. `int`, `void`,...
- ▶ Vrednost vrnemo z uporabo ukaza `return`;
- ▶ Primer: metoda `main`

- ▶ **Naloga 2:** Napišite metodo, ki kot parameter prejme niz in znak ter vrne število pojavitev znaka v nizu. Nato metodo kličite v metodi `main`, parametra pa naj bosta argumenta programa.

- ▶ Izjema se zgodi ob nepričakovanem dogodku
- ▶ Npr. program pričakuje celo število, dobi pa črko
- ▶ Izjeme lahko "ujamemo" s stavkom `try-catch`
- ▶ Sintaksa:

```
try {  
    ukaziKiJihPreverjamo  
}  
catch (TipIzjeme imeIzjeme) {  
    ukaziPoUjetjuIzjeme  
}
```

- ▶ Metoda, ki bi lahko vrnila izjemo mora v deklaraciji vsebovati `throws` ali pa imeti "nevarno" kodo v `try-catch`

```
public class Izjema1 {  
    public static void main(String[] args) {  
  
        try {  
            int stevilo = Integer.parseInt(args[0]);  
            System.out.println("Argument je " + stevilo + ".");  
        }  
        catch(IndexOutOfBoundsException ind) {  
            System.out.println("Pozabil si vnesti argument!");  
        }  
        catch(NumberFormatException form) {  
            System.out.println("Argument ni število!");  
        }  
    }  
}
```

Izjeme - primer 2

```
import java.io.*;

public class Izjema2 {
    public static void main(String[] args) throws Exception {
        int stevilo = method();
        System.out.println("Argument je " + stevilo + ".");
    }

    public static int method() throws IOException {
        throw new IOException("Error!");
    }
}
```

Branje s tipkovnice

- ▶ Vnos parametrov med izvajanjem programa
- ▶ Potrebujemo ukaz `import java.io.*;`
- ▶ Prestrezali bomo 'napake' - `throws Exception`
- ▶ Tipkovnica je 'vhodni tok', odpremo ga z ukazom
`InputStreamReader vt = new
InputStreamReader(System.in);`
- ▶ Vrstico preberemo z ukazoma:
`BufferedReader vhod = new BufferedReader(vt);
String vrstica = vhod.readLine();`

Branje s tipkovnice - primer

```
import java.io.*;

public class BranjeSTipkovnice {
    public static void main(String[] args) throws Exception {

        InputStreamReader vt = new InputStreamReader(System.in);
        BufferedReader vhod = new BufferedReader(vt);

        System.out.print("Vnesi poljuben niz: ");
        String niz = vhod.readLine();

        System.out.print("Vnesi celo stevilo: ");
        int celo = Integer.parseInt(vhod.readLine());

        System.out.print("Vnesi realno stevilo: ");
        double realno = Double.parseDouble(vhod.readLine());

        System.out.print("Vnesi znak: ");
        char znak = vhod.readLine().charAt(0);

        System.out.println("\nVnesel si niz '" + niz + "', celo število '" + celo +
            "', realno število '" + realno + "' in znak '" + znak + "'.");
    }
}
```

- ▶ **Naloga 3:** Spremenite program iz naloge 2 tako, da boste niz in znak prebrali potem, ko bo program že zagnan.

Branje iz datotek

- ▶ Podatki so običajno zapisani v datotekah
- ▶ Poznati moramo pot do datoteke (oziroma samo ime, če je datoteka v isti mapi kot program)
- ▶ Pred branjem jo moramo odpreti:
`FileReader d = new FileReader("imeDat.txt");`
- ▶ Celotno vrstico preberemo z ukazom
`BufferedReader dat = new BufferedReader(d);`
`String vrstica = dat.readLine();`
- ▶ Na koncu moramo datoteko zapreti:
`dat.close();`

Branje iz datotek - primer

```
import java.io.*;

public class PreberiDatoteko {
    public static void main(String[] args) throws Exception {
        String ime = args[0];
        FileReader fr = new FileReader(ime);
        BufferedReader dat = new BufferedReader(fr);

        while(dat.ready()) {
            System.out.println(dat.readLine());
        }
        dat.close();
    }
}
```

Število samoglasnikov

- ▶ **Naloga 4:** V datoteko *Besedilo.txt* shranite nekaj odstavkov poljubnega besedila. Nato napišite program, ki bo z uporabo metode iz naloge 2 preštel vse samoglasnike v besedilu.

- ▶ Podatki so običajno zapisani v datotekah
- ▶ Tudi pred pisanjem moramo datoteko odpreti:
`FileWriter d = new FileWriter("imeDat.txt");`
- ▶ Uporabimo razred `PrintWriter` za zapisovanje
`PrintWriter dat = new PrintWriter(d);`
`dat.println("Poljuben niz znakov.");`
- ▶ Na koncu moramo datoteko zapreti:
`dat.close();`

Pisanje v datoteke - primer

```
import java.io.*;

public class ZapisiVDatoteko {
    public static void main(String[] args) throws Exception {
        String ime = "NovaDatoteka.txt";

        FileWriter fw = new FileWriter(ime);
        PrintWriter dat = new PrintWriter(fw);

        dat.println("Prva vrstica v datoteki.");
        dat.println();
        dat.println("Zadnja vrstica v datoteki.");
        dat.close();
    }
}
```

- ▶ **Naloga 5:** Napišite program, ki prebere niz iz datoteke *Besedilo.txt* in zamenja vse znake 'e' z znakom 'a' in jih zapišite v datoteko *BesediloBrezE.txt*.

5. domača naloga

- ▶ Napišite metodo `obrnjNiz`, ki kot parameter prejme niz, vrne pa njegovo obrnjeno vrednost. Primer `obrnjNiz("žirafa") = "afariž"`. Nato iz tipkovnice preberite ime datoteke *A*, v kateri imate besedilo ter ime datoteke *B*, v katero boste zapisovali nove podatke. Na koncu zapišite vsako izmed vrstic iz datoteke *A* v datoteko *B* z njeno obrnjeno vrednostjo.