

Objektno programiranje, Ispit JUN1, Grupa 2

Matematički fakultet

Školska godina 2017/2018

Napomena: Napraviti direktorijum /home/ispit/IdeaProjects. Pokrenuti *IntelliJ Idea* i za Project location postaviti /home/ispit/IdeaProjects direktoriju, a za Project name oop_Ime_Prezime_Indeks_Asistent (npr. oop_Pera_Peric_mi12082_NM). U napravljenom projektu, paket takođe nazvati isto tako.

Kod **ne sme** imati sintaksnih grešaka niti izbacivanje NullPointerException-a.

Vreme za rad: **2.5 sata**

Inicijalni asistenti: Biljana - BS, Anja - AB, Božidar - BA, Nemanja - NM

U tekstu je dat opis klasa, njihovih atributa i metoda. **Dozvoljeno** je (i ohrabrujemo Vas) dodati nove attribute, klase, metode, enume, interfejsa u slučaju da Vam olakšavaju implementaciju, i/ili smatrate da Vam poboljšavaju kvalitet koda i slično. Nekada će zahtevi u zadatku i zahtevati od Vas da dodate novi atribut ili slično.

Da bi se uspešno položio ispit potrebno je osvojiti barem 50% poena.

1. Napraviti klasu `Ranac` koju karakterišu atributi `maxTezina` (`int` maksimalna težina koju ranac može izdržati) i `slobodnoTezina` (`int`). Implementirati `get` metod za atribut `slobodnoTezina`, konstruktor koji prihvata vrednosti atributa `maxTezina` i na istu vrednost postavlja i atribut `slobodnoTezina`. Implementirati metod `bool smestiDragulj` (`double tezina`) koji proverava da li predmet zadate težine može stati u ranac i umanjuje vrednost polja `slobodnoTezina` ako može. Implementirati `toString` metod da ispisuje ranac u obliku:

```
ranac: slobodnoTezina(maxTezina)
```

2. Napraviti apstraktnu klasu `ObjekatIgre` koja predstavlja jedan objekat u simulaciji igre koju pravite. Klasa se karakteriše sledećim atributima:

- `String naziv` - naziv Objekta
- `(protected)int X` - x koordinata položaja objekta u 2D prostoru
- `(protected)int Y` - y koordinata položaja objekta u 2D prostoru

Implementirati konstruktor koji prihvata `naziv`, `X` i `Y`. Implementirati potrebne `get` metode. Klasa poseduje apstraktan metod `public abstract void izvrsiAkciju()`.

Implementirati `toString` tako da objekte ispisuje na sledeći način:

```
[X,Y] naziv  
[4,5] Indiana Jones  
[5,5] Rubin  
[0,0] Scrooge McDuck
```

3. Napraviti klasu `Rudnik` koja nasleđuje klasu `ObjekatIgre`. `Rudnik` poseduje dodatne attribute `otvorenJos` (`int` koliko je još poteza igre rudnik otvoren) i `tezinaDragulja` (`double`). Implementirati konstruktor koji prihvata attribute nadklase i oba dodatna atributa, potrebne `get` metode kao i metod `isOtvoren` (`boolean`). Implementirati metod `public void izvrsiAkciju()` koja ukoliko je rudnik otvoren umanjuje vrednost polja `otvorenJos` za 1.

Implementirati `toString` da ispisuje u obliku:

```
Rudnik: \textit{naziv} je otvoren jos otvorenJos poteza
```

4. Napraviti klasu `Tragac` koja nasleđuje klasu `ObjekatIgre`. `Tragac` poseduje dodatni atribut `ranac` (`Ranac`) i privatna statička polja `random` (`Random`) i `smerKretanja` (`String`) koji može imati vrednost `dd` (dole-desno) ili `gl` (gore-levo). Implementirati konstruktor koji osim polja nadklase prihvata i atribut `ranac`, `get` metodu za polje `ranac`, statičku metodu `setSmerKretanja(String sk)` koja `smerKretanja` postavlja na `sk` i metod `pokupiDragulj` (`Rudnik rudnik`) (`boolean`) koja pozivanjem metode `smestiDragulj` svog ranca pokušava da pokupi dragulj iz rudnika. Implementirati metod `public void IzvrsiAkciju()` koji u zavisnosti od vrednosti polja `smerKretanja` menja poziciju tragača tako što nasumično bira da li će se tragač pomeriti jedno polje dole ili jedno polje desno u slučaju da `smerKretanja` ima vrednost `dd` ili će se pomeriti jedno polje gore ili jedno polje levo u slučaju da `smerKretanja` ima vrednost `gl`. *Polje po kom se igra možete zamisliti kao dvodimenzionu matricu i koordinate u kretanju menjati tome shodno.*

Implementirati `toString` da ispisuje u obliku:

```
Tragac: \textit{naziv} ima ranac: slobodnoTezina(maxTezina)
```

5. Napraviti klasu `Mapa` koja nasleđuje `Application` klasu biblioteke `javafx` i izgleda kao na slici 1. Obezbediti da je unapred selektovano prvo radio dugme (gledano odozgo) i da u svakom trenutku može biti selektovano tačno jedno radio dugme.

Na klik dugmeta `Ucitaj` iz datoteke `objekti.txt` učitavaju se objekti igre i smeštaju u `List<ObjekatIgre>`. Potom se lista sortira tako da prvo u listi idu rudnici (u datoteci počinju sa `r`) i to neopadajuće po otvorenom `Jos`, a potom tragači (u datoteci počinju sa `t`) i to nerastuće po maksimalnoj težini ranca koji nose. Nakon sortiranja u gornji `TextArea` element se ispisuju objekti igre u sortiranoj listi, svaki u novom redu (koristeći `toString`) nad klasom `ObjekatIgre` (pogledati sliku 2). Sadržaj radio dugmadi postaviti na vrednosti kao na slici 2.

Na klik dugmeta `Izaberi smer`, tragačima se postavlja smer kretanja (svima isti) u zavisnosti od toga šta je odabrao korisnik (koristiti metod `setSmerKretanja(String smer)`).

Na klik dugmeta `Simuliraj potez` simulira se jedan potez u igri tako što prvo svaki `ObjekatIgre` izvrši akciju pozivom `izvrsiAkciju`. Zatim tragači koji se nalaze na polju na kom i neki od otvorenih rudnika pokušavaju da pokupe dragulj iz njega (metodom `pokupiDragulj`). Ako je tragač uspeo da pokupi dragulj u donji `TextArea` ispisati ime tragača i poruku o uspešnom sakuljanju dragulja (slika 2) a u suprotnom ispisati da nije uspeo da smesti dragulj u ranac.

Nakon svakog odigranog poteza, potrebno je u donji `TextArea` element ispisati sledeće:

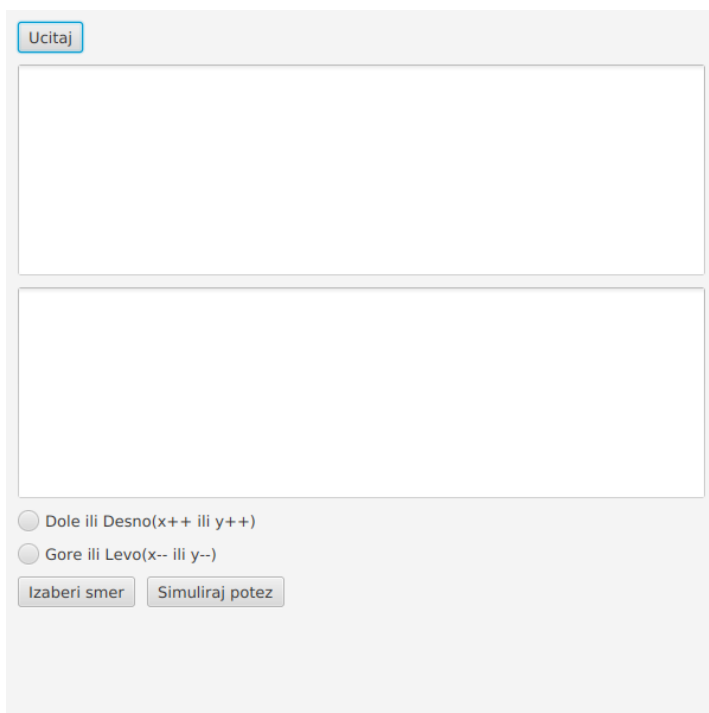
- Broj do sada pokupljenih dragulja (slika 2)
- Stanje svih objekata igre (slika 2)

Dozvoljeno je proširiti klase dodatnim atributima i metodama kako biste realizovali prethodno navedene zahteve.

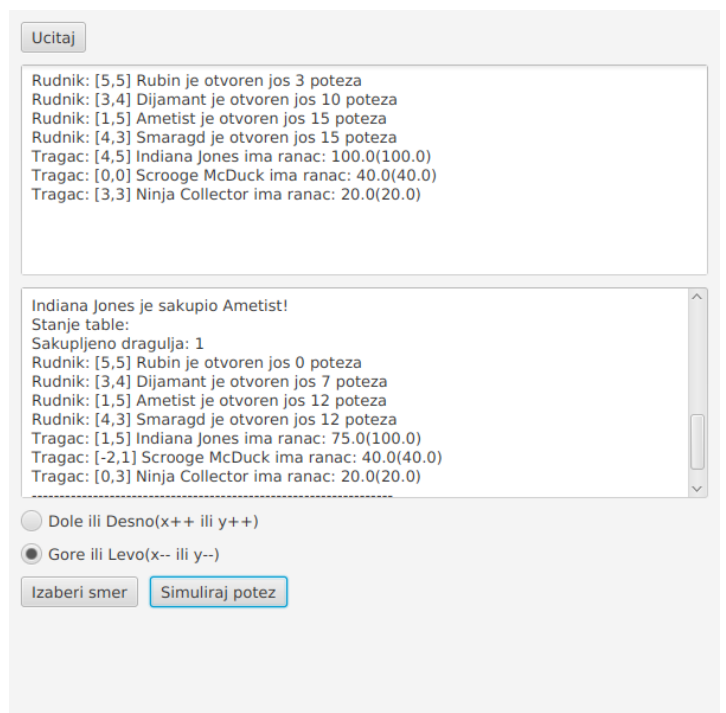
[`objekti.txt`]

`t, ime tragača, x koordinata, y koordinata, maksimalna tezina koju ranac moze da ponese`
`r, naziv rudnika, x koordinata, y koordinata, tezina dragulja, koliko je poteza jos otvoren`

```
t, Indiana Jones, 4, 5, 100
r, Rubin, 5, 5, 15, 3
t, Scrooge McDuck, 0, 0, 40
r, Dijamant, 3, 4, 25, 10
t, Ninja Collector, 3, 3, 20
r, Ametist, 1, 5, 25, 15
r, Smaragd, 4, 3, 30, 15
```



Slika 1: Izgled grafičkog korisničkog interfejsa



Slika 2: Nakon učitavanja i nekoliko rundi simulacije