Dodatne naloge – Vsebovalniki

Splošna navodila

Pri nalogah 4 in 6 lahko svojo rešitev preverite z množico vhodnih in pripadajočih izhodnih datotek, pri ostalih nalogah pa z množico testnih razredov in pripadajočih izhodnih datotek.

1 Zrcalna slika seznama

Naloga

Napišite razred `ZrcalnaSlikaSeznama` z metodo

```
public static <T> List<T> zrcalnaSlika(List<T> seznam)
```

ki vrne nov seznam, ki vsebuje elemente podanega seznama v obratnem vrstnem redu. Na primer, za seznam [30, 20, 50, 40] naj metoda vrne seznam [40, 50, 20, 30].

2 Sploščitev seznama

Naloga

Napišite razred `SploscitevSeznama` z metodo

```
public static <T> List<T> splosci(List<List<T>> seznami)
```

ki izdela in vrne nov seznam, v katerem so po vrsti nanizani elementi notranjih seznamov podanega seznama. Na primer, za seznam [[30], [20, 50, 10], [30, 40]] naj metoda vrne seznam [30, 20, 50, 10, 30, 40].

3 Presek množic

Naloga

Napišite razred `PresekMnozic` z metodo

```
public static <T> Set<T> presek(Collection<Set<T>> mnozice)
```

ki vrne presek množic iz podane zbirke. Na primer, za zbirko $\langle\{2, 4, 5, 7, 8, 9\}, \{4, 6, 7, 8, 9, 10\}, \{1, 2, 4, 5, 7, 8\}\rangle$ naj metoda vrne množico $\{4, 7, 8\}$.

4 Manjkajoče besede

Naloga

Napišite program, ki prebere dve množici besed in v leksikografskem vrstnem redu izpiše vse besede, ki nastopajo v prvi množici, ne pa tudi v drugi. Vsako besedo naj izpiše samo po enkrat.

Vhod

Vhod je sestavljen iz štirih vrstic. Prva podaja število besed v prvi množici ($m \in [1, 1000]$), druga vsebuje m besed, ki pripadajo prvi množici, tretja podaja število besed v drugi množici ($n \in [1, 1000]$), četrta pa vsebuje n besed, ki pripadajo drugi množici. V vrsticah, ki podajata množici besed, so besede med seboj ločene s presledkom. Besede so sestavljene zgolj iz velikih in malih črk angleške abecede, števk in podčrtajev. Nobena beseda ni daljša od 100 znakov.

Izhod

Izpišite iskane besede. Vsaka beseda naj bo izpisana v svoji vrstici.

Testni primer 1

Vhod:

```
8
medtem ko je programski jezik java objektno usmerjen
6
je programski jezik C proceduralno usmerjen
```

Izhod:

```
java
ko
medtem
objektno
```

5 Potenčna množica

Naloga

Napišite razred `PotencnaMnozica` z metodo

```
public static <T> Set<Set<T>> potencna(Set<T> mnozica)
```

ki vrne množico vseh podmnožic podane množice. Na primer, za množico {1, 2, 3} naj metoda vrne množico {{}, {1}, {2}, {3}, {1, 2}, {1, 3}, {2, 3}, {1, 2, 3}}.

6 Pogostost besed

Naloga

Napišite program, ki prebere zaporedje besed in izpiše seznam besed in njihovih pogostosti. Seznam naj bo urejen po padajoči pogostosti, enako pogoste besede pa naj se izpišejo v leksikografskem vrstnem redu.

Vhod

Vhod je sestavljen iz ene same vrstice, ta pa vsebuje besede, ločene s presledkom. Besede so sestavljene zgolj iz velikih in malih črk angleške abecede, števk in podčrtajev.

Izhod

Za vsako različno besedo v vhodnem besedilu izpišite vrstico sledeče oblike:

beseda `(`*pogostost*`)`

Testni primer 1

Vhod:

```
marko skace marko skace po zeleni trati aj aj aj aj aj po zeleni trati
```

Izhod:

```
aj (5)
marko (2)
po (2)
skace (2)
trati (2)
zeleni (2)
```

7 Obrat slovarja

Naloga

Napišite razred `ObratSlovarja` z metodo

```
public static<K, V> Map<V, Set<K>> obrni(Map<K, V> slovar)
```

ki vrne nov slovar, ki objekt v preslika v množico vseh objektov k, ki jih podani slovar preslika v objekt v. Na primer, za slovar s preslikavami

Slovenija $\mapsto$ 4,
Avstrija $\mapsto$ 8,
Italija $\mapsto$ 6,
Hrvaška $\mapsto$ 5,
Češka $\mapsto$ 4,
Slovaška $\mapsto$ 5,

Srbija $\mapsto$ 8,
Belgija $\mapsto$ 4

naj metoda vrne slovar s preslikavami

4 $\mapsto$ {Slovenija, Češka, Belgija},
5 $\mapsto$ {Hrvaška, Slovaška},
6 $\mapsto$ {Italija},
8 $\mapsto$ {Avstrija, Srbija}.

8 Slovar dvobojev

Naloga

Razred `SlovarDvobojev` dopolnite z metodo

```
public static NavigableMap<String, NavigableMap<String, Integer>>
        partije2slovar(List<Partija> partije)
```

ki na podlagi seznama šahovskih partij izdela in vrne slovar, ki igralca a preslika v slovar, ki igralca $b \neq a$ preslika v skupno število točk, ki jih je igralec a zbral v dvobojih z igralcem b. Igralci so enolično določeni z imeni. Zmaga prinese 2 točki, remi 1 točko, poraz pa 0 točk.[1] Slovar naj vsebuje vnos za vsak par igralcev a in b, kjer je $a \neq b$. Če igralca a in b nista odigrala nobene partije, naj izraz s`.get(`a`).get(`b`)` potemtakem ne sproži izjeme, ampak naj vrne 0. Pri vseh slovarjih (pri zunanjem in notranjih) naj bodo ključi leksikografsko urejeni.

Razred `Partija` je definiran takole:

```
public class Partija {
    private String beli;    // ime igralca z belimi figurami
    private String crni;    // ime igralca s črnimi figurami
    private int izid;       // izid z vidika belega (2: zmaga, 1: remi, 0: poraz)

    public Partija(String beli, String crni, int izid) {
        this.beli = beli;
        this.crni = crni;
        this.izid = izid;
    }
}
```

Na primer, če je Ana premagala Bojana, Bojan Cvetko, Cvetka pa je remizirala z Ano, naj metoda vrne slovar s sledečimi preslikavami:

Ana $\mapsto$ {Bojan $\mapsto$ 2, Cvetka $\mapsto$ 1},
Bojan $\mapsto$ {Ana $\mapsto$ 0, Cvetka $\mapsto$ 2},
Cvetka $\mapsto$ {Ana $\mapsto$ 1, Bojan $\mapsto$ 0}.

[1] V šahu zmaga dejansko prinese eno točko, remi pa pol točke, vendar pa se želimo izogniti decimalkam.