

5. laboratorijska vježba

Važna napomena: u svim zadacima potrebno je napisati Javadoc komentare za svaki razred te generirati dokumentaciju. Svi nazivi razreda, metoda i varijabli moraju biti na engleskom. Sav napisani programski kod mora biti napisan u skladu s konvencijama imenovanja varijabli, metoda i razreda (varijable i metode: malo početno slovo, camel-case; razredi i sučelja: veliko početno slovo, camel-case; konstante: uobičajeno sve veliko i razdvajanje podvlakom) te ostalim pozitivnim praksama (uključivo i korektno uvlačenje redaka; smisleno razdvajanje više različitih semantički grupiranih redaka praznim recima, pravilnim razmještajem otvorene i zatvorene vitičaste zagrade i slično). Za više informacija pogledajte <http://www.oracle.com/technetwork/java/codeconventions-150003.pdf>

Zadatak 1.

U nekom programu vodi se evidenciju o poslovnim partnerima. Svaki poslovni partner jednoznačno je određen OIB-om koji se, postava prilikom stvaranja objekta i više se ne smije mijenjati niti se smije dopustiti nadjačavanje njegova *gettera*. Prirodni komparator poslovnih partnera je po OIB-u. Poslovni partneri mogu biti osobe ili tvrtke i ne smije postojati neka treća vrsta poslovnog partnera. Osobe imaju ime i prezime, a tvrtke naziv. Svaka tvrtka ima kolekciju svojih zaposlenika, a pojedina osoba može raditi za više tvrtki. Prilikom ispisa osobe ispisuje se njeno ime, prezime i OIB, a prilikom ispisa tvrtke ispisuje se naziv tvrtke, OIB te popis njenih zaposlenika složenih po redoslijedu dodavanja u popis zaposlenika.

Dovršite klase za entitete iz gornjeg teksta, a zatim dopunite glavni program potrebnim kodom i/ili dodatnim klasama kako bi ispis bio isti kao u primjeru. U primjeru se partneri iz kolekcije ispisuje na način da se prvo ispisuju tvrtke sortirano silazno po broju zaposlenih, a zatim osobe sortirane po OIB-u. U slučaju da dvije tvrtke imaju jednak broj zaposlenih prvo se ispisuje ona čiji je OIB prije po leksikografskom poretku.

```
public static void main(String[] args) {
    Collection<PoslovniPartner> col = new TreeSet<>( new PoslovniPartnerComparator());
    Osoba o1 = new Osoba("42345678901", "Darko", "Darić");
    Osoba o2 = new Osoba("58855678121", "Bianka", "Banić");
    Osoba o3 = new Osoba("28964789143", "Ivan", "Ivić");
    Osoba o4 = new Osoba("15678842125", "Ante", "Antić");
    Osoba o5 = new Osoba("35678842125", "Mate", "Matić");

    Tvrtka t1 = new Tvrtka("66879845127", "Meho und jaranen");
    Tvrtka t2 = new Tvrtka("45174579172", "Otraj-dotraj");
    Tvrtka t3 = new Tvrtka("25174579172", "Kupi-prodaj");

    t1.getZaposlenici().add(o1);
    t1.getZaposlenici().add(new Osoba("42345678901", "Darko", "Darić"));
    t1.getZaposlenici().add(o1);
    t1.getZaposlenici().add(o4);

    t2.getZaposlenici().add(o3);
    t2.getZaposlenici().add(o2);
    t2.getZaposlenici().add(o1);

    t3.getZaposlenici().add(o2);
    t3.getZaposlenici().add(o3);
    t3.getZaposlenici().add(o4);

    col.add(t1);
    col.add(o5);
    col.add(t2);
    col.add(t3);

    for(PoslovniPartner p : col ){
        System.out.println(p);
    }
}
```

ISPIS

Kupi-prodaj 25174579172
 Bianka Banić - 58855678121
 Ivan Ivić - 28964789143
 Ante Antić - 15678842125

Otraj-dotraj 45174579172
 Ivan Ivić - 28964789143
 Bianka Banić - 58855678121
 Darko Darić - 42345678901

Meho und jaranen 66879845127
 Darko Darić - 42345678901
 Ante Antić - 15678842125

Mate Matić - 35678842125

Zadatak 2.

Klasa *IntegerSet* predstavlja iterabilni skup elemenata tipa *MyNumber* pri čemu *MyNumber* predstavlja cijeli broj koji omogućava iteriranje po znamenkama broja (od najznačajnije prema najmanje značajnom mjestu). Dopišite kôd klase *IntegerSet* tako da se omogući navedena funkcionalnost i da priloženi glavni program ima ispis kao u donjem okviru.

```
public static void main(String[] args) {
    MyNumber number = new MyNumber(123);
    for(Integer i : number){
        System.out.format("%d ", i);
    }
    System.out.println();
    number = new MyNumber(-4781);
    for(Integer i : number){
        System.out.format("%d ", i);
    }
    System.out.println();

    IntegerSet set = new IntegerSet();
    set.add(157);
    set.add(12345);
    set.add(-3005);

    for(MyNumber numbers : set){
        System.out.format("Broj %s ima znamenke:", numbers.toString());
        for(Integer i : numbers){
            System.out.format(" %d", i);
        }
        System.out.println();
    }
}
```

ISPIS PROGRAMA:

```
1 2 3
4 7 8 1
Broj -3005 ima znamenke: 3 0 0 5
Broj 157 ima znamenke: 1 5 7
Broj 12345 ima znamenke: 1 2 3 4 5
```