

NAPOMENA: Vežbe podrazumevaju da je student ovladao teorijom iz dela Uvod u objektno orijentisano programiranje: "Ulazno izlazni podsistem" (22_ulaz-izlaz.ppt)

Ulaz, izlaz – tastatura

System je statička klasa koja se ne može instancirati. Sadrži tokove (strimove) koji nam omogućuju standardni ulaz (*System.in* - čitanje sa tastature), standardni izlaz (*System.out* - pisanje na ekran/konzolu) i izlaz greške (*System.err*).

Primer 1. Izlazni tok pisanje na ekran. Formatizovani ispis.

```
System.out.println("Koriscenje izlaznog toka stampanja na ekran:");
System.out.printf("Boolean ispis, vrednost %b je ispisana\n", true);
System.out.printf("Slovo ispis, vrednost %c je ispisana\n", 'A');
System.out.printf("Integer ispis, vrednost %d je ispisana\n", 10);
System.out.printf("Heksadecimal ispis, vrednost %H je ispisana\n", 15);
System.out.printf("Real ispis, vrednost %f je ispisana\n", 10.5);
System.out.printf("Real ispis, vrednost %e je ispisana\n\n", 123000000.123);
System.out.println("Formatizovani ispis u obliku %[širina][.preciznost]tip");
System.out.printf("Real ispis, vrednost %.2f je ispisana\n", 100.54321);
```

Primer 2. Čitanje sa tastature korišćenjem klase *Scanner* (jednostavan tekster koji omogućuje da se sa ulaza čitaju primitivni tipovi podataka i stringovi).

```
Scanner sc = new Scanner(System.in);
//citanje teksta
System.out.print("Unesi recenicu: ");
String text = sc.nextLine();
System.out.println("Ispis recenice: " + text);

//citanje promenljive integer
System.out.print("Unesi broj: ");
int ceoBroj = 0;
/*try{*/
    ceoBroj = sc.nextInt();
/*
}catch(Exception e){
    System.out.println("Niste uneli broj\n"+e);
    System.exit(0); // može a i ne mora exit(0)
}*/
System.out.println("Ispis broja: " + ceoBroj);

//citanje promenljive float
System.out.print("Unesi realni broj: ");
float realniBroj = sc.nextFloat();
System.out.println("Ispis realnog broja: " + realniBroj);

sc.close();
System.out.print("Program izvrsen");
```

Primer 3. Kreiranje metoda koje omogućuju siguran unos sa tastature korišćenjem klase *Scanner*.

```
// citanje teksta
static String ocitajTekst(Scanner sc) {
    String tekst = sc.nextLine();
    return tekst;
}

// citanje promenljive integer
static int ocitajBroj(Scanner sc) {
    int ceoBroj = 0;
    boolean notRead = true;
    do {
        if (sc.hasNextInt()) {
            ceoBroj = sc.nextInt();
            notRead = false;
        } else {
            if (!sc.nextLine().isEmpty()) {
                System.out.println("GREŠKA - Pokušajte ponovo: ");
            }
        }
        // očitava znak za novi red
        sc.nextLine();

        // drugi način
        /*try {
            ceoBroj = sc.nextInt();
            notRead = false;
        } catch (Exception e) {
            System.out.println("GREŠKA - Pokušajte ponovo: ");
        }
        sc.nextLine();*/
    } while (notRead);

    return ceoBroj;
}
```

Primer 4. Rad sa strimovima i čitačima – čitanje teksta sa tastature. Strimovi su mehanizmi koji omogućavaju unificiran pristup podacima. Omogućuju čitanje sa različitih izvora podataka ili pisanje na različita odredišta. Baziraju se na prenosu niza bajtova. Čitači dopunjuju strim za čitanje tako što omogućuju prenos Unicode stringova.

```
System.out.println("Nacin 1:");
//citanje teksta sa Strimom
InputStreamReader inStream =
    new InputStreamReader(System.in);
char [] charBuff = new char [100];
int brojKaraktera = 0;
try {
    System.out.print("Unesi tekst: ");
    brojKaraktera = inStream.read(charBuff);
} catch (IOException e) {
    e.printStackTrace();
}
System.out.print("Tekst je: ");
for (int i = 0; i < brojKaraktera; i++) {
    System.out.print(charBuff[i]);
}

System.out.println("\nNacin 2:");
//citanje teksta sa Citacem
BufferedReader inReader =
    new BufferedReader(new InputStreamReader(System.in));
String s = null;
try {
    System.out.print("Unesi tekst: ");
    s = inReader.readLine();
} catch (IOException e) {
    e.printStackTrace();
}
System.out.println("Tekst je: " + s);

try {
    inStream.close();
    inReader.close();
} catch (Exception e) {
    e.printStackTrace();
}
System.out.print("Program izvršen");
```

Zadaci

Zadatak 1. Po uzoru na primer 3, definisati klasu *Utility* koja sadrži metode koje omogućuju siguran unos primitivnih tipova sa tastature, korišćenjem klase *Scanner*. Unos treba da se ponavlja dokle god se ne unese očekivan tip vrednosti. Dakle, potrebno je definisati sledeće statičke metode:

- za očitavanje teksta
- za očitavanje celog broja

Ulaz, izlaz - tastatura

- za očitavanje realnog broja (32 bita)
- za očitavanje realnog broja (64 bita)
- za očitavanje 8-bitne vrednosti
- za očitavanje karaktera
- za očitavanje bool vrednosti
- za očitavanje odluke (da ili ne)