

Izpit pri predmetu Programiranje 1

1. februar 2019

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

① Dopolnite sledeči metodi:

- `public static int stakni(int a, int b)` [J1–J5, S1–S25]

Vrne stik pozitivnih števil `a` in `b`. Na primer, stik števil 726 in 3819 je število 7263819 oziroma $726 \cdot 10^4 + 3819$.

Uporabljate lahko le tipa `int` in `boolean` ter celoštevilske operacije `+`, `-`, `*`, `/` in `%`. Kršitev te zahteve bo kaznovana z razpolovitvijo točk.

- `public static boolean vednoBoljRes(boolean[][] t)` [J6–J10, S26–S50]

Vrne `true` natanko v primeru, če vsaka vrstica pravokotne tabele `t` (razen prve) vsebuje strogo več elementov `true` kot vrstica pred njo. Takšna je, na primer, sledeča tabela:

false	false	false	false	false	false
true	false	false	true	true	false
false	true	true	false	true	true
true	true	true	true	true	true

② Za mizo sedi n igralcev, označenih z indeksi $0, 1, \dots, n-1$, in igra igro z nk kartami z vrednostmi $0, 1, \dots, nk-1$. Igralci prejmejo po k kart, nato pa odigrajo k krogov (»štihov«); v vsakem štihi vsak od igralcev odvrže poljubno karto. Prvi štihi prične (tj. prvi odvrže karto) igralec 0, nato pa po eno karto po vrsti odvržejo igralci $1, 2, \dots, n-1$. Igralec, ki je v pravkar odigranem štihi odvrzel karto z najvišjo vrednostjo, pospravi vse odvržene karte v svoj kupček in prične naslednji štihi. Če štihi prične, denimo, igralec r , potem za njim odvržejo karto igralci $r+1, r+2, \dots, n-1, 0, 1, \dots, r-1$.

Podana sta razreda `Stih` in `Partija`:

```
class Stih {
    private int zacetnik; // igralec, ki prične štihi
    private int[] karte;  // vrednosti kart, ki so jih igralci po vrsti odvrgli v štihi
} // (na indeksu 0 je karta, ki jo je odvrzel začetnik štiha)

class Partija {
    private int stIgralcev; // vrednost n v zgornjem besedilu
    private Stih[] stihi;  // po vrsti odigrani štihi (podatki o začetnikih štihov)
} // so usklajeni s potekom partije
```

Dopolnite sledeče metode:

- `public int vrednost()` v razredu `Stih` [J1–J2, S1–S12]

Vrne vsoto vrednosti kart, odvrženih v štihi `this`.

- `public int dobitnik()` v razredu `Stih` [J3–J5, S13–S25]

Vrne indeks igralca, ki je v štihi `this` odvrzel karto z največjo vrednostjo.

- `public int kdoJeVrgel(int karta)` v razredu `Partija` [J6–J7, S26–S37]

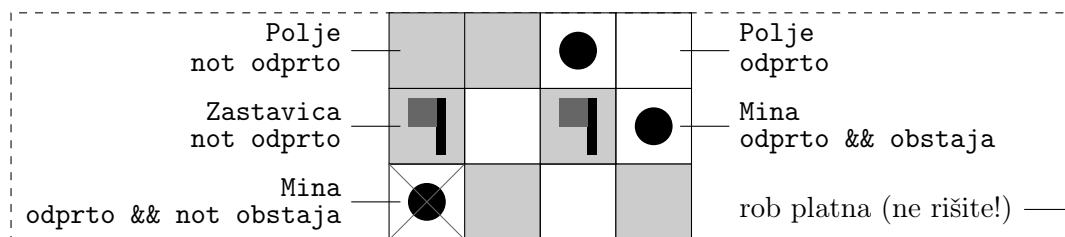
Vrne indeks igralca, ki je v partiji `this` odvrzel karto z vrednostjo `karta`, oziroma `-1`, če podana karta ni bila odvržena.

- `public int[] tocke()` v razredu `Partija` [J8–J10, S38–S50]

Vrne tabelo z n elementi, v kateri element na indeksu i podaja skupno vsoto vrednosti kart, ki jih je v partiji `this` v svoj kupček pospravil igralec i .

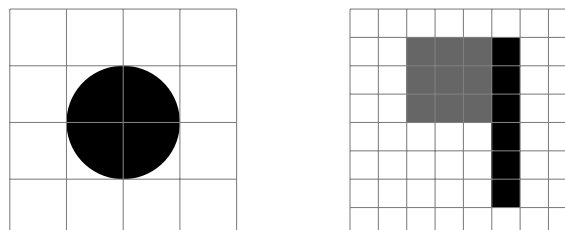
- ③ Metodo `narisi` dopolnite tako, da bo na podlagi pravokotne tabele `this.polja` (tipa `Polje[][]`) narisala igralno površino igre Minolovec. Igralna površina je pravokotnik, sestavljen iz enako velikih kvadratnih polj. Elementi tabele `this.polja` podajajo stanja istoležnih polj. Polja so treh tipov: polje z mino (razred `Mina`), polje z zastavico (razred `Zastavica`) in polje brez vsebine (razred `Polje`). Razreda `Mina` in `Zastavica` sta podrazreda razreda `Polje`. Polje je lahko odprto ali zaprto (atribut `odprto`). Polje z mino ima poleg tega še atribut `obstaja`, ki pove, ali mina na polju dejansko obstaja ali pa gre le za zgrešeno domnevo.

Sledeča slika prikazuje platno z igralno površino v testnem primeru J9:



Igralna površina naj bo v celoti vidna na platnu. Po eni od dimenzij naj se razteza čez celotno stranico platna, po drugi pa naj leži na sredini platna.

Zaprta polja naj bodo pobarvana z barvo `BARVA_ZAPRTO`, odprta pa z barvo `BARVA_ODPRTO`. Robovi polj naj bodo narisani z barvo `Color.BLACK`. Mine naj bodo pobarvane z barvo `Color.BLACK`, zastavice pa z barvama `Color.BLACK` (»drog«) in `Color.RED` (»blago«). Križ čez neobstoječo mino naj bo narisani z barvo `Color.RED`. Pri risanju mine in zastavice se zgledujte po sledečih shemah:



Poleg metode `narisi` dopolnite tudi metodi `stranicaPolja`, ki vrne dolžino stranice polja (v slikovnih pikah), in `zgornjiLeviKot`, ki vrne tabelo z dvema elementoma: prvi podaja koordinato x , drugi pa koordinato y zgornjega levega kota igralne površine.