

Objektno-orientisano programiranje, Jun 1

Matematički fakultet

Školska godina 2021/2022

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, paket takođe nazvati tako.

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

Vreme za rad: **3 sata**

Inicijali: IR: 2I1A | SS: 2I1B | VK: 2I2A, 2I2B

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 25 poena**.

1. Napraviti klasu `Broj` koja se karakteriše poljima `kodDrzave` (`String`), `pozivniBroj` (`String`), `brojTelefona` (`String`) i `fiksniTelefon` (`boolean`). Implementirati:

- konstruktor koji uzima jedan `String` formata `+XXX XX XX.X` i na osnovu njega inicijalizuje sva polja novog objekta (da li je broj fiksnog telefona ili ne određuje se na osnovu prve cifre pozivnog broja - kod mobilnih telefona to je uvek cifra '6')
- sve potrebne `get` metode
- metod `boolean istaDrzava(Broj b)` koji proverava da li je dati broj iz iste države kao i broj `b`
- metod `toString` koji vraća nisku kao u test primeru (slika 2)

Definisati kao kriterijum poredjenja objekata tipa `Broj` leksikografsko poredjenje njihovih stringovskih reprezentacija.

2. Napraviti apstraktnu klasu `Usluga` koja se karakteriše poljima `vreme` (`String`, vreme realizovanja/iniciranja telefonske usluge), `broj0d` i `brojKa` (`Broj`, broj koji je inicirao uslugu i broj kome je usluga upućena). Implementirati:

- konstruktor koji prima vrednosti za sva polja i **vrši duboko kopiranje mutabilnih polja**
- sve potrebne `get` metode
- metod `toString` koji vraća nisku formata `vreme broj0d -> brojKa`

Klasa sadrži i apstraktne metode `double cena()` koji računa cenu realizovane telefonske usluge. Kao kriterijum poredjenja objekata tipa `Usluga` definisati njihovo poredjenje prema vremenu realizovanja/iniciranja telefonske usluge.

3. Napraviti klasu `Poziv` koja nasleđuje klasu `Usluga`. Klasa se dodatno karakteriše poljem `trajanjeS` (`int`, trajanje poziva u sekundama). Implementirati konstruktor koji uzima vrednosti za sva polja, metod `toString` koji vraća nisku kao u test primeru (slika 4), kao i nasleđeni apstraktni metod. Cena poziva se formira na osnovu cene uspostavljanja veze i cene svakog započetog minuta razgovora:

- uspostavljanje veze između fiksnih telefona u domaćem saobraćaju je besplatno, a svaki započeti minut naplaćuje se 8 dinara
- uspostavljanje veze između fiksnih i mobilnih telefona u domaćem saobraćaju je 5 dinara, dok se svaki započeti minut naplaćuje 10 dinara
- uspostavljanje veze između mobilnih telefona u domaćem saobraćaju je 0 dinara, dok se svaki započeti minut naplaćuje 12 dinara
- uspostavljanje veze između fiksnih i/ili mobilnih telefona u inostranom saobraćaju je 30 dinara, dok se svaki započeti minut naplaćuje 50 dinara

Trajanje poziva bilo koje vrste koji traje 0 sekundi označava da veza nije uspostavljena, te je cena takvog poziva 0 dinara.

4. Napraviti klasu `Poruka` koja nasleđuje klasu `Usluga`. Klasa se dodatno karakteriše poljem `tekst` (`String`, sadržaj poruke) i `poslata` (`boolean`, identifikator da li je poruka uspešno poslata). Implementirati konstruktor koji uzima vrednosti za sva polja, metod `toString` koji vraća nisku kao u test primeru (slika 4), kao i nasleđeni apstraktni metod. Cena poslate poruke u domaćem saobraćaju je 3 dinara, a ka inostranom 20 dinara. Za poruku koja nije uspešno poslata cena je 0 dinara.

5. Napraviti šablonski interfejs `Kopirajuci<T>` koji se karakteriše apstraktnim metodama:

- `boolean mozeDaSeKopira()` — vraća indikator da li instance klase koja implementira ovaj interfejs mogu da se kopiraju ili ne
- `T kopiraj(T)` — duboko kopira argument i vraća kopiju

Implementirati `Kopirajuci<T>` za klasu `Usluga`. Duboko kopiranje iz konstruktora prebaciti u implementaciju metoda `kopiraj` i pozvati metod `kopiraj` iz konstruktora. Klasa `Usluga` implementira metod `mozeDaSeKopira` tako što vraća `true`. Implementirati `Kopirajuci<T>` za klasu `Broj`. `Broj` implementira metod `mozeDaSeKopira` tako što vraća `false`, a metod `kopiraj` vraća `null`.

6. Napraviti klasu `TelefonskiListing` koja nasleđuje klasu `Application` biblioteke `javafx` i izgleda kao na slikama. Klasa sadrži polje `evidentiraneUsluge` (`Map<Broj, List<Usluga>>`) koje omogućava efikasnu pretragu liste usluga za dati broj telefona.

Iz datoteke `evidentirane_usluge.txt` (primer je dat na kraju teksta) učitavaju se informacije o telefonskim uslugama i smeštaju u liste usluga odgovarajućih brojeva telefona. Pojedinačne usluge treba da budu evidentirane u listama i broja koji je inicirao uslugu i broja kome je usluga upućena. Tip usluge može se prepoznati na osnovu broja i tipa elemenata koji su zadati u jednom redu datoteke. Održavati mapiranje `evidentiraneUsluge` tako da čuva informacije o svim telefonskim uslugama za svaki od brojeva telefona.

Prilikom kreiranja grafičkog korisničkog interfejsa kroz komponentu `ChoiceBox` treba ponuditi sve brojeve telefona koji su na raspolaganju za generisanje listinga (na osnovu evidencije usluga pročitanih iz datoteke). Za dodavanje opcija u `ChoiceBox` komponentu možete koristiti metode `getItems().add(String)`.

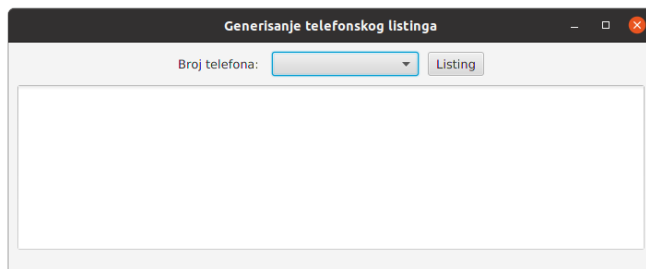
Klikom na dugme `Listing` čitaju se selektovan broj telefona putem `ChoiceBox` komponente (metod `getValue()`). U `TextArea` element ispisuje se listing usluga od i ka selektovanom broju telefona u sortiranom redosledu. Na kraju se ispisuje i ukupna cena telefonskih usluga pruženih datom broju. Dolazni pozivi i poruke se ne naplaćuju.

U datoteci `evidentirane_usluge.txt` svaka linija sadrži informacije o jednoj telefonskoj usluci u sledećem formatu oblika:

- za pozive - vreme, brojOd, brojKa, trajanjeS
- za poruke - vreme, brojOd, brojKa, identifikatorPoslato, tekstPoruke

Pretpostaviti da će sadržaj datoteke biti ispravan. Primer sadržaja datoteke:

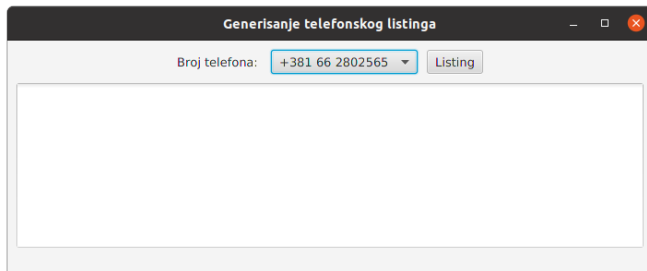
```
2022-01-12 13:05:47, +381 66 2802565, +381 11 270231, 176
2021-12-28 22:13:01, +381 64 1234567, +381 66 2802565, 0, "Pozovi me, nemam kredita."
2022-02-01 10:00:52, +381 11 270231, +381 26 317056, 542
2022-01-12 16:11:09, +381 66 2802565, +387 66 9870012, 1, "Srecan rođendan, Majo! :*"
2022-01-23 21:59:31, +381 26 317056, +387 66 9870012, 324
2021-12-31 00:10:47, +381 66 2802565, +381 64 1234567, 82
2022-02-02 18:51:03, +387 66 9870012, +382 30 890235, 188
2022-02-01 17:34:21, +381 64 1234567, +387 66 9870012, 1, "Krenuo sam kuci. Kad ti stizes?"
```



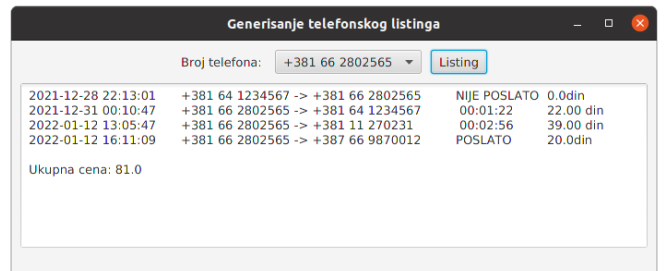
Slika 1: Klik na Početni izgled aplikacije



Slika 2: Opcije u okviru ChoiceBox-a



Slika 3: Izbor broja telefona



Slika 4: Generisanje listinga