

## Objektno-orijentisano programiranje, Ispit SEPTEMBAR2

Matematički fakultet

Školska godina 2017/2018

**Napomena:** Na Desktop-u napraviti direktorijum pod imenom `oop_Ime_Prezime_Indeks_Asistent` (npr. `oop_Pera_Peric_mi12082_NM`). Pokrenuti *Intellij Idea* i u napravljenom direktorijumu napraviti projekat sa istim nazivom. U napravljenom projektu, paket takođe nazvati 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, interfejsu 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.

Igra podmornice igra se na tabli  $10 \times 10$  u dvoje. Svaki igrač ima 5 podmornica veličine od 1 do 5 polja, po jednu od svake. Cilj je potopiti sve protivničke podmornice pre nego što on potopi Vaše.

1. Napraviti klasu `Polje` koja predstavlja jedno polje na tabli, a opisuju je dva celobrojna atributa `x` i `y`. Implementirati konstruktor koji prima vrednosti za sve attribute, potrebne `get` metode i metod `toString()` koji vraća string oblika `[x, y]`.
2. Napraviti klasu `Podmornica` koja predstavlja jednu podmornicu na tabli. Opisuju je atributi `pocetnoPolje` (`Polje`), `pravac` (`char`, uzima vrednosti 'd' - dole i 'r' - desno) i `velicinaPodmornice` (`int`). `Podmornica` se prostire od polja `pocetnoPolje` u zatom pravcu preko `velicinaPodmornice` polja. Klasa sadrži atribut `zauzetaPolja` (`Map<Polje, Boolean>`) u kom se pamte polja preko kojih se podmornica prostire i da li ih je protivnik gađao. Implementirati:
  - konstruktor koji prima vrednosti za prva tri polja (podrazumevati da su sve vrednosti ispravne), a četvrto inicijalizuje na osnovu prva tri,
  - metod `boolean pogodjenaPodmornica(Polje k)`, koji proverava da li se podmornica prostire preko zatomog polja `k`,
  - metod `void pogodjenoPolje(Polje k)`, koji označava da je protivnik pogodio polje `k` tako što postavlja vrednost za dato polje u mapi na `true`,
  - metod `toString()` koji vraća string oblika: `[x, y] - [q, w]` (`velicinaPodmornice`) gde su `x` i `y` koordinate početnog polja, a `q` i `w` su koordinate krajnjeg polja na kojima se prostire podmornica.
3. Napraviti apstraktnu klasu `Igrac`. Klasa se karakteriše atributima `podmornice` (`Map<Podmornica, Boolean>` - vrednosti označavaju da li je podmornica potopljena ili ne) i protivnik (`Igrac`). Klasa sadrži i apstraktni metod `boolean inicijalizuj()` kojim se postavljaju igračeve podmornice na tabli. Implementirati:
  - podrazumevani konstruktor, koji poziva ovaj apstraktni metod,
  - potrebne `get` i `set` metode,
  - metod `dodajPodmornicu(Podmornica p)`, koji dodaje podmornicu `p` u mapu a vrednost postavlja na `false`,
  - metod `potopiPodmornicu(Podmornica p)`, koji menja vrednost odgovarajućeg elementa u mapi na `true`.
4. U klasi `Igrac` implementirati metod `boolean odigrayPotez(Polje k)`. Ovaj metod gađa polje `k`, ukoliko ga ranije nije gađao, i proverava da li je pogodena neka podmornica (metodom `pogodjenaPodmornica`). Ukoliko jeste, ažurira mapu zauzetih polja (metodom `pogodjenoPolje`) i vraća `true`. Dodatno, metod proverava da li je pogodena podmornica ovim potezom i potopljena (sva polja koja zauzima su pogodena). U tom slučaju potrebno je ažurirati odgovarajuću vrednost u protivničkoj mapi (metodom `potopiPodmornicu`). Ukoliko se na polju `k` ne nalazi nijedna protivnička podmornica ili je polje već gađano (bez obzira da li je nešto pogodeno ili ne) treba vratiti `false`.
5. U klasi `Igrac` implementirati metod `boolean pobedio()` koji proverava da li je igrač pobedio. Igrač je pobedio ukoliko su sve protivničke podmornice potopljene.
6. Napraviti klasu `Covek` koja nasleđuje klasu `Igrac`. Potrebno je implementirati apstraktni metod `natklase`. Podaci o čovekovim podmornicama nalaze se u datoteci `podmornice.txt`. Jedan red datoteke sadrži podatke o jednoj podmornici u obliku:  
`x, y, pravac, velicinaPodmornice`  
U slučaju uspešnog učitavanja vratiti `true`, u suprotnom `false`. Pretpostaviti da su podaci u datoteci ispravni.

7. Napisati klasu `Racunar` koja nasleđuje klasu `Igrac`. Implementirati apstraktni metod natklase tako što će svi potrebni podaci o podmornicama biti nasumično odabrani. Koordinate se biraju iz intervala  $[0, 9]$ . Napraviti po jednu podmornicu u svakoj veličini (od 1 do 5). Voditi računa da se cela podmornica nalazi na tabli (podmornica veličine 3 ne može početi na polju npr.  $[1, 8]$  i prostirati se nadesno i slično) i da nema preklapanja dve podmornice (na jednom polju može se naći deo samo jedne podmornice - npr. podmornice  $[3, 1] - [3, 4]$  (4) i  $[1, 3] - [5, 3]$  (5) preklapaju se na polju  $[3, 3]$ ). Ova inicijalizacija je uvek uspešna, odnosno, uvek vraća `true`.
8. Napraviti klasu `PotapanjePodmornica` koja nasleđuje `Application` klasu biblioteke `javafx` i izgleda kao na slici 1.
9. Na klik dugmeta `Inicijalizuj` prave se dva igrača, jedan čovek i jedan računar, i smeštaju u promenljive tipa `Igrac`. Svakom postaviti suprotnog igrača za protivnika. U elementu `TextArea` ispisati podmornice prvog igrača, a onda i drugog. Ispis treba da bude sortiran prema pravcu tako da prvo idu podmornice koje se prostiru nadesno, a onda one koje se prostiru nadole, i rastuće prema poljima (prvo po x, pa po y). U datoteci `pobede.txt` nalazi se istorija partija. Svaki red sadrži dva cela broja koja označavaju pobeđe čoveka, odnosno računara, redom. Učitati datoteku i upisati rezultate poslednje partije u odgovarajuće donje `TextField` elemente (pogledati sliku 1). Pretpostaviti da su podaci u datoteci ispravni.
10. Klik na dugme `Odigraj potez` simulira jedan potez igrača. Potrebno je pozivati metod `odigrajPotez` nad oba igrača. Prvo treba odrediti polje koje se gađa. Čovek unosi u elemente `TextField` koordinate polja koje želi da gađa, a za računar ih odrediti nasumično (iz intervala  $[0, 9]$ ). Ispisati ishod svakog poteza - ko je gađao koje polje i da li je nešto pogodeno (pogledati sliku 2). Dodatno, treba proveriti da li je neki igrač pobedio (pozivom metoda `pobednik()` za svakog). Ukoliko jeste, uvećati vrednost u odgovarajućem `TextField` elementu za jedan, a u element `TextArea` ispisati ko je pobednik (pogledati sliku 2).
11. Na klik dugmeta `Sačuvaj` u datoteku `pobede.txt` dopisati vrednosti koje se nalaze u elementima `TextField`, ukoliko ima izmena u rezultatu, u suprotnom prekinuti akciju.

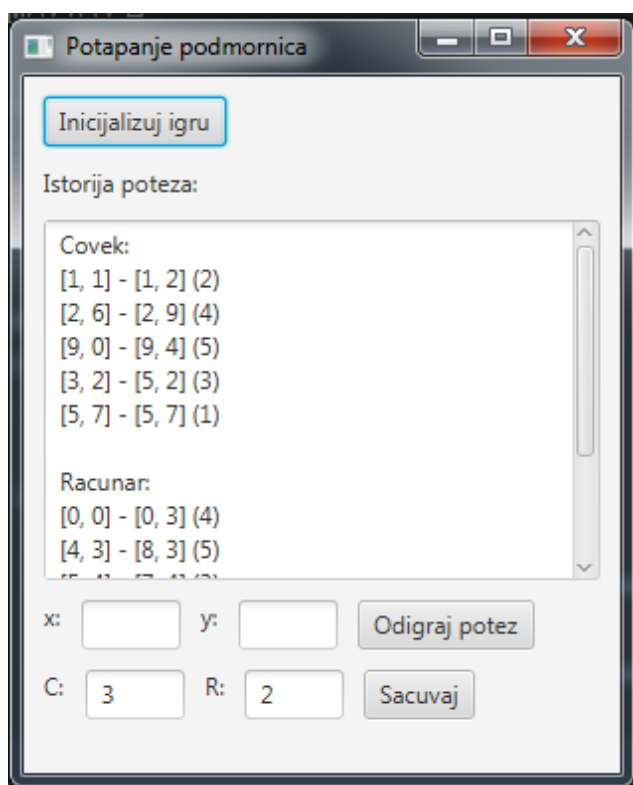
Dozvoljeno je proširiti klase dodatnim atributima i metodama kako biste realizovali prethodno navedene zahteve.

[podmornice.txt]

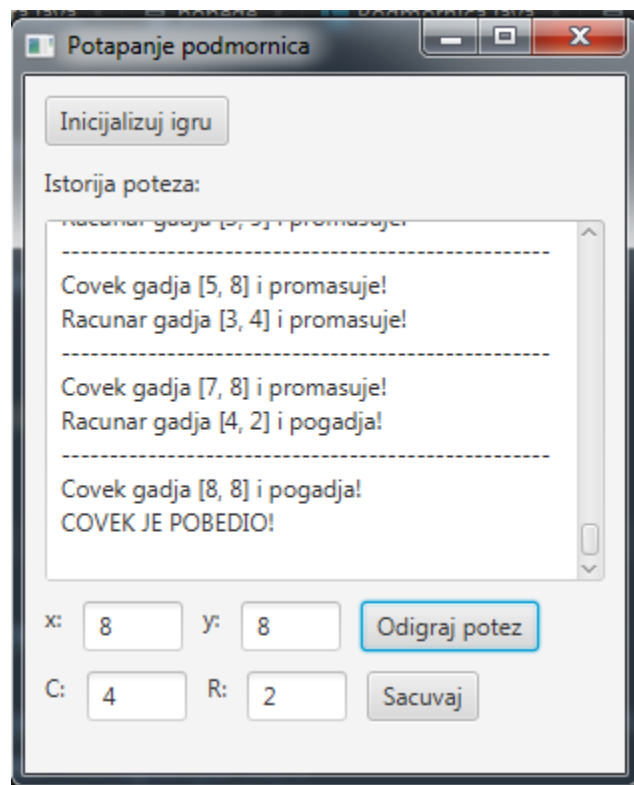
```
1, 1, r, 2
5, 7, d, 1
2, 6, r, 4
3, 2, d, 3
9, 0, r, 5
```

[pobede.txt]

```
1 0
2 0
2 1
2 2
3 2
```



Slika 1: Stanje nakon inicijalizacije



Slika 2: Stanje nakon odigrane partije