Objektno-orijentisano programiranje, Ispit JUN2
Matematički fakultet
Školska godina 2020/2021

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**
Inicijalini asistenata: `Nevena - NC`, `Ognjen - OM`, `Denis - DA`, `Filip - FV`

U tekstu je dat opis klasa, njihovih atributa i metoda. **Dozvoljeno** je (i ohrabrujemo Vas) dodati nove atribute, klase, metode, enume, interfejse 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 35 poena.

1. Napraviti apstraktnu klasu `Memorija` koju karakterišu atributi `naziv` (`String`) i `kolicinaMemorije` (`int`). Klasa sadrži i apstraktan metod `double cena()`. Implementirati konstruktor koji prima vrednosti za sva polja, potrebne `get` metode, kao i metod `toString()` koji vraća nisku kao u test primeru

   ```
   naziv - kolicinaMemorije GB
   ```

2. Napraviti klasu `Hdd` koja nasleđuje klasu `Memorija` i predstavlja hard disk zasnovan na okretanju pločice. Hdd se dodatno karakteriše poljem `brzina` (`int`) koje označava brzinu okretanja pločice (meri se u RPM - Revolutions Per Minute). Implementirati konstruktor koji prima vrednosti za polja natklase (`naziv` i `kolicinaMemorije`) i pomenutu brzinu okretanja. Implementirati apstraktni metod koji izračunava cenu telefona. Cena se računa po formuli: $kolicinaMemorije * brzina/1000$.
   Dodati neophodne `get` metode kao i `toString` koji vraća nisku kao u test primeru

   ```
   [HDD]: naziv - kolicinaMemorije GB, brzina RPM, cena
   ```

3. Napraviti nabrojivi tip `SsdTip` koja može imati vrednosti `FLASH` i `DRAM`. Implementirati konstruktor, get metod za polje statički metod `SsdTip napraviTip(String s)` koji konstruiše odgovarajuću nabrojivu vrednost ako se prosledi vrednost za `s` koja je `"FLASH"` ili `"DRAM"`. Inače, vratiti vrednost `FLASH`.

4. Napraviti klasu `Ssd` koja nasleđuje klasu `Memorija`. Klasa se karakteriše poljem `tip` (`SsdTip`). Implementirati konstruktor koji prima vrednosti za polja natklase (`naziv` i `kolicinaMemorije`) i pomenut tip ssd memorije. Implementirati apstraktni metod koji izračunava cenu. Cena se računa po formuli $kolicinaMemorije * 10$, u slučaju FLASH memorije, odnosno $kolicinaMemorije * 20$, u slučaju DRAM memorije. Dodati neophodne `get` metode kao i `toString` koji vraća nisku kao u test primeru

   ```
   [SSD]: naziv - kolicinaMemorije GB, tip, cena
   ```

5. Napraviti klasu `KupovinaMemorije` koja nasleđuje `Application` klasu biblioteke `javafx` i izgleda kao na slici 1. Obezbediti da je unapred selektovano prvo radio dugme i da u svakom trenutku može biti selektovano tačno jedno radio dugme. Klasa `KupovinaMemorija` sadrži kao atribut `List<Memorija> memorije`.U klasi `KupovinaMemorije` implementirati statički generički metod `void ispis(List<T> memorije, TextArea ta)` koji briše sadržaj `TextArea` elementa, a zatim u njega ispisuje sadržaj liste tako da svaki element bude u zasebnom redu.

   Na klik dugmeta `Ucitaj` iz datoteke `memorije.txt` učitavaju se memorije i smeštaju u `List<Memorija>`. Potom se lista sortira tako da u listi prvo idu ssd memorije (u datoteci počinju sa `SSD`), a onda i hdd (u datoteci počinju sa `HDD`) i to nerastuće po kolicini memorije, a ako postoje dve memorije sa istom količinom memorije, rastuće po brzini okretanja (u slučaju hdd), odnosno po tipu (prvo FLASH pa DRAM memorija, u slućaju ssd) - na primer može se koristiti operator `instanceof`. Nakon sortiranja u `TextArea` element se ispisuju memorije u sortiranoj listi, svaki u novom redu (koristeći `toString` nad klasom `Memorija` - pogledati sliku 1).

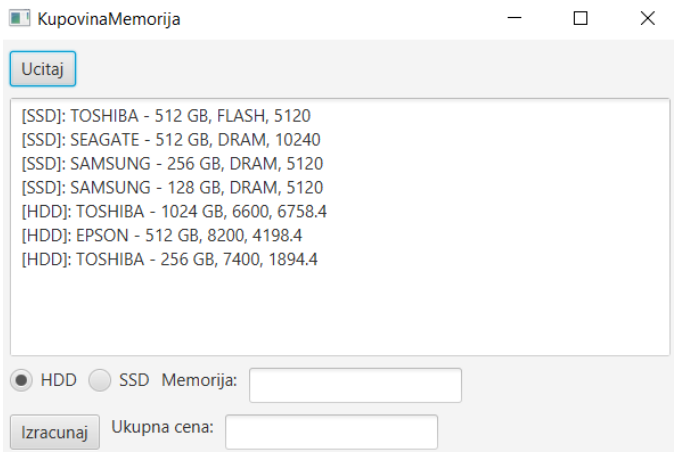
   Na klik dugmeta `Izračunaj` u `TextArea` element ispisati podatke o svim hdd ili ssd memorijama koje zadovoljavaju zadate uslove. U zavisnosti od toga da li je izabrano radio dugme „SSD“ ili „HDD“ iz liste se biraju odgovarajuće memorije za ispis. Dodatno, u polje ,,Memorija” moguće je uneti željenu količinu memorije i treba prikazati samo one memorije koje imaju bar toliku količinu memorije (slika 2). Polje ,,Memorija” ne mora biti

popunjeno, i u tom slučaju uzeti sve memorije odabranog tipa u obzir (slika 3). Na kraju, u polje ,,Ukupna Cena" treba ispisati ukupnu sumu koja treba da bude plaćena za tu vrstu memorije, u formatu kao na slikama 2, 3 i 4. Ako u listi nema memorije izabranog tipa ili sa odabranom količinom memorije, ispisati poruku ,,Nema trazenih hdd memorija!" tj. ,,Nema trazenih ssd memorija!" (slike 4).

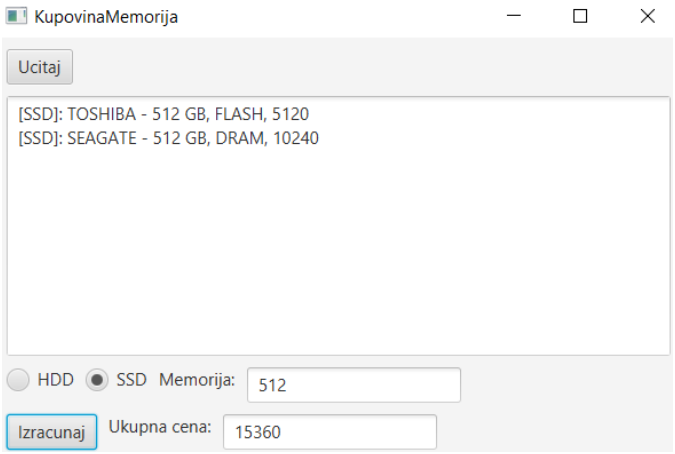
Dozvoljeno je proširiti klase dodatnim atributima i metodama kako biste realizovali prethodno navedene zahteve.

```
[memorije.txt]
HDD, naziv, kolicinaMemorije, brzina
SSD, naziv, kolicinaMemorije, tip
```

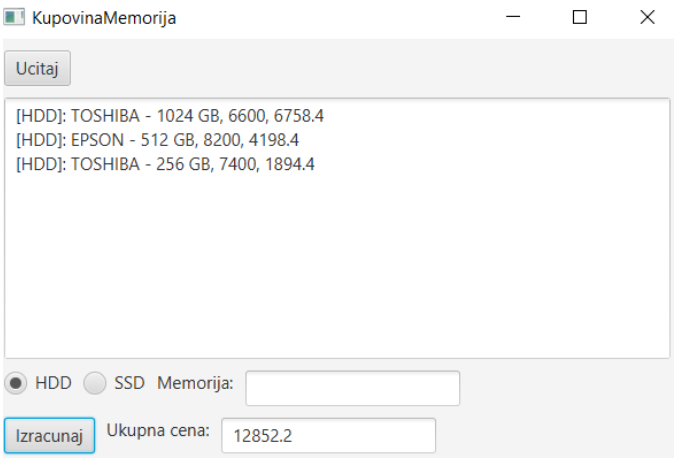
```
HDD, TOSHIBA, 256, 7400
HDD, EPSON, 512, 8200
SSD, TOSHIBA, 512, FLASH
SSD, SAMSUNG, 256, DRAM
HDD, TOSHIBA, 1024, 6600
SSD, SAMSUNG, 128, DRAM
SSD, SEAGATE, 512, DRAM
```



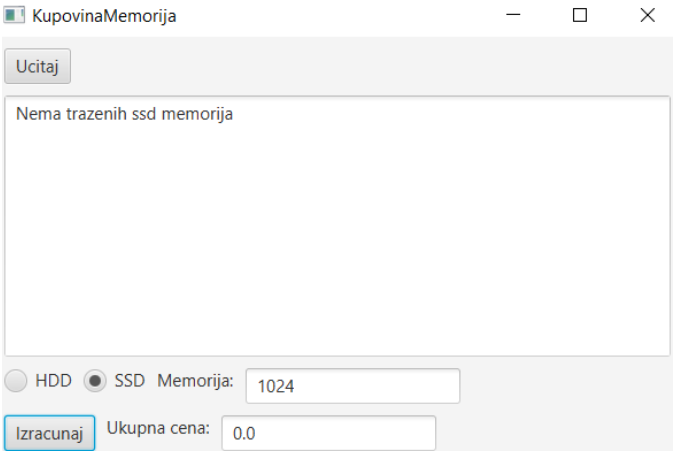
Slika 1: Prikazivanje sadržaja datoteke



Slika 2: Prikazivanje uređaja sa određenom memorijom



Slika 3: Prikazivanje uređaja bez određene memorije



Slika 4: Prikazivanje poruke