

## 4. Četvrta laboratorijska vježba

### 4.1. ZBIRKE, GENERIČKO PROGRAMIRANJE I LAMBDA IZRAZI U JAVI

Svrha laboratorijske vježbe je korištenje različitih zbirki za sortiranje podataka, generičkog programiranja u Javi zbog postizanja veće razine proširivosti Java aplikacije, zajedno s korištenjem lambda izraza i uspoređivanjem količine programskog koda koji je potrebno napisati u slučaju s korištenjem ili bez korištenja lambda izraza.

### 4.2. ZADATAK ZA PRIPREMU

Nastaviti razvoj aplikacije iz treće laboratorijske vježbe i implementirati korištenje zbirki podataka, generičkog programiranja i lambda izraza. Potrebno je implementirati sljedeće korake:

1. Kopirati projekt iz treće laboratorijske vježbe i povećati mu indeks vježbe s „3“ na „4“.
2. Unutar paketa **hr.java.vjezbe.entitet** uvesti enumeraciju **RadSenzora** koja će sadržavati tri elementa: „STREAMING“, „PING“ i „OSTALO“.
3. Unutar paketa **hr.java.vjezbe.entitet** uvesti enumeraciju **VrstaMjesta** koja će sadržavati tri elementa: „GRAD“, „SELO“ i „OSTALO“.
4. U klase **Mjesto** i **Senzor** ugraditi navedene enumeracije kao varijable i dopuniti klase dodavanjem „get“ i „set“ metoda.
5. U svakom djelu unutar klase Glavna gdje se od korisnika traži unos mjesta i senzora dopuniti programski kod na način da se traži i unos enumeracija iz proteklih koraka. Ako korisnik ne unese neku od ponuđenih vrijednosti, odabir je potrebno postaviti na „OSTALO“.

Primjer programskog koda koji omogućava odabir jednog od dostupnih rada senzora i odabir „OSTALO“ ako je odabran rad senzora koji ne postoji može izgledati ovako:

```
for (int i = 0; i < RadSenzora.values().length - 1; i++) {
    System.out.println((i + 1) + ". " + RadSenzora.values()[i]);
}

Integer redniBrojSenzora = null;

while (true) {
    System.out.print("Odabir rada senzora >> ");
    try {
        redniBrojSenzora = scanner.nextInt();
        break;
    } catch (InputMismatchException ex) {
        System.out.println("Neispravan unos!");
        logger.error("Neispravan unos rada senzora!", ex);
    }
}
```

```
if (redniBrojSenzora >= 1 && redniBrojSenzora <
RadSenzora.values().length) {
    radSenzora = RadSenzora.values()[redniBrojSenzora - 1];
} else {
    radSenzora = RadSenzora.OSTALO;
}
```

6. Sva polja koja se koriste u programu potrebno je zamijeniti listama i implementirati logiku da program radi ispravno. Kod mjernih postaja potrebno je pripaziti na metodu „dohvatiSenzore“ koju je potrebno implementirati pomoću lambda izraza. Mjerne postaje se također moraju nalaziti u listi.
7. Unutar objekta države potrebno je kreirati generičku listu županija. Za dohvat liste potrebno je kreirati „get“ metodu. Listu je potrebno inicijalizirati (napraviti novi objekt) u konstruktoru. Prilikom unosa podataka potrebno je puniti županije u državu. Ako je unesena ista država županije (usporediti nazive), potrebno je unutar županije dodati postojeću referencu države, a u klasi država dodati novo unesenu županiju. Sve usporedbe je potrebno napraviti pomoću lambda izraza.
8. Identičnu logiku iz prošlog koraka potrebno je upotrijebiti sa slučajem županija koje moraju imati listu mjesta i mjesta koje moraju imati listu mjernih postaja. Primjer programskog koda za prošlu i ovu točku slijedi u nastavku:

```
private static Zupanija dohvatiZupaniju(Scanner scanner) {
    Drzava drz = dohvatiDrzavu(scanner);
    System.out.println("Unesite naziv županije:");
    String naziv = scanner.nextLine();
    Optional<MjernaPostaja> postaja = postaje.stream().filter(p ->
p.getMjesto().getZupanija().getNaziv().equals(naziv)).findFirst();

    if(postaja.isPresent())
        return postaja.get().getMjesto().getZupanija();

    Zupanija zup = new Zupanija(naziv, drz);
    drz.getZupanije().add(zup);
    return zup;
}
```

9. Kreirati novi paket **hr.java.vjezbe.sortiranje**
10. Unutar paketa **hr.java.vjezbe.sortiranje** kreirati klasu **ZupanijaSorter** koji implementira sučelje **Comparator** i definira logiku za sortiranje objekata klase **Zupanija** prema nazivu.
11. U klasi glavna na kraju je potrebno ispisati sve županije sortirane pomoću klase **ZupanijaSorter**. Ispis ne smije sadržavati duplikate.
12. Prilagoditi dosadašnji ispis da se ispisuju vrijednosti enumeracija kreiranih na početku vježbe.
13. Korištenjem zbirke tipa "Map" implementirati logiku koja će za svako mjesto ispisivati koji se senzori u njemu nalaze (npr. senzor vlage ili temperature).

### Primjer izvođenja programa:

```
Unesite prvu mjernu postaju:  
Unesite naziv mjerne postaje:  
Maksimir  
Unesite naziv države:  
Hrvatska  
Unesite površinu države:  
56594  
Unesite naziv županije:  
Grad Zagreb  
Unesite naziv mjesta:  
Zagreb  
1. GRAD  
2. SELO  
Odabir >> 3  
Unesite Geo koordinatu X:  
5  
Unesite Geo koordinatu Y:  
50  
Unesite elektroničku komponentu za senzor temperature:  
nepoznato  
Unesite vrijednost senzora temperature:  
25  
1. STREAMING  
2. PING  
Odabir rada senzora >> 3  
Unesite vrijednost senzora vlage:  
80  
1. STREAMING  
2. PING  
Odabir rada senzora >> 2  
Unesite veličinu senzora brzine vjetra:  
veliki  
Unesite vrijednost senzora vjetra:  
20  
1. STREAMING  
2. PING  
Odabir rada senzora >> 1  
Unesite drugu mjernu postaju:  
Unesite naziv mjerne postaje:  
Zavižan  
Unesite naziv države:  
Hrvatska  
Unesite naziv županije:  
Ličko-senjska  
Unesite naziv mjesta:  
Zavižan  
1. GRAD  
2. SELO  
Odabir >> 2  
Unesite Geo koordinatu X:
```

```
1
Unesite Geo koordinatu Y:
2
Unesite elektroničku komponentu za senzor temperature:
DTH11
Unesite vrijednost senzora temperature:
11
1. STREAMING
2. PING
Odabir rada senzora >> 2
Unesite vrijednost senzora vlage:
20
1. STREAMING
2. PING
Odabir rada senzora >> 1
Unesite veličinu senzora brzine vjetra:
veliki
Unesite vrijednost senzora vjetra:
5
1. STREAMING
2. PING
Odabir rada senzora >> 2
Unesite treću mjernu postaju:
Unesite naziv radio sondažne mjerne postaje:
Zemunik
Unesite visinu radio sondažne mjerne postaje:
50
Unesite naziv države:
Hrvatska
Unesite naziv županije:
Zadarska županija
Unesite naziv mjesta:
Zadar
1. GRAD
2. SELO
Odabir >> 1
Unesite Geo koordinatu X:
5
Unesite Geo koordinatu Y:
5
Unesite elektroničku komponentu za senzor temperature:
X2X
Unesite vrijednost senzora temperature:
28
1. STREAMING
2. PING
Odabir rada senzora >> 1
Unesite vrijednost senzora vlage:
60
1. STREAMING
2. PING
Odabir rada senzora >> 1
```

Unesite veličinu senzora brzine vjetra:

mali

Unesite vrijednost senzora vjetra:

30

1. STREAMING

2. PING

Odabir rada senzora >> 1

-----  
Naziv mjerne postaje: Maksimir

Postaja se nalazi u mjestu Zagreb (OSTALO), županiji Grad Zagreb, državi Hrvatska

Točne koordinate postaje su x:5 y:50

Vrijednosti senzora postaje su:

Vrijednost: 80 % vlage zraka, Način rada: PING

Veličina: veliki, vrijednost: 20 km/h, Način rada: STREAMING

Komponenta: nepoznato, vrijednost: 25 °C, Način rada: OSTALO

-----  
Naziv mjerne postaje: Zavižan

Postaja se nalazi u mjestu Zavižan (SELO), županiji Ličko-senjska, državi Hrvatska

Točne koordinate postaje su x:1 y:2

Vrijednosti senzora postaje su:

Vrijednost: 20 % vlage zraka, Način rada: PING

Veličina: veliki, vrijednost: 5 km/h, Način rada: PING

Komponenta: DTH11, vrijednost: 11 °C, Način rada: STREAMING

-----  
Naziv mjerne postaje: Zemunik

Postaja je radio sondažna

Visina radio sondažne mjerne postaje:50

Postaja se nalazi u mjestu Zadar (GRAD), županiji Zadarska županija, državi Hrvatska

Točne koordinate postaje su x:5 y:5

Vrijednosti senzora postaje su:

Vrijednost: 60 % vlage zraka, Način rada: STREAMING

Veličina: mali, vrijednost: 30 km/h, Način rada: STREAMING

Komponenta: X2X, vrijednost: 28 °C, Način rada: STREAMING

Ispis sortiranih županija:

Grad Zagreb

Ličko-senjska

Zadarska županija

U mjestu Zagreb su sljedeći senzori:

Senzor vlage

Senzor brzine vjetra

Senzor temperature

U mjestu Zavižan su sljedeći senzori:

Senzor vlage

Senzor brzine vjetra  
Senzor temperature

U mjestu Zadar su sljedeći senzori:

Senzor vlage  
Senzor brzine vjetra  
Senzor temperature

#### NAPOMENE:

1. Usporedbu jednakosti Stringova obavljati korištenjem metode „equals“.
2. Kod ponovnog unošenja vrijednosti nakon pogrešnog unosa koristiti metodu „nextLine“ u klasi „Scanner“ kako bi se izbjeglo iskorištavanje neispravno unesene vrijednosti.