Izpit pri predmetu Programiranje 1
22. avgust 2019

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

(1) Dopolnite sledeči metodi:

- `public static int odrezi(int n)` [J1–J5, S1–S25]

  Vrne število, ki ostane, ko iz pozitivnega števila `n` odstranimo vse končne ničle. Na primer, pri `n` = 8056000 je rezultat enak 8056.

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

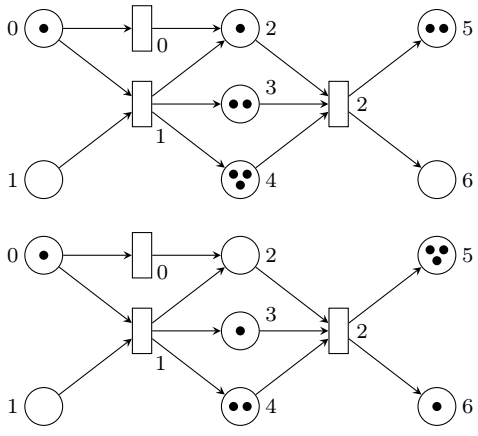
- `public static boolean enakeVsote(int[][] t)` [J6–J10, S26–S50]

  Vrne `true` natanko v primeru, če imajo vse vrstice pravokotne tabele `t` enako vsoto. Takšna je, na primer, sledeča tabela:

$$\begin{bmatrix} 3 & 2 & 4 & 1 \\ 0 & 10 & 0 & 0 \\ 8 & 3 & -6 & 5 \end{bmatrix}$$

(2) *Petrijevo mrežo* tvorijo *mesta* in *prehodi*. Vsako mesto je povezano s poljubno mnogo vstopnimi in izstopnimi prehodi, vsak prehod pa z vsaj enim vstopnim in vsaj enim izstopnim mestom. Mesta lahko vsebujejo *žetone*. Prehod se lahko *sproži*, če ima vsako od njegovih vstopnih mest vsaj po en žeton. Če se prehod sproži, se iz vsakega od vstopnih mest odvzame po en žeton, v vsako od izstopnih mest pa doda po en žeton. Izpis na levi prikazuje razrede `Mesto`, `Prehod` in `Mreza`, slika na desni pa začetno stanje Petrijeve mreže in stanje po sprožitvi prehoda z indeksom 2 (mesta so prikazana s krogi, prehodi pa s pravokotniki):

```
class Mesto {
    private Prehod[] v;     // vstopni prehodi
    private Prehod[] iz;    // izstopni prehodi
    private int stZetonov; // število žetonov
}
class Prehod {
    private Mesto[] v;  // vstopna mesta
    private Mesto[] iz; // izstopna mesta
}
class Mreza {
    private Mesto[] mesta;     // mesta v mreži
    private Prehod[] prehodi; // prehodi v mreži
}
```



Dopolnite sledeče metode:

- `public boolean seLahkoSprozi()` v razredu `Prehod` [J1–J2, S1–S12]

  Vrne `true` natanko v primeru, če se prehod `this` lahko sproži.

- `public void sprozi()` v razredu `Prehod` [J3–J5, S13–S25]

  Izvrši sprožitev prehoda `this`. V vseh klicih te metode v testnih razredih je pogoj za sprožitev izpolnjen.

- `public boolean jeNaslednikOd(Mesto mesto)` v razredu `Mesto` [J6–J7, S26–S37]

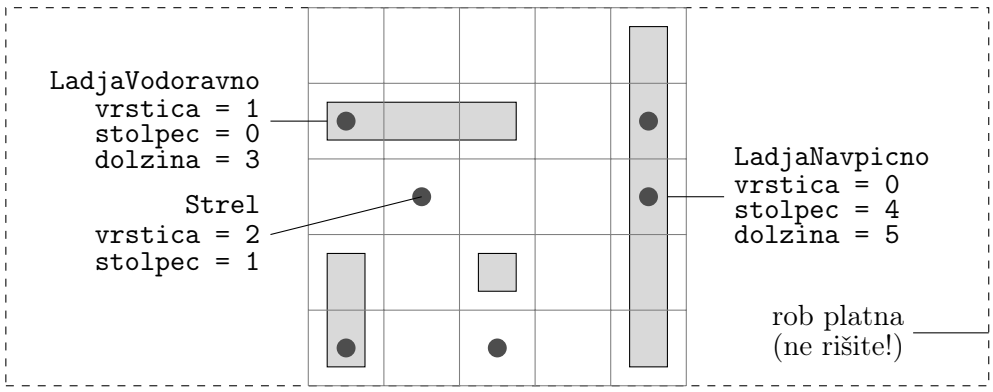
  Vrne `true` natanko v primeru, če obstaja prehod, ki je hkrati vstopni prehod mesta `this` in izstopni prehod mesta `mesto`.

- `public void doKonca()` v razredu `Mreza` [J8–J10, S38–S50]

  Naj bo p število prehodov. Metoda `doKonca` naj po vrsti sproži prehod 0 (tj. prehod z indeksom 0 v tabeli `this.prehodi`), prehod 1, ..., prehod $p-1$, nato spet prehod 0, prehod 1, ..., prehod $p-1$, nato ponovno prehod 0, prehod 1 itd. Če katerega od prehodov ni mogoče sprožiti, naj ga ignorira. Postopek naj se izvaja tako dolgo, dokler še obstaja prehod, ki ga je mogoče sprožiti.

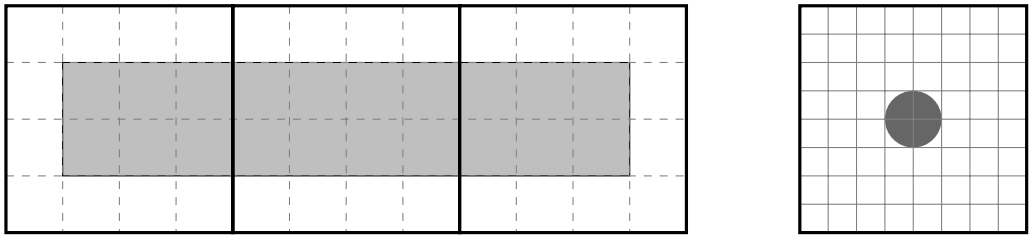
(3) Metodo `narisi` dopolnite tako, da bo na podlagi števila `this.stranica` ter tabel `this.ladje` (tipa `Ladja[]`) in `this.streli` (tipa `Strel[]`) narisala igralno površino igre Potapljanje ladjic. Igralna površina je kvadrat, sestavljen iz `this.stranica` × `this.stranica` enako velikih kvadratnih polj. Elementi tabel `this.ladje` in `this.streli` vsebujejo podatke o posameznih ladjah oziroma strelih. Strel je določen z indeksoma vrstice in stolpca, ladja pa z indeksoma vrstice in stolpca zgornjega levega polja ter z dolžino (številom polj, ki jih zavzema). Ladja je lahko postavljena vodoravno (razred `LadjaVodoravno`) ali navpično (razred `LadjaNavpicno`). Razreda `LadjaVodoravno` in `LadjaNavpicno` sta podrazreda razreda `Ladja`.

Sledeča slika prikazuje platno z igralno površino v testnem primeru J8:



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.

Mrežne črte narišite z barvo `Color.GRAY`. Ladje pobarvajte z barvo `Ladja.BARVA` in obrobite z barvo `Color.BLACK`, strele pa pobarvajte z barvo `Strel.BARVA`. Pri risanju ladje in strela 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.