

Objektno programiranje, Ispit SEP2

Matematički fakultet

Školska godina 2018/2019

Napomena: Na Desktop-u napraviti direktorijum pod imenom `oop_Asistent_Prezime_Ime_Indeks` (npr. `oop_NM_Peric_Pera_mi12082`). Pokrenuti IntelliJ Idea i u napravljenom direktorijumu napraviti projekat sa istim nazivom. U napravljenom projektu, **napraviti paket sa istim nazivom**.

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.

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

Vreme za rad: **2.5 sata**

Inicijalni asistenata: Biljana - BS, Anja - AB, Ivan - IR, Nemanja - NM, Rastko - RD

1. Napraviti klasu `Rec` koju karakterišu atributi

- `String rec` - reč koja se traži
- `List<Character> isprobanaSlova` - sva isprobana slova za datu reč
- `int tezina` - procenjena težina pronalaska reči.

Implementirati statičku funkciju `tezinaReci` koja na osnovu reči računa njenu težinu po sledećoj formuli: $\text{procenatSuglasnikaUReci} * (1 - (\text{dužinaReči} / (\text{dužinaReči} + 10)))$. Težina 1 odgovara intervalu $[0\%, 33\%]$, 2 intervalu $[33.3\%, 66.6\%]$, a 3 ostatku vrednosti. Implementirati konstruktor koji prihvata vrednost atributa `rec`. Na osnovu vrednosti za reč računa se težina reči, a isprobana slova se postavljaju na praznu listu. Implementirati potrebne `get` metode. Implementirati funkciju `public String pogodjenaSlovaReci()` koja vraća reč sa podvlakama umesto slova koja nisu pogođena (kao na slici 2).

Implementirati `toString` metod za predstavljanje reči u obliku:

reč, težina

2. Napraviti apstraktnu klasu `Igrac` koja predstavlja jednog igrača igre. Klasa se karakteriše sledećim atributima:

- `String ime` - ime Igrača
- `int nivo` - nivo Igrača (broj koji najmanje može biti 1)
- `int brOdigranihPartija` - broj odigranih partija
- `int brPobedenihPartija` - broj pobedenih partija

Implementirati konstruktor koji prihvata sve navedene attribute. Implementirati potrebne `get` metode. Klasa poseduje apstraktan metod `public abstract char odaberiSlovo(Rec trazenaRec)`.

Implementirati `toString` tako da igrače predstavlja na sledeći način:

```
[nivo] ime, procenatPobedenihPartija%  
[1v13] Filip, 20.00%  
[1v19] Deep Blue, 66.67%  
[1v142] AlphaZero, 99.99%
```

3. Napraviti klasu `LjudskiIgrac` koja nasleđuje klasu `Igrac`. Ljudski igrač poseduje atribut `izabranoSlovo` (`char`) čije vrednosti mogu biti slova engleskog alfabeta. Implementirati metod `void setIzabranoSlovo(char c)` koja postavlja slovo ako je deo engleskog alfabeta, a inače ne radi ništa. Implementirati metod `public char odaberiSlovo(Rec trazenaRec)` koji vraća slovo u zavisnosti od vrednosti atributa `izabranoSlovo`.

4. Napraviti klasu `KompjuterskiIgrac` koja nasleđuje klasu `Igrac`. Klasa poseduje privatno statičko polje `random` (`Random`).

Implementirati metod `public char odaberiSlovo(Rec trazenaRec)` tako da sa jednakom verovatnoćom vraća neko od slova koja nisu među trenutno isprobanim slovima reči `trazenaRec`.

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 `VesalaArena` koja nasleđuje `Application` klasu biblioteke `javafx` i izgleda kao na slici 1.

Obezbediti da u svakom trenutku može biti selektovano tačno jedno radio dugme. Na početku kreirati jednog ljudskog i jednog kompjuterskog igrača sa fiksnim vrednostima u kodu.

Na klik dugmeta `Ucitaj` iz datoteke `reci.txt` učitavaju se reči i smeštaju u mapu `Map<Rec, List<String> >` `recnik` koja slika reči u listu nagoveštaja. Možemo imati proizvoljan broj nagoveštaja, ali znamo da će postojati bar jedan. Nagoveštaji u sebi neće sadržati zareze. Format:

```
rec, nagovestaj1, nagovestaj2, ... nagovestajN.
```

Potom sve reči sa dodeljenim nagoveštajima treba ispisati u gornji `TextArea` element na sledeći način: reči moraju biti uređene rastuće po težini, a potom leksikografski rastuće (uređenje elemenata prilikom ispisa omogućiti implementacijom odgovarajućeg interfejsa za poređenje elemenata). Pogledati sliku 1 za primer ispisa.

Na klik dugmeta `Zapocni igru` pokreće se igra. Ljudski igrač uvek igra prvi. Bira se nasumična reč iz rečnika u zavisnosti od izabrane težine i ispisuje se odgovarajući tekst u donje tekstualno polje.

Na klik dugmeta `Unesite slovo` ljudskom igraču se postavlja odgovarajuće slovo u zavisnosti od toga šta je u uneo korisnik (koristiti metod `void setIzabranoSlovo(char c)`).

Na klik dugmeta `Odigraj potez` dešava se simulacija jednog poteza (slika 2). Prvo se proverava ko je trenutno na redu i da li su sva slova reči pronađena. Ako je to slučaj, za porednika se proglašava protivnik porukom `Pobedio je imePobednika`.

Ako još nisu pronađena sva slova polimorfno se poziva metod `odaberiSlovo` nad trenutnim igračem. Ako se izabrano slovo nalazi u traženoj reči, ažurira se ispis pronađenih slova i isti igrač ostaje na redu za sledeći potez. Ako se izabrano slovo ne nalazi u traženoj reči, na red dolazi drugi igrač.

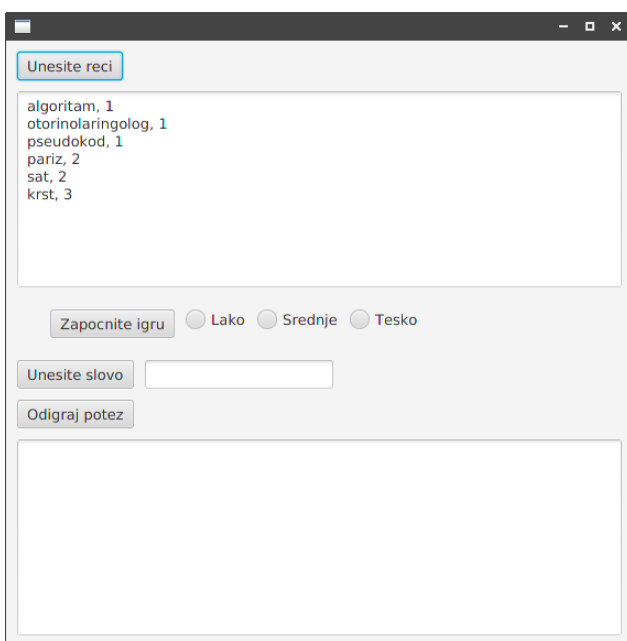
Nakon svakog poteza treba ažurirati broj isprobanih slova i traženu reč na odgovarajući način. Takođe, treba ispisati poruku o potezu na dnu tekst oblasti (`Slovo se (ne) nalazi u traženoj reči`).

Nakon svakog četvrtog izabranog slova, ako reč još nije pronađena treba ispisati nagoveštaj u polje za igru. To znači da se posle 4. izabranog slova ispisuje prvi nagoveštaj u nizu, pa nakon 8. drugi, itd. dok ima nagoveštaja.

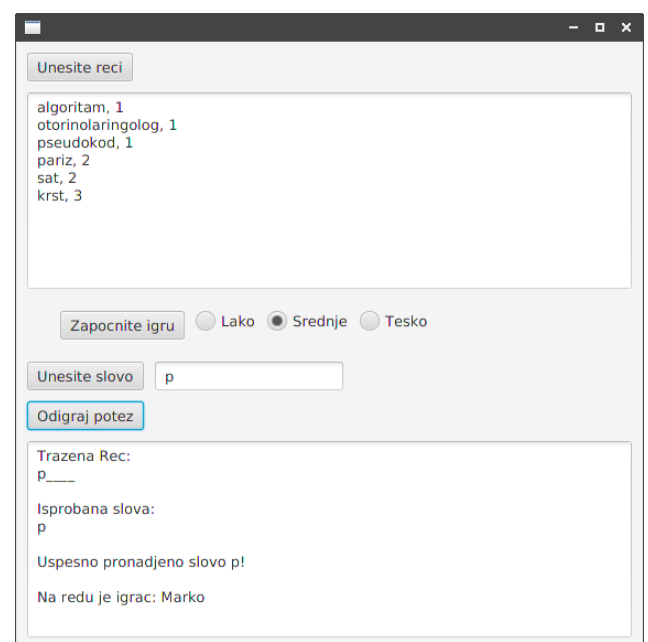
[reči.txt]

reč, nagoveštaji

```
pariz, U celom gradu postoji samo jedan stop znak, Le capitale de la France
sat, Deo naopako okrenute vage
otorinolaringolog, Čuje a ne vidi, Sluša a ne misli
krst, C++, Plus i plus daju plus, Plus nije samo plus, +
pseudokod, Najveća zmiya u kraju, Nije anakonda
algoritam, Logaritam, Logaritam al' za malo, Logaritam u blenderu
```



Slika 1: Nakon unosa reči



Slika 2: Nakon jednog odigranog poteza