

Izpit pri predmetu Programiranje I

15. februar 2018

Vse naloge lahko testirate z ukazom `tj.exe` (brez parametrov).
Veliko uspeha!

① Dopolnite sledeči metodi:

- `public static boolean jePodvojena(int[] t) [J1–J5, S1–S25]`

Vrne `true` natanko v primeru, če so vsi elementi v prvi polovici tabele `t` enaki istoležnim elementom v drugi polovici. Primer takšne tabele je $\{7, 10, 8, 7, 10, 8\}$, saj je prvi element enak četrtemu, drugi petemu, tretji pa šestemu. Tabela `t` ima v vseh testnih primerih sodo število elementov.

- `public static int[] steviloDeljivih(int[][] t, int d) [J6–J10, S26–S50]`

Vrne tabelo, v kateri i -ti element pove, koliko elementov v i -tem stolpcu tabele `t` je deljivih s številom `d`. Število elementov vrnjene tabele naj bo enako številu stolpcev tabele `t`.

② Igralci, označeni z indeksi od 0 do `stIgralcev - 1`, igrajo partije prek šahovskega portala. Vsaka posamezna zmaga prinese po dve točki, remi eno, poraz pa nič točk.

V razredu `Portal` dopolnite sledeče elemente:

- `public Portal(int stIgralcev)`

Inicializira objekt, ki predstavlja šahovski portal s `stIgralcev` igralci.

- `public void zabeleziPartijo(int beli, int crni, int izid)`

Zabeleži dejstvo, da sta igralca z indeksoma `beli` in `crni` (`beli \neq crni`) odigrala partijo, v kateri je igralec `beli` igral z belimi, igralec `crni` pa s črnimi figurami. Igra se je zaključila z izidom `izid` (0: remi, 1: zmaga bele, 2: zmaga črne).

- `public int steviloRemijev() [J1–J2, S1–S12]`

Vrne skupno število remijev v vseh doslej odigranih partijah.

- `public int tocke(int igralec) [J3–J5, S13–S25]`

Vrne skupno število točk, ki jih je doslej zbral igralec z indeksom `igralec`.

- `public int[] medsebojneTocke(int a, int b) [J6–J7, S26–S37]`

Vrne tabelo z dvema elementoma: prvi vsebuje skupno število točk, ki jih je igralec z indeksom `a` zbral v igrah proti igralcu z indeksom `b`, drugi pa skupno število točk, ki jih je igralec `b` zbral v igrah proti igralcu `a`. Na primer, če je dvakrat zmagal igralec `b` (ne glede na to, ali je igral z belimi ali črnimi figurami), ena partija pa se je končala z remijem, bi metoda vrnila tabelo $\{1, 5\}$.

- `public int pariZRemijem() [J8–J10, S38–S50]`

Vrne število različnih parov igralcev, pri katerih se je vsaj ena medsebojna partija končala z remijem. Para (i, j) in (j, i) štejemo kot en in isti par, ne kot dva različna para.

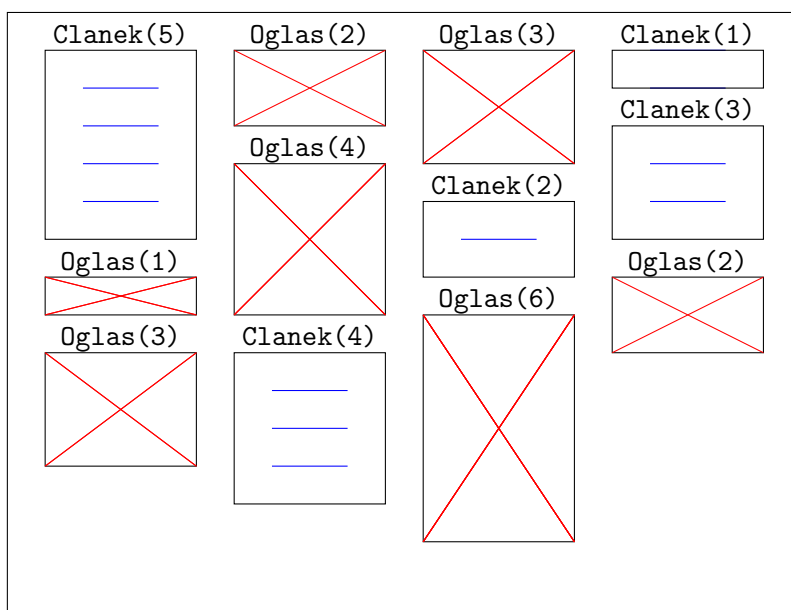
- ③ Metodo `narisi` dopolnite tako, da bo narisala izgled časopisne strani. Časopisno stran si lahko predstavljamo kot matriko objav (tabela `this.objave` tipa `Objava[][]`), v kateri so vsi stolpci enako široki, objave znotraj stolpca pa niso nujno enako visoke. (Kljub temu vsi stolpci vsebujejo enako število objav.) Stolpci so široki po `this.sirinaStolpca` centimetrov, objava pa je visoka `visina` (atribut razreda `Objava`) centimetrov. Razmik med posameznimi stolpci in med posameznimi objavami znotraj stolpca znaša 1 cm, enako velik pa je tudi odmik skrajno levega oz. desnega stolpca od levega oz. desnega roba platna in odmik prve vrstice objav od zgornjega roba platna.

Objave so dveh tipov: *članek* (razred `Clanek`) in *oglas* (razred `Oglas`). Vsako objavo narišite kot pravokotnik z belim polnilom in črno obrobo. Pravokotnik za članek naj vsebuje `visina - 1` enakomerno razmaknjenih modrih vodoravnih daljic, ki se se raztezajo čez polovico širine pravokotnika in so po širini postavljene na sredini pravokotnika. Pravokotnik za oglas naj vsebuje rdeč križ.

Število pik, ki ustreza enemu centimetru, določite tako, da se bo časopisna stran raztezala čez celotno širino platna.

Sledeča slika prikazuje videz platna za vrednost `sirinaStolpca = 4` in tabelo objav s sledečo vsebino (testni primer J8):

```
{ { new Clanek(5), new Oglas(2), new Oglas(3), new Clanek(1) },
  { new Oglas(1), new Oglas(4), new Clanek(2), new Clanek(3) },
  { new Oglas(3), new Clanek(4), new Oglas(6), new Oglas(2) } }
```



Poleg metode `narisi` dopolnite tudi metodo `pikNaCm`, ki vrne število pik na platnu, ki ustreza enemu centimetru na časopisni strani.