

Simulacija igre - Kviz

CS101

Uvod u objektno orijentisano programiranje

Projektna dokumentacija

02.01.2021

Student: Darko Tasevski 4374

Asistent: Jovana Jović

Sadržaj

[1. Predlog teme 3](#_Toc491866984)

[2. Opis funkcionalnosti 3](#_Toc491866985)

[3. Struktura aplikacije 3](#_Toc491866986)

[4. Korisničko uputstvo 3](#_Toc491866987)

[5. Zaključak 3](#_Toc491866988)

[6. Literatura 3](#_Toc491866989)

# Predlog teme

Simulacija kviza, koja obuhvata učitavanje pitanja iz eksternog fajla, kao i učitavanja podataka o sačuvanim igračima i njihovim rezultatima ukoliko takav fajl postoji. Kviz ima mogućnost igre za više igrača u toku jedne sesije.

# Opis funkcionalnosti

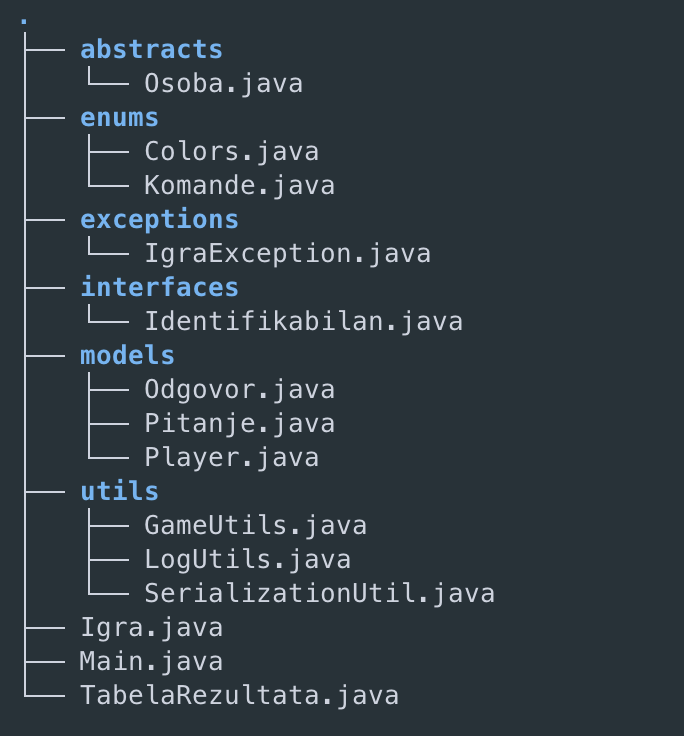
Pri pokretanju aplikacije, učitavaju se pitanja i podaci o igračima iz eksternih fajlova. Ukoliko eksterni fajl sa pitanjima ne postoji, pitanja se učitavaju iz programa, i kreira se eksterni fajl koji će se koristiti pri sledećem pokretanju programa. Ukoliko se program pokreće prvi put, eksterni fajl o igračima ne postoji, ali nakon završenog prvog kviza i izlaska iz programa, biće kreiran i učitan pri sledećem pokretanju programa.

Igra se sastoji od 10 pitanja, koja su bazirana na 20 sačuvanih pitanja, i pri svakom pokretanju igre, ovih dvadeset pitanja su nasumično izmešana i koristi se prvih deset za igru. Svako pitanje ima nekoliko ponudjenih odgovora, i igrač odgovara na pitanje tako što unosi redni broj pitanja preko komandne linije.  
  
Nakon završene igre, nudi se pokretanje igre ponovo, pri čemu se može izabrati drugo korisničko ime i napraviti novi unos u eksternom fajlu sačuvanih rezultata.

# Struktura aplikacije

Kod aplikacije se nalazi u **src** direktorijumu, u paketu **main.** Za strukturu aplikacije sam koristio [Maven Standard Directory Layout guide](http://maven.apache.org/guides/introduction/introduction-to-the-standard-directory-layout.html) s tim da kod nisam stavio u java direktorijum, sa obzirom da je program veoma jednostavan.

U glavnom paketu main se nalaze sledeće klase, enumi, i interfejsi:



Klase, Igra, TabelaRezultata i Main su glavne klase:

* Igra klasa sadrži osnovnu logiku za pokretanje kviza i kreiranje nove igre
* TabelaRezultata sadrži osnovnu logiku za sortiranje i prikazivanje liste rezultata
* Main klasa je glavna klasa projekta, i sadrži osnovnu logiku za obradu korisničkog unosa preko komandne linije

Direktorijum abstracts, sadrži abstraktnu klasu Osoba, koju extend-uje klasa Player.

Direktorijum enums, sadrži Colors i Komande enume.

* Colors je enum sa listom enum polja koja imaju svoje vrednosti. Ovaj enum se koristi u klasi LogUtils kao vrednost boja za štampanje teksta, tako što se ubacuju ANSI escape kodovi za obojeni tekst. Moguće da je ova funkcionalnost ne radi na računarima koji koriste Windows OS.
* Komande enum, sadrži listu enum polja sa String vrednostima mogućih komandi u programu.

Direktorijum exceptions sadrži IgraException izuzetak koji je custom exception klasa. Na kraju projekta nisam imao nikakve potrebe za specijalnim izuzecima, tako da se ovaj izuzetak ne koristi nigde.

Direktorijum interfaces sadrži Identifikabilan interfejs, koji označuje da klasa može biti identifikovana prema int id polju.

Direktorijum models sadrži osnovne model klase koje se koriste u projektu:

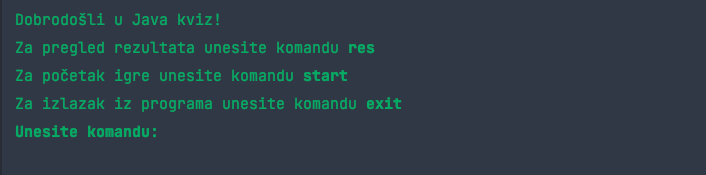
* Odgovor klasa - ova klasa koristi Identifikabilan interfejs i može se serijalizovati, tj. čuvati kao objekat u eksternom fajlu
* Pitanje klasa - ova klasa koristi Serializable, i Identifikabilan interfejse, može se čuvati kao objekat u eksternom fajlu
* Player klasa - ova klasa nasledjuje abstraktnu klasu Osoba, i koristi Serializable, Comparable<Player>, i Identifikabilan interfejse, tako da se može čuvati eksternim fajlovima kao objekat, uporedjivati sa ostalim Player klasama i može biti identifikovana prema id.

Direktorijum utils sadrži razne klase koje se koriste u projektu za logovanje, čitanje i pisanje podataka u eksterne fajlove, validaciju i generisanje podataka.

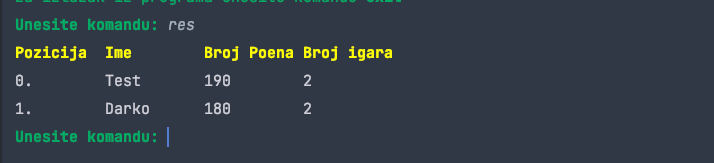
* GameUtils klasa se koristi za razne funkcije u programu, kao što su štampanje poruka o komandama, validaciji komandi, prosledjivanje učitanih podataka iz eksternih fajlova drugim klasama, i generisanje pitanja i odgovora ukoliko ona ne postoje u eksternom fajlu. Ova klasa sadrži samo statičke metode, i uz to je i final klasa, što znači da ne može biti nadogradjivana i nasledjivana. Ova klasa ima private konstruktor, i tako onemogućava inicijalizaciju klase.
* LogUtils klasa, koristi System.out.println i System.out.print metode da štampa razne poruke u ostalim klasama, a koristeći Colors enum vrednosti, u mogućnosti je da te poruke štampa u raznim bojama i u regular i bold text varijantama. Ova klasa ima private konstruktor, i tako onemogućava inicijalizaciju klase, i sadrži samo statičke metode, a uz to je i final klasa, što znači da ne može biti nadogradjivana i nasledjivana.
* SerializationUtil je glavna klasa za rad sa eksternim fajlovima,tj. upisivanje i čitanje objekata iz istih. Ova klasa je generička klasa, koja prima kao argument tip podatka čiji podaci trebaju da se pročitaju ili upišu u ekterni fajl. Ova klasa ima ključnu reč final, i stoga ne može biti nasledjivana ili extend-ovana od strane ostalih klasa.

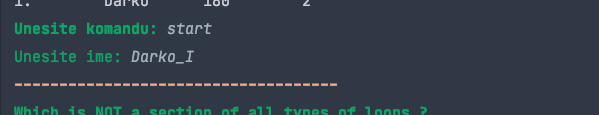
# Korisničko uputstvo

**1.** Pri pokretanju programa, korisnika dočekuje ovaj tekst:

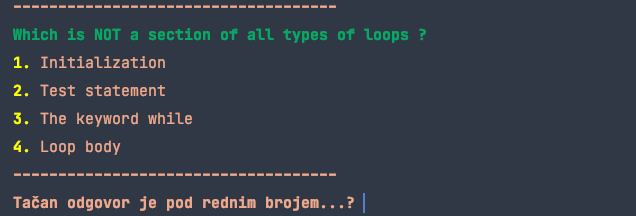
**

**2.** Ukoliko korisnik izabere da vidi rezultate, komandom **res**, prikazuje se sledeći tekst:

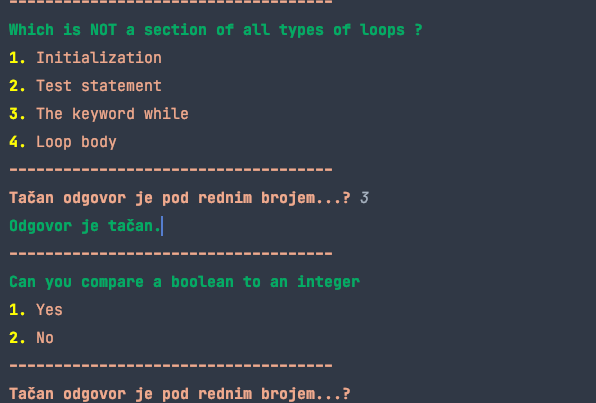
**gde prikazujemo listu igrača koji su prethodno pokretali i završili kviz, sa njihovim imenom, brojem poena i brojem odigranih partija. Broj poena se računa tako što se broj tačnih odgovora pomnoži sa 10.  
  
**3.** Ukoliko korisnik nakon ovoga pokrene igru komandom **start**, prvo pitamo za korisničko ime pre pokretanja igre:

**

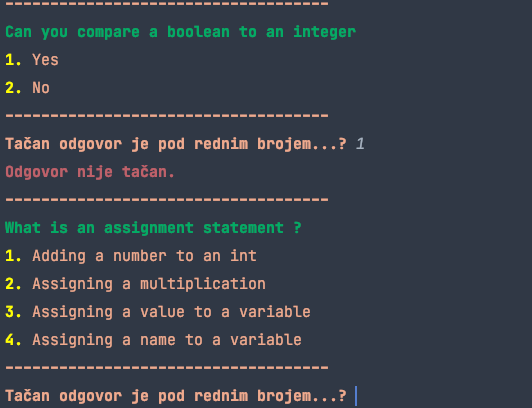
i nakon unosa imena (koje mora biti izmedju 2 i 10 karaktera), pokrećemo igru:

**

Ukoliko je odgovor tačan prikazujemo tekst *" Odgovor je tačan."* i prikazujemo sledeće pitanje:

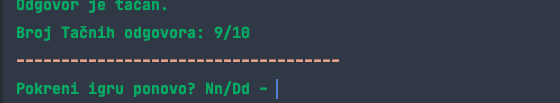
**

Ukoliko odgovor nije tačan, prikazujemo *" Odgovor nije tačan."* i prikazujemo sledeće pitanje:

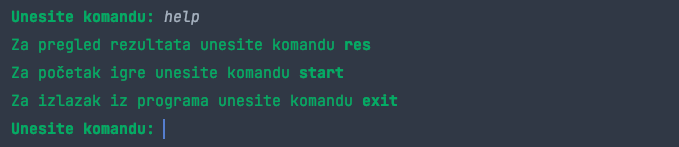
**

Ponavljamo isto za sva pitanja do kraja kviza (10 pitanja). Korisnik ne može da unese broj veći/manji od broja ponudjenih odgovora. Na primer, na pitanje iznad, ukoliko korisnik unese broj veći od 5 ili manji od 1, ispisaće se greška na ekranu: *" Uneta vrednost mora biti ceo broj izmedju 1 i 4. Pokušajte ponovo:".*

Nakon što su dati odgovori na sva pitanja, prikazujemo rezultat kviza, tj. broj tačnih odgovora prema ukupnom broju pitanja, i pitamo igrača da li želi da pokrene igru od početka:

**

Ukoliko, korisnik odgovori karakterom **D** ili **d**, igra se pokreće ponovo (korak 3.), a ukoliko se dobije **N**, **n**, ili bilo koja druga vrednost vraćamo se na početni ekran (1.).  
  
Unosom komande **help** dok je igrač na početnom ekranu, može se videti lista mogućih komandi:

**

Unosom komande **exit**, program se završava.

# Zaključak

U projektu su kombinovana znanja stečena iz lekcija CS101, kao i par stvari na koje sam naleteo na internetu, a za koje mislim da su se dobro uklopile u projekat, kao što su obojeni logovi i generičke klase.

Cilj aplikacije je da korisnik proveri svoje osnovno znanje Java programskog jezika, i nakon nekoliko odigranih kvizova mogu da kažem da sam naučio par stvari. Jedan od ciljeva je takodje bio da napravim prvi veći projekat u Javi, i sa obzirom da do pre par meseci nisam znao da pišem kod u Javi, zadovoljan sam krajnjim rezultatom.  
  
U smislu poboljšanja, imam par stvari na umu:

* opcija da korisnik doda svoja pitanja i odgovore na listu pitanja
* dodavanje polja objašnjenje na Pitanje, koje bi bilo prikazano ukoliko igrač odgovori netačno na neko pitanje
* izvorni kod programa, najverovatnije se može napisati i struktuirati bolje
* prevod pitanja na srpski jezik
* Čuvanje pitanja, odgovora i podataka o igračima u JSON formatu, i/ili čitanje istih sa web-a

# Literatura

- [Stack Overflow](https://stackoverflow.com/questions/tagged/java)

- CS101 beleške

- [Oracle Java dokumentacija](https://docs.oracle.com/javase/8/docs/)