

Objektno-orijentisano programiranje, ispit JUN2
Matematički fakultet, školska godina 2023/2024

Napomena: Na Desktop-u napraviti direktorijum pod imenom `oop_InicijaliAsistenta.ImePrezime.Alas` (npr. `oop_IA.Marko.Markovic_mr22101`). Pokrenuti *Intellij Idea* i u napravljenom direktorijumu napraviti projekat sa istim nazivom. U napravljenom projektu, paket nazvati **resenje**.

Inicijali asistenata: 2R1 grupe - JB, 2R2 grupe - LS, MNVL grupe - MK.

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

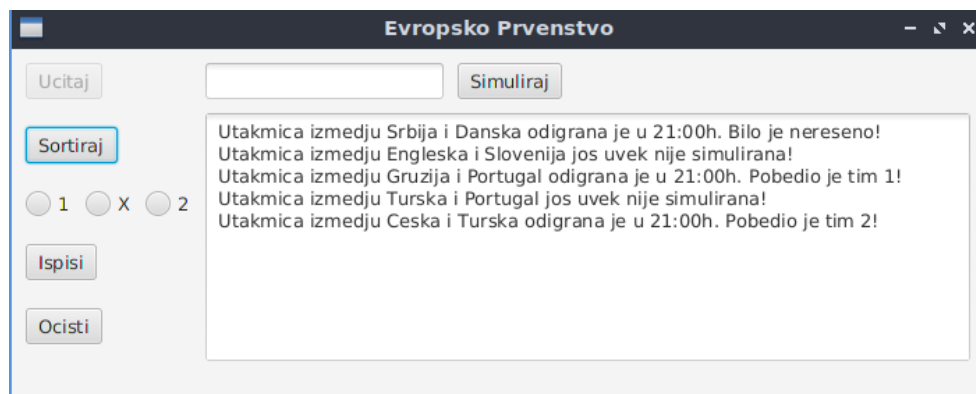
U tekstu je dat opis klasa, njihovih atributa i metoda. **Dozvoljeno** je dodati nove attribute, klase, metode, enume, interfejsu u slučaju da olakšavaju implementaciju i/ili poboljšavaju kvalitet koda i slično.

Da bi se uspešno položio ispit potrebno je osvojiti **barem 35 poena**.

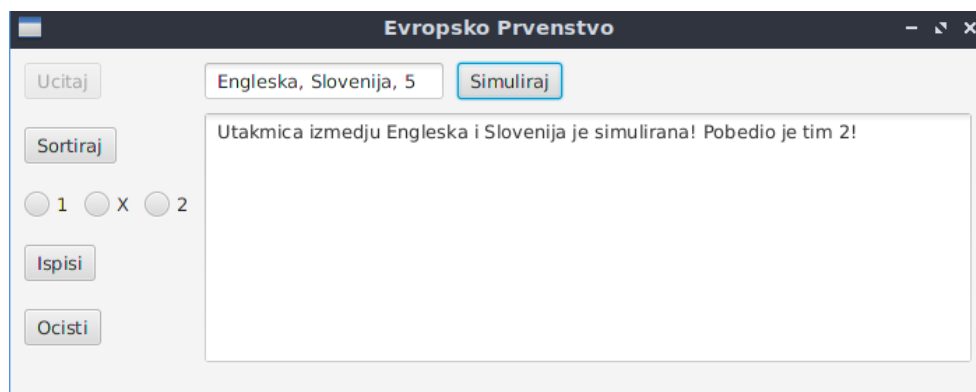
1. [4p] Definirati nabrojivi tip `RezultatUtmice` čije su vrednosti: `TIM1`, `X` i `TIM2`.
 - Definirati statički metod `RezultatUtmice napraviIzRezultata(String rezultat)` koji na osnovu argumenta vraća odgovarajući rezultat. Argument se zadaje u formatu `golovi:golovi2`. U slučaju greške izbaciti izuzetak klase `IllegalArgumentException` sa odgovarajućom porukom.
 - Predefinirati metod `toString` koji u slučaju da je pobedio tim 1, vraća `Pobedio je tim 1!`, analogno i za tim 2, a u slučaju nerešenog rezultata vraća `Bilo je nerešeno!`.
2. [3p] Definirati apstraktnu klasu `Utmica` koja ima polja `tim1 (String)`, `tim2 (String)` i `vremeUtmice (String)`, kao i apstraktan metod `RezultatUtmice getResultat()`. Definirati odgovarajući konstruktor.
3. [3p] Definirati klasu `OdigranaUtmica` koja nasleđuje klasu `Utmica`, i implementira apstraktan metod. Dodati polje `rezultat (RezultatUtmice)`. Definirati konstruktor i predefinirati metod `toString` tako da ispis bude oblika: `"Utmica između Srbije i Danske odigrana je u 21:00h. Pobedio je tim1!"`.
4. [2p] Definirati interfejs `SimulacijaUtmice` koji ima apstraktan metod `void simulirajIgranjeUtmice(int n)`.
5. [8p] Definirati klasu `SimuliranaUtmica` koja nasleđuje klasu `Utmica` i implementira interfejs `SimulacijaUtmice`, kao i apstraktan metod `getResultat()`. Dodati polje `rezultat (RezultatUtmice)` koje će biti nepoznato dok se ne pokrene simulacija utmice. Implementirati metod `void simulirajIgranjeUtmice(int n)` tako da svaka ekipa ima `n` prilika da postigne gol, koje se smenjuju naizmenično, dok u svakoj prilici ekipa ima 30% šanse da postigne gol. Na osnovu broja postignutih golova inicijalizovati polje `rezultat` na odgovarajući način. Definirati odgovarajući konstruktor i konstruktor kopije. Predefinirati metod `toString` tako da ispis bude u formatu kao na slici 1.
6. [5p] Definirati klasu `KomparatorUtmica` koja implementira interfejs `Comparator<Utmica>` tako da definiše sledeći kriterijum poređenja:
 - u sortiranom redosledu prvo idu odigrane pa simulirane utmice
 - među sobom utmice se porede prema rezultatu, tako da prvo idu one na kojima je pobedio domaćin, pa one gde je bilo nerešeno, i na kraju one gde su pobedu odneli gosti
 - ukoliko rezultat za utmicu ne postoji, ona ide na kraj poretka
7. [6p] Definirati klasu `EvropskoPrvenstvo` koja ima polja `mestoOdrzavanja (String)`, `datumOdrzavanja (String)`, `utmice (List<Utmica>)`. Definirati odgovarajući konstruktor kao i getere. Definirati metod `void dodajUtmicu(Utmica u)` koji dodaje utmicu u listu. Definirati metod `String simulirajUtmicu(String tim1, String tim2)` koji izvršava simulaciju utmice između dva tima, ukoliko je to moguće, i vraća odgovarajuću poruku, kao na slikama 2, 3, 4. Predefinirati metod `toString` tako da bude oblika kao na slici 5.
8. [10p] Definirati klasu `Main` koja nasleđuje klasu `Application` biblioteke `javafx`. Klasa ima atribut `prvenstvo (EvropskoPrvenstvo)` i izgleda kao na slikama. Dimenzije prozora su npr. `500x250`.
9. Klikom na dugme:
 - [2p] Ocisti čiste se tekstualna polja, označeni dugmići i `TextArea`.

- [9p] Učitaj učitavaju se podaci iz datoteke `prvenstvo.txt`. Prva linija sadrži informacije o prvenstvu i ona je data u obliku: `Mesto održavanja, Datum održavanja`. Linije nakon nje predstavljaju podatke o utakmicama, koje je potrebno učitati u listu, a dati su u formatu `Tim 1, Tim2, Vreme utakmice, Rezultat`. Rezultat je opcionalna vrednost, i njega imaju samo odigrane utakmice. Nakon učitavanja ispisati utakmice iz liste u `TextArea`, kao na slici 1.
- Obezbediti da se učitavanje može uraditi tačno jednom.
- Pretpostaviti da je sadržaj datoteke ispravan.
- [4p] Sortiraj sortiraju se utakmice u skladu sa već definisanim poretkom, a potom ispisuju u `TextArea`.
- [4p] Ispisi ispisuju se u `TextArea` sve utakmice čiji rezultat odgovara izabranom `RadioButton`-u. Omogućiti da istovremeno može da se selektuje samo jedno dugme.
- [5p] Simuliraj simulira se igranje utakmice za timove čija se imena, kao i broj šansi pri simulaciji navode u okviru `TextField`-a, u formatu `Tim1, Tim2, n`. Nakon toga odgovarajuću poruku o simulaciji ispisati u `TextArea` kao na slikama 2, 3, 4.

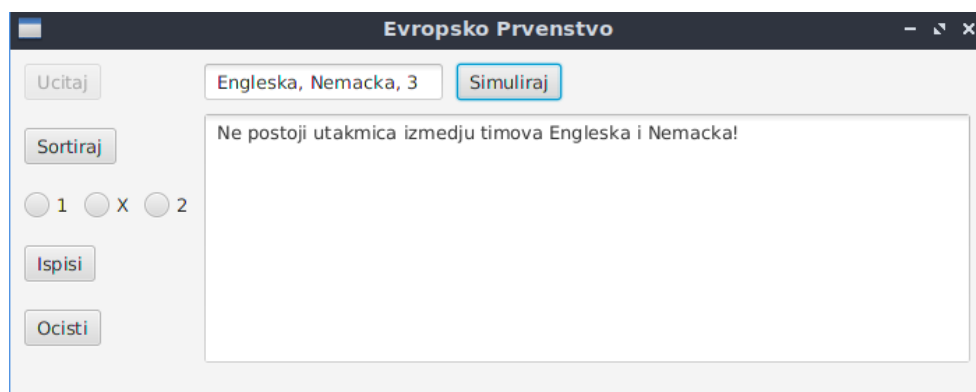
Nemacka, 14.6.2024
 Srbija, Danska, 21:00, 0:0
 Engleska, Slovenija, 21:00
 Gruzija, Portugal, 21:00, 2:0
 Turska, Portugal, 18:00
 Ceska, Turska, 21:00, 1:2



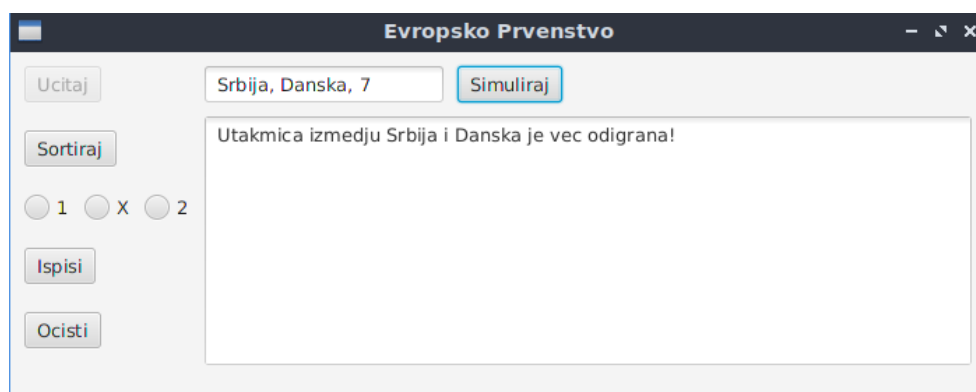
Slika 1: Učitavanje



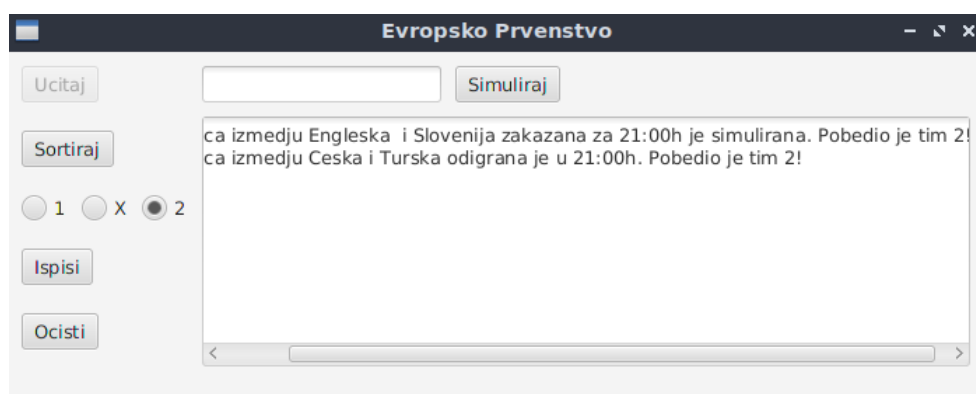
Slika 2: Simulacija



Slika 3: Simulacija



Slika 4: Simulacija



Slika 5: Ispis nakon simulacije