

# Izpit pri predmetu Programiranje I

## 29. avgust 2018

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

① Dopolnite sledeči metodi:

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

Vrne vsoto  $a_0 - a_1 + a_2 - a_3 + \dots$ , kjer  $a_0$  predstavlja *zadnjo* (!) številko pozitivnega števila  $n$ ,  $a_1$  predzadnjo itd.

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

Vrne tabelo, v kateri  $i$ -ti element podaja položaj (indeks vrstice) najnižje ležečega znaka `*` v  $i$ -tem stolpcu pravokotne tabele `t`. Če v stolpcu ni znaka `*`, naj ima pripadajoči element vrnjene tabele vrednost  $-1$ . Dolžina vrnjene tabele naj bo enaka številu stolpcev tabele `t`.

② Podani so razredi `TrgovskiCenter`, `Trgovina` in `Izdelek` (prikazani so samo atributi):

```
class TrgovskiCenter {
    private Trgovina[][] trgovine;    // trgovine, ki tvorijo trgovski center
}
class Trgovina {
    private Izdelek[] izdelki;        // izdelki, ki jih trgovina prodaja
    private int[] zaloga;              // zaloga[i]: trenutna zaloga izdelka izdelki[i]
    private int saldo;                 // trenutno stanje na računu trgovine
}
class Izdelek {
    private String naziv;
    private int cena;
}
```

V razredu `Trgovina` dopolnite sledeči metodi:

- `public int poiisciIzdelek(String naziv)` [J1–J2, S1–S12]

Vrne indeks izdelka z nazivom `naziv` v tabeli `this.izdelki` oziroma  $-1$ , če iskanega izdelka ni v tabeli.

- `public int prodaj(int indeks, int n)` [J3–J5, S13–S25]

Če ima trgovina `this` na zalogi vsaj  $n$  izdelkov z indeksom `indeks`, jih proda  $n$ , sicer pa toliko, kolikor jih ima na zalogi. Metoda naj vrne prihodek od prodaje izdelkov, še pred tem pa naj poveča saldo (za prihodek) in zmanjša zalogo.

V razredu `TrgovskiCenter` dopolnite sledeči metodi:

- `public boolean vSosescini(int vr, int st, String naziv)` [J6–J7, S26–S37]

Vrne `true` natanko v primeru, če trgovina na položaju (`vr`, `st`) (indeks vrstice, indeks stolpca) ali pa katerakoli od njenih sosed prodaja izdelek z nazivom `naziv`, četudi ga morda trenutno nima na zalogi. V vseh testnih primerih ima trgovina na položaju (`vr`, `st`) osem sosed: levo, desno, zgornjo, spodnjo in štiri diagonalne.

- `public int pohod(int[][] premiki, String naziv)` [J8–J10, S38–S50]

Pričnemo v trgovini na položaju (0, 0), potem pa se premikamo v skladu z navodili, zapisanimi v vrsticah tabele `premiki`. V vsaki trgovini (tudi v začetni in končni) pokupimo celotno zalogo izdelkov z nazivom `naziv`. Metoda naj ustrezno posodablja zalogo in saldo v obiskanih trgovinah, vrne pa naj skupno količino denarja, ki ga porabimo na pohodu.

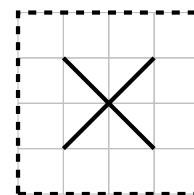
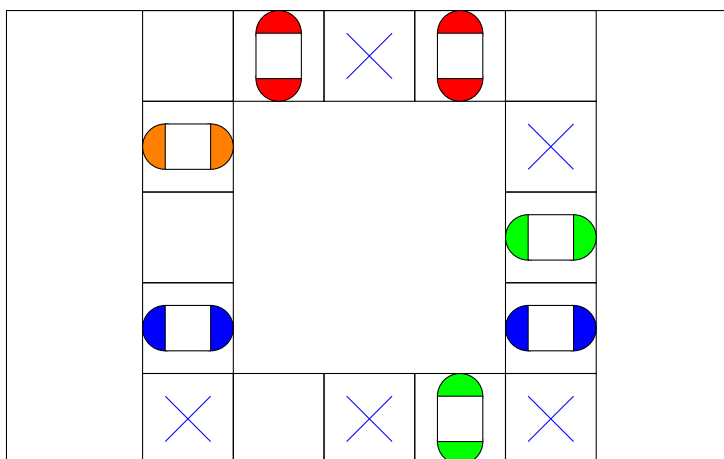
Vsaka vrstica tabele `premiki` ima po dva elementa. Prvi podaja vrstični premik (razliko med indeksoma ciljne in trenutne vrstice), drugi pa stolpčni premik. V vseh testnih primerih so vsi premiki legalni.

- ③ Metodo `narisi` dopolnite tako, da bo narisala igralno površino različice igre Monopoly. Igralna površina ima obliko kvadratnega okvirja z `this.dolzina` polji po stranici. Podatki o posameznih poljih so nanizani v tabeli `this.polja` (tipa `Polje[]`). Prvi element tabele se nanaša na polje v zgornjem levem kotu igralne površine, nato pa si polja sledijo v smeri urinega kazalca.

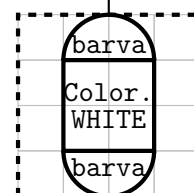
Polja so treh vrst: *navadno polje* (razred `Polje`), *presenečenje* (razred `Presenecenje`) in *posest* (razred `Posest`). Vsako polje v osnovi narišite kot kvadrat z belim polnilom in črno obrobo. Polje presenečenja naj poleg tega vsebuje še križ barve `Color.BLUE`, posest pa »kapsulo« barve `barva` (atribut v razredu `Posest`). Polja v vogalih igralne površine so lahko samo navadna polja ali presenečenja.

Sledi primer, za katerega velja `this.dolzina == 5`. Pri risanju križa in kapsule si pomagajte s shemama na desni. Upoštevajte, da je kapsula pri posestih na levem in desnem robu igralne površine obrnjena za 90 stopinj.

```
this.polja: {
    new Polje(), new Posest(Color.RED), new Presenecenje(), new Posest(Color.RED),
    new Polje(), new Presenecenje(), new Posest(Color.GREEN), new Posest(Color.BLUE),
    new Presenecenje(), new Posest(Color.GREEN), new Presenecenje(), new Polje(),
    new Presenecenje(), new Posest(Color.ORANGE), new Polje(), new Posest(Color.BLUE),
};
```



`Color.BLACK`



Igralna površina naj zavzema celotno krajšo stranico platna, po daljši pa naj bo prikazana na sredini.

Poleg metode `narisi` dopolnite še metode `stranicaPolja` (vrne dolžino stranice polja, merjeno v pikah), `xLevo` (vrne koordinato  $x$  levega roba igralne površine) in `yZgoraj` (vrne koordinato  $y$  zgornjega roba igralne površine).