

Tretji izpitni rok pri predmetu Programiranje 1

1. september 2023

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 vhodu je podana zamaknjena piramida zvezdic, vendar pa je vsak presledek predstavljen s številom 0, vsaka zvezdica s številom 1, vsak prelom vrstice pa s številom 2. Na primer, piramida na desni (višina = 4, zamik = 3) je zapisana takole (`test01.in`):

```
0 0 0 0 0 0 1 2 0 0 0 0 0 1 1 1 2 0 0 0 0 1 1 1 1 1 2 0 0 0 1 1 1 1 1 1 1 2
```

```
UUUUUU*
UUUUU***
UUUU*****
UUU*****
|←→|
zamik
```

Napišite program (`Prva.java`), ki v prvi vrstici izpiše višino, v drugi pa zamik piramide. V gornjem primeru bi moral program izpisati sledeče (`test01.out`):

```
4
3
```

Vhod vsebuje največ 10^4 števil. Števila na vhodu so med seboj ločena s po enim presledkom. Zadnje vhodno število je vedno enako 2.

Višina piramide v vseh testnih primerih znaša najmanj 1, zamik pa najmanj 0. V 50% testnih primerov je zamik enak 0.

- ② V datoteki `Druga.java` dopolnite metodo

```
public static int[] manhattan(boolean[][] t),
```

ki sprejme pravokotno tabelo s H vrsticami in W stolpci, vrne pa tabelo dolžine $H + W - 1$, v kateri element na indeksu i pove, koliko parov elementov `true` v tabeli `t` leži na medsebojni manhattanski razdalji i . Manhattanska razdalja med elementoma na koordinatah (v_1, s_1) (indeks vrstice, indeks stolpca) in (v_2, s_2) znaša $|v_1 - v_2| + |s_1 - s_2|$.

Na desni strani so s črkami A, B, C, D in E označeni položaji elementov `true` v tabeli `t` v testnem razredu `Test01`. Za ta primer bi morala metoda vrniti tabelo `[0, 3, 4, 1, 2, 0]`. Na primer, element na indeksu 2 je enak 4 zato, ker imamo štiri pare na manhattanski razdalji 2: (A, B), (A, D), (B, D) in (B, E).

```
[ - - A - ]
[ - B C D ]
[ E - - - ]
```

V vseh testnih primerih velja $H \in [1, 50]$ in $W \in [1, 50]$. V 30% testnih primerov velja, da tabela `t` vsebuje natanko dva elementa `true`. V nadaljnjih 30% testnih primerov velja $H = 1$.

- ③ V razredu **Tretja** so definirani sledeči statični notranji razredi:

```
class Beseda {
    private String zapis;    // npr. mizi, kamnitih, ...
}
class Samostalnik extends Beseda {
    private char spol;      // 'M', 'Z' ali 'S'
    private int sklon;      // 1-6 (imenovalnik-orochnik)
    private int stevilo;    // 1-3 (ednina-mnozina)
}
class Pridevnik extends Beseda {
    private char spol;      // 'M', 'Z' ali 'S'
    private int sklon;      // 1-6
    private int stevilo;    // 1-3
}
class Glagol extends Beseda {
    private int oseba;      // 1-3
    private int stevilo;    // 1-3
}
class Stavek {
    private Beseda[] besede; // zaporedje besed v stavku
}
```

Rešite sledeče naloge:

- [32%] Dopolnite metodo `public Glagol poiisciGlagol()` v razredu **Stavek**, tako da bo vrnila prvi glagol v stavku `this`. Če stavek nima glagola, naj metoda vrne `null`. (Naloga je seveda javanska, ne slovnična: poiskati morate preprosto prvi objekt tipa **Glagol**.)
- [34%] Dopolnite metodo `public boolean preveriPS()` v razredu **Stavek**, tako da bo vrnila `true` natanko v primeru, če sta izpolnjena sledeča pogoja: (1) vsakemu pridevniku sledi bodisi pridevnik bodisi samostalnik; (2) stavek se *ne* konča s pridevnikom.
- [34%] V hierarhijo umestite abstraktni razred **SamPrid**. Razred naj bo podrazred razreda **Beseda** in nadrazred razredov **Samostalnik** in **Pridevnik**. Razred **SamPrid** naj poleg podedovanega atributa vsebuje vse tri skupne attribute razredov **Samostalnik** in **Pridevnik** ter konstruktor, ki nastavi vse štiri attribute. Ustrezno popravite tudi razreda **Samostalnik** in **Pridevnik**.

- ④ V razredu **Cetrta** je definiran statični notranji razred **Miks<T>**. Vsak njegov objekt vsebuje seznam (**List<T>**) in množico (**Set<T>**). Rešite sledeči nalogi:

- [50%] Dopolnite metodo `public void zamenjaj()`, tako da bo v objektu `this` med seboj zamenjala vsebino seznama in množice. Po zaključku metode mora seznam torej vsebovati elemente, ki jih je prej vsebovala množica, množica pa elemente, ki jih je prej vseboval seznam (a seveda brez morebitnih podvojitvev). Vrstni red elementov ni pomemben.
- [50%] Dopolnite razred tako, da bo implementiral vmesnik **Iterable<T>**. Iterator naj se najprej sprehodi po vseh elementih seznama (v vrstnem redu, kot ga določa iterator seznama), nato pa še po vseh elementih množice (v vrstnem redu, kot ga določa iterator množice).