

**Objektno-orijentisano programiranje, ispit SEP1**  
*Matematički fakultet, školska godina 2023/2024*

**Napomena:** Na Desktop-u napraviti direktorijum pod imenom `oop_InicijaliAsistenta.Ime_Prezime.Alas` (npr. `oop_IA_Marko_Markovic_mi22101`). Pokrenuti *Intellij Idea* i u napravljenom direktorijumu napraviti projekat sa istim nazivom. Zadatke sačuvati redom u okviru paketa `zadatak1`, `zadatak2` i `zadatak3`.

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

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

Da bi se uspešno položio ispit potrebno je osvojiti **barem 30 poena**.

1. [20p] Implementirati sistem za računanje cena i kupovinu svemirskih brodova.

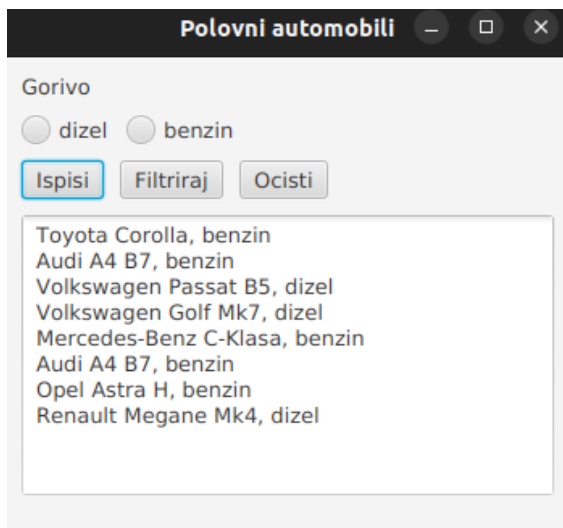
- Definisati apstraktnu klasu **SvemirskiBrod** koja ima:
  - Atribut **ime** koji predstavlja naziv svemirskog broda.
  - Atribut **proizvodjac** koji predstavlja naziv kompanije koja je proizvela svemirski brod.
  - Atribut **snagaMotora** koji je tipa `float`.
  - Apstraktnu metodu **cena()** koja će biti implementirana u podklasama.
- Implementirati tri klase koje nasleđuju apstraktnu klasu **SvemirskiBrod**:
  - **Jednokrilni** sa dodatnim atributom **duzina**. Cena se računa kao zbir dužine i snage motora.
  - **Dvokrilni** sa dodatnim atributima **duzina** i **sirina**. Cena se računa po formuli:
$$(duzina + snagaMotora) * sirina$$
  - **Okrugli** sa dodatnim atributom **poluprecnik**. Cena se računa kao površina kruga pomnožena sa snagom motora.
- Definisati interfejs **Informacije** koji ima metodu **ispisiInformacije()** koja će biti implementirana u svakoj klasi. Metoda **ispisiInformacije()** treba da ispisuje vrednosti atributa i cenu svemirskog broda.
- Implementirati klasu **Cenovnik** koja sadrži listu objekata tipa **SvemirskiBrod** i definisati konstruktor koji kao argument dobija niz svemirskih brodova. U klasu **Cenovnik** dodati sledeće metode:
  - **ukupnaCena()** koja vraća ukupnu cenu svih svemirskih brodova u listi.
  - **kupiBrod(String ime)** koja kreira fajl "*ime-svemirskog-broda.txt*". U fajl upisati iste informacije koje ispisuje metoda **ispisiInformacije()** za kupljeni svemirski brod. Kupljeni svemirski brod zatim izbaciti iz liste. Pretpostaviti da ne postoje dva svemirska broda sa istim imenom.
- U klasi **Main** napisati kod koji koristi ove klase i interfejsu:
  - Kreirati instancu klase **Cenovnik**.
  - Dodati nekoliko svemirskih brodova različitog tipa u cenovnik korišćenjem konstruktora koji prihvata niz.
  - Pozvati metodu **ukupnaCena()** i ispisati rezultat.
  - Iterirati kroz cenovnik i za svaki svemirski brod pozvati **ispisiInformacije()**.
  - Pozvati metodu **kupiBrod(String ime)** za neki svemirski brod sa cenovnika.

2. [20p] Implementirati aplikaciju u klasi **PolovniAutomobili** korišćenjem JavaFX biblioteke koja izgleda kao na slici 1. U datoteci **automobili.txt** date su informacije o modelu automobila kao i gorivu koje automobil koristi.

- Klikom na dugme **Ispisi** ispisuje se sadržaj datoteke. Obezbediti da se učitavanje iz datoteke vrši tačno jednom.
- Klikom na dugme **Filtriraj** filtrira se pročitani sadržaj datoteke u zavisnosti od označenog goriva koje automobil koristi. Ukoliko gorivo nije označeno, u prozoru za tekst ispisati poruku o grešci.
- Klikom na dugme **Obrisi** čisti se tekstualni sadržaj i označeni dugmići.

Sadržaj datoteke **automobili.txt**:

```
Toyota Corolla, benzin
Audi A4 B7, benzin
Volkswagen Passat B5, dizel
Volkswagen Golf Mk7, dizel
Mercedes-Benz C-Klasa, benzin
Audi A4 B7, benzin
Opel Astra H, benzin
Renault Megane Mk4, dizel
```



(a) Ispisi



(b) Filtriraj

Slika 1: Polovni automobili

3. [25p] U klasi **Stablo** implementirati binarno pretraživačko stablo čiji čvor može čuvati vrednost proizvoljnog tipa.

- Definirati metod **void dodajElement(T element)** koji dodaje čvor u stablo sa zadatom vrednošću.
- Definirati metod **boolean pronadji(T element)** koji vraća odgovarajuću logičku vrednost u zavisnosti od toga da li element postoji u nekom čvoru stabla ili ne.
- Definirati metod **void ispiSortirano()** koji sortirano ispisuje elemente čvorova stabla.
- Definirati klasu **Student** koja ima polja **String ime**, **String prezime** i **String alasNalog**. Studenti se porede leksikografski po alas nalogu.
- Definirati klasu **Test** koja testira implementirane metode za rad sa binarnim pretraživačkim stablom čiji su elementi u čvorovima objekti tipa **Student**.