

Tretji izpitni rok pri predmetu Programiranje 1

23. avgust 2021

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

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

- ① Na vходу sta podani celi števili $d \in [1, 6]$ in $v \in [1, 9d]$, ločeni s presledkom. Napišite program, ki izpiše najmanjše d -mestno število, v katerem je vsota števk enaka v .

Primer (vhod/izhod):

4 23

1499

- ② Miha si je omislil *Nekoliko nenavaden nosilec nadležne navlake* (NNNNN). To je predalnik, razdeljen na m »vrstic«, visokih h_1, h_2, \dots, h_m , in n »stolpcev«, širokih w_1, w_2, \dots, w_n . Vanj želi shraniti čimveč od svojih k pravokotnih predmetov, velikih $a_1 \times b_1, a_2 \times b_2, \dots, a_k \times b_k$. Problema se loteva tako, da na običajen način »potuje« po predalih (po vrsticah od zgoraj navzdol, znotraj posameznih vrstic pa od leve proti desni) in vsakokrat v trenutni predal shrani predmet z največjo ploščino, ki ga je tja še mogoče vstaviti; če je takih predmetov več, izbere prvega od njih. Predmete lahko obrača za 90 stopinj. S postopkom zaključi, ko v trenutni predal ne more več shraniti nobenega preostalega predmeta. V primeru na spodnji sliki lahko v NNNNN shrani tri predmete, saj v prvi predal v drugi vrstici ne more več ničesar vstaviti.

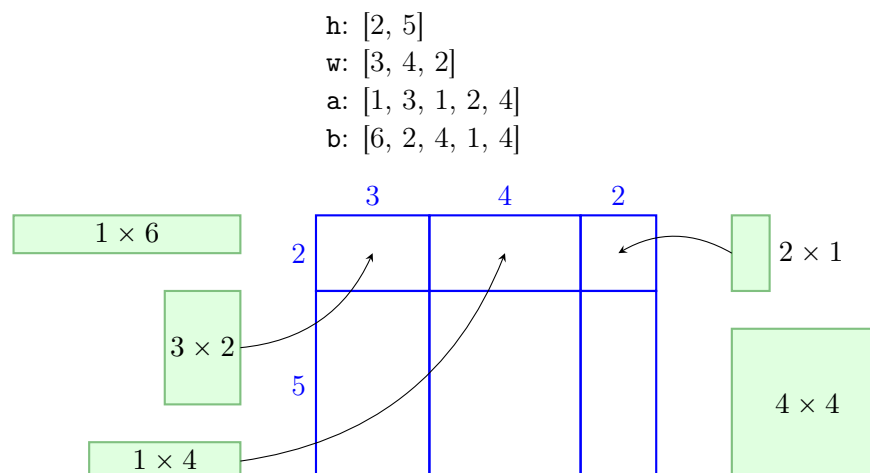
V razredu Druga dopolnite metodo

```
public static int steviloShranjenih(int[] h, int[] w, int[] a, int[] b),
```

ki vrne število predmetov, ki jih Miha z opisanim postopkom shrani v svoj NNNNN. Tabela h hrani števila h_i , tabela w hrani števila w_i itd. Vsaka tabela vsebuje najmanj enega in največ 200 elementov, vsi elementi pa pripadajo intervalu $[1, 1000]$. Tabeli a in b sta enako dolgi.

V metodi lahko po želji spreminjate elemente vhodnih tabel.

V 60% skritih testnih primerov so vsi predmeti kvadratne oblike, v polovici od njih pa so vsi enako veliki.



- ③ Sledeči razredi so definirani kot statični notranji razredi v razredu Tretja:

```
abstract class Zival {}
class Pes extends Zival {}
class Macka extends Zival {}
class RodovniskiPes extends Pes {
    private String pasma;
    private RodovniskiPes oče;    // null, če je this začetnik rodbine
    private RodovniskiPes mati;   // null, če je this začetnik rodbine
    // v vseh testnih primerih velja: oče == null ⇔ mati == null
}
```

Rešite sledeče naloge:

- [34%] V razredu Tretja dopolnite metodo

```
public static int prestej(RodovniskiPes[] psi, String pasma),
```

ki vrne število psov podane pasme v podani tabeli.
- [34%] Razrede dopolnite tako, da bo metoda

```
public static void koncert(Zival[] zivali) {
    for (Zival zival: zivali) {
        System.out.println(zival.oglasise());
    }
}
```

za vsakega psa izpisala hov, za vsako mačko pa mijav.
- [32%] V razredu RodovniskiPes dopolnite metodo

```
public boolean preveri(),
```

ki vrne `true` natanko v primeru, če velja eno od sledečega: (1) pes `this` je začetnik rodbine; (2) oče in mati sta oba iste pasme kot pes `this`, poleg tega pa tudi vsak od njiju izpolnjuje pogoj (1) ali (2).

- ④ Razred Cetrti vsebuje sledeči statični notranji razred:

```
class Prostor {
    private int nadstropje;
    private int kvadratura;
    private String namembnost;    // npr. "pisarna", "kuhinja" ...
}
```

Dopolnite sledeče metode (po potrebi lahko dopolnite tudi razred Prostor):

- [34%]

```
public static Set<String> namembnosti(List<Prostor> prostori)
```

Vrne množico vseh namembnosti, ki jih imajo prostori v podanem seznamu.
- [34%]

```
public static void uredi(List<Prostor> prostori)
```

Uredi seznam prostorov po naraščajočih nadstropjih, prostore v istem nadstropju pa po padajočih kvadraturah.
- [32%]

```
public static Map<Integer, Map<String, Integer>>
    statistika(List<Prostor> prostori)
```

Vrne slovar, ki število n preslika v slovar, ki niz s preslika v število prostorov iz seznama `prostori`, ki se nahajajo v nadstropju n in imajo namembnost s .