

Tretji izpitni rok pri predmetu Programiranje 1

27. avgust 2020

Oddajte datoteke Prva.java, Druga.java, Tretja.java in Cetrt.a.java. Testirate jih lahko takole:

(1) tj.exe Prva.java . . (2) tj.exe (3) tj.exe (4) tj.exe

- ① Število je *obilno*, če je manjše od vsote svojih pravih deliteljev (deliteljev brez števila samega). Na primer, število 12 je obilno, ker velja $12 < 1 + 2 + 3 + 4 + 6$. Napišite program, ki prebere število $n \in [1, 1000]$ in izpiše najmanjše obilno število, ki je enako najmanj n .

Primer (vhod/izhod):

13

18

- ② Nepravokotna tabela r tipa `int[][]` podaja raven delcev PM10 po posameznih dneh koledarskega leta. Element $r[m][d]$ podaja raven, izmerjeno na dan z indeksom d v mesecu z indeksom m (npr. element $r[1][6]$ se nanaša na 7. februar). Dolžina vrstice z indeksom m je enaka številu dni v mesecu z indeksom m .

Napišite sledeči metodi:

- [50%] `public static int stPresezkov(int[][] r, int meja):`

Vrne skupno število dni (v celotnem letu), v katerih je raven delcev presegla vrednost *meja*.

- [50%] `public static int nedeljskoPovprecje(int[][] r):`

Vrne navzdol zaokroženo povprečje ravni delcev ob nedeljah, če vemo, da je bil 1. januar ponedeljek.

- ③ Podana sta razred za predstavitev gozda in hierarhija razredov za predstavitev dreves:

```
class Gozd {
    private Drevo[] drevesa;
}

abstract class Drevo {
    private int x, y;    // koordinati drevesa (v intervalu [0,1000])
    public abstract String vrsta();
}

abstract class Listavec extends Drevo { ... }
abstract class Iglavec extends Drevo { ... }

class Hrast extends Listavec { ... }
class Bukev extends Listavec { ... }
class Breza extends Listavec { ... }
```

```
...
class Smreka extends Iglavec { ... }
class Jelka extends Iglavec { ... }
class Bor extends Iglavec { ... }
...
```

Metoda **vrsta** je v abstraktnih razredih hierarhije abstraktna, v neabstraktnih pa vrne niz, ki je enak imenu pripadajočega razreda. Na primer, v razredu **Hrast** vrne niz **Hrast**.

Rešite sledeče naloge:

- [32%] V razredu **Drevo** napišite metodo

```
public int d2(Drevo drevo),
```

 ki vrne kvadrat evklidske razdalje (tj. $(x_2 - x_1)^2 + (y_2 - y_1)^2$) med drevesoma **this** in **drevo**.
- [34%] V razredu **Gozd** napišite metodo

```
public boolean jeListnat(),
```

 ki vrne **true** natanko tedaj, ko so vsa drevesa v gozdu **this** listavci.
- [34%] V razredu **Gozd** napišite metodo

```
public Drevo najblizjeIstovrstno(Drevo drevo),
```

 ki med drevesi v gozdu **this**, ki so iste vrste kot drevo **drevo**, poišče (in vrne) tisto, ki ima najmanjšo evklidsko razdaljo do drevesa **drevo**. Drevo **drevo** je eno od dreves v gozdu; če je edino svoje vrste, naj metoda vrne **null**.

④ Podan je sledeči vmesnik:

```
interface Generator {
    public int naslednji();
}
```

Napišite metodo

```
public static int stKlicev(Generator gen, int k),
```

ki vrne podatek o tem, kolikokrat moramo metodo **naslednji** zaporedoma poklicati, da bomo **k**-tič dobili isti rezultat. Na primer, če prvih 10 klicev metode **naslednji** proizvede zaporedje

5, -6, 1263, -6, 1263, -42, -6, 1263, 5, 1263

in če je **k** enak 3, potem mora metoda vrniti 7, saj sedmi klic metode **naslednji** tretjič vrne isto število (v našem primeru je to -6).

Namig: morda vam bo prišel prav slovar (vmesnik **Map**, razred **HashMap**) in njegove metode **containsKey**, **get** in **put**.