

## Objektno programiranje, Ispit JUN1, Grupa 1

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 takode nazvati isto tako.

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

Vreme za rad: **2.5 sata**

Inicijalni asistenata: 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, enum, interfejs 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.

*Pokemoni predstavljaju jedan od popularnijih (japanskih) crtanih filmova. Crtani je originalno emitovan krajem devedestih i početkom dvehiljadite, priča se vrti oko Pokemon trenera koji sakupljaju stvorenja (Pokemone), treniraju ih i potom se međusobno takmiče u borbama Pokemona. Danas ste Vi trener Pokemona i potrebno je da simulirate borbu Vašeg najboljeg Pokemona protiv računara!*

1. Napraviti klasu `Magija` koju karakterišu atributi `snaga (int)` i `ime (String)`. Implementirati `get` metode za navedene attribute, konstruktor koji prihvata vrednosti za oba atributa i konstruktor kopije. Implementirati `toString` metod da ispisuje magiju u obliku:

```
ime snaga
```

2. Napraviti apstraktnu klasu `Pokemon` koja predstavlja Pokemona. Klasa se karakteriše sledećim atributima:

- `String ime` - ime Pokemona
- `int nivo` - nivo Pokemona (broj koji najmanje može biti 1)
- `Magija magija1` - prva magija koju poseduje Pokemon
- `Magija magija2` - druga magija koju poseduje Pokemon
- `int zivot` - procenat života Pokemona (na početku 100)

Implementirati konstruktor koji prihvata `ime`, `nivo`, `magija1`, `magija2`, a `zivot` postavlja na 100. Implementirati potrebne `get` metode. Implementirati metod `void nanesiStetu(int kolikoStete)` koji umanjuje `zivot` za vrednost `kolikoStete`. Klasa poseduje apstraktan metod `public abstract void baciMagiju(Pokemon neprijatelj)`.

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

```
[nivo] ime  
[lv13] Pikachu  
[lv14] Bulbasaur  
[lv19] MewTwo
```

3. Napraviti klasu `IgracPokemon` koja nasleđuje klasu `Pokemon`. Igrač poseduje atribut `izabranaMagija (int)` čije vrednosti mogu biti 1 ili 2. Implementirati metod `Magija getIzabranaMagija()` koji vraća prvu ili drugu magiju u zavisnosti od vrednosti atributa `izabranaMagija` i metod `void setIzabranaMagija(int i)` koja postavlja izabranu magiju na prvu ukoliko je `i=1`, na drugu ukoliko je `i=2`, a inače ne radi ništa. Implementirati metod `public void baciMagiju(Pokemon neprijatelj)` tako da prosleđenom pokemonu nanosi štetu koju poseduje izabrana magija.
4. Napraviti klasu `CpuPokemon` koja nasleđuje klasu `Pokemon`. Klasa poseduje privatno statičko polje `random (Random)`. Implementirati metod `public void baciMagiju(Pokemon neprijatelj)` tako da sa verovatnoćom 0.5 baca prvu magiju, i sa verovatnoćom 0.5 baca drugu magiju na prosleđenog neprijatelja.

Klasa `Random` poseduje metod `nextDouble` koji generiše pseudo-slučajni broj iz intervala `[0, 1]` i može Vam koristiti u implementaciji nekog metoda klase.

5. Napraviti klasu `PokemonArena` 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 `pokemoni.txt` učitavaju se pokemoni i smeštaju u `List<Pokemon>`. Potom se lista sortira tako da prvo u listi idu pokemoni koji pripadaju igraču (u datoteci počinju sa `p`), a potom računaru (u datoteci počinju sa `n`) - na primer može se koristiti operator `instanceof`. Ukoliko igrač (ili računar) poseduju više pokemona,

potrebno ih je sortirati po atributu `nivo` nerastuće. Nakon sortiranja u gornji `TextArea` element se ispisuju pokemoni u sortiranoj listi, svaki u novom redu (koristeći `toString`) nad klasom `Pokemon` (pogledati sliku 2). Nakon ispisa, igraču se postavlja njegov najjači pokemon (prvi element sortiranje liste), a računaru njegov najjači pokemon (prvi računarski pokemon nakon poslednjeg igračevog). Sadržaj radio dugmasti postaviti na vrednosti `toString` metoda za magije igračevih pokemona (pogledati radio dugmad na slici 2).

Na klik dugmeta `Izaberi magiju`, igračevom pokemonu se postavlja prva ili druga magija u zavisnosti od toga šta je odabrao korisnik (koristiti metod `setIzabranaMagija(int i)`). Nakon toga, u donji `TextArea` potrebno je dodati tekst koji obaveštava korisnika programa da je igrač izabrao magiju, na primer:

```
[igrac] Bira magiju vodeni mlaz 20
```

Na klik dugmeta `Simuliraj` dešava se simulacija jedne runde borbe pokemona (slika 2). Prvo se proverava da li nečiji pokemon ima  $\leq 0$  života. Ako je to slučaj, za pobednika se proglašava drugi pobednik porukom `Pobedio je racunar` ili `Pobedio je igrac`.

Ako pokemoni imaju više od 0 života, prvo igra igračev pokemon i ispaljuje odabranu magiju - koristiti metod `baciMagiju`, a nakon toga igra računar i ispaljuje svoju magiju - koristiti takođe metod `baciMagiju`. Nakon odigrane **jedne runde** poteza, potrebno je u donji `TextArea` element prijaviti šta se desilo i to:

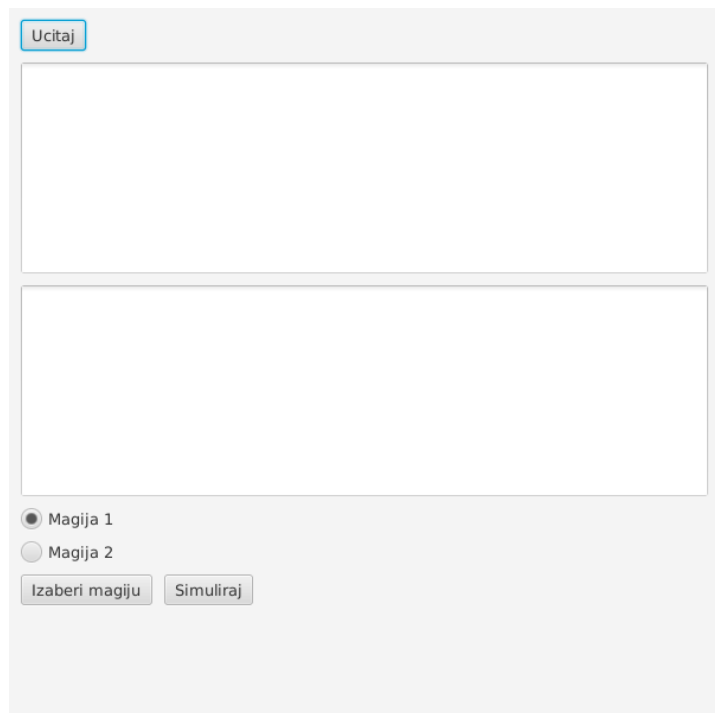
- Koju magiju je bacio igrač
- Koju magiju je bacio računar
- Koliko života sada ima igrač
- Koliko života sada ima računar

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

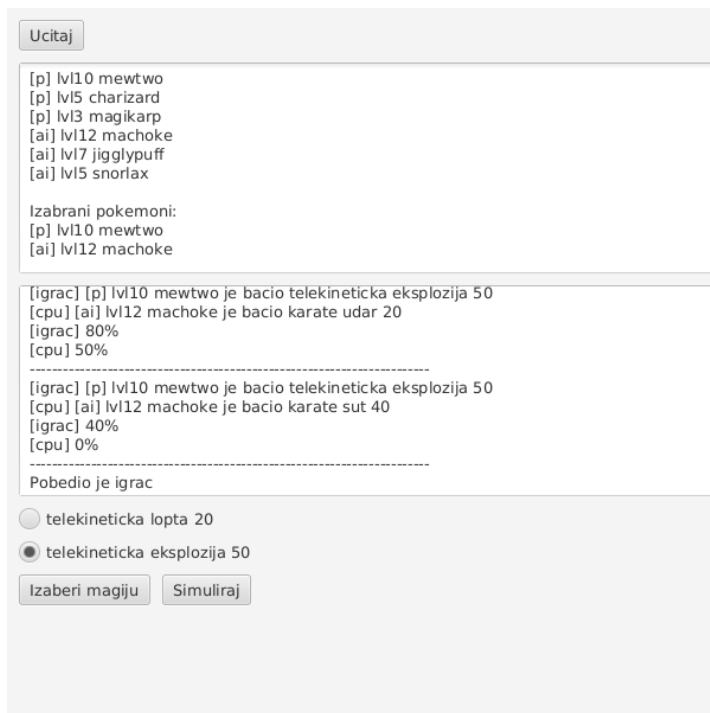
```
[pokemoni.txt]
```

```
ciji je pokemon, ime pokemona, nivo pokemona, ime magije1, snaga magije1, ime magije2, snaga magije2
```

```
p, charizard, 5, vatrena lopta, 20, vatreno koplje, 30
p, mewtwo, 10, telekineticka lopta, 20, telekineticka eksplozija, 50
p, magikarp, 3, vodeni mehur, 10, vodeni mlaz, 20
n, snorlax, 5, zaspi, -30, sedi na protivnika, 50
n, machoke, 12, karate udar, 20, karate sut, 40
n, jigglypuff, 7, napad slatkoce, 20, napad zagrljajima, 30
```



Slika 1: Izgled grafičkog korisničkog interfejsa



Slika 2: Nakon učitavanja i nekoliko rundi simulacije