*KOZMETIČARKE*

*-Grupa 3-*

*Članovi tima:*

*Anisa Mahmutović*

*Merima Murtić*

*Lejla Mujić*

**Osnovne klase:**

1. Osoba
2. Klijent
3. Uposlenik
4. Administrator
5. Tretman
6. Rezervacija
7. Kategorija
8. Obavijest
9. Poruka

**Kontejnerske klase:**

1. KozmetickiSalon

**Interfejsi:**

1. Prijava
2. Registracija
3. PosaljiObavijest
4. PosaljiPoruku
5. HitanSlucaj

**Klasa Osoba**

* apstraktna klasa (iz koje izvodimo klase Klijent, Administrator i Uposlenik)
* sadrži osnovne podatke o korisniku aplikacije
* atributi
  + *ime (string)*
  + *prezime (string)*
  + *korisnickoIme(string)*
  + *lozinka(string)*
  + *e-mail/telefon (string)*
* Konstruktori
* Metode
* *Geteri*
* *seteri*

**Klasa Administrator**

* izvedena je iz klase Osoba
* Konstruktor

**Klasa Uposlenik**

* izvedena iz klase Osoba
* interakcija uposlenika i klijenta ostvaruje se putem poruka
* Atributi
* *poruke(List<Poruka>)*
* Metode
* *Geteri*
* *Seteri*
* *void dodajPoruku(Poruka poruka)*
* Konstruktori

**Klasa Klijent**

* kako korisnik aplikacije može biti i registrovani i neregistrovani klijent, potreban je atribut koji će nam davati informacije o tome. Ukoliko je klijent registovan, atribut “registrovan” je true, u suprotnom je false (defaultna vrijednost)
* interakcija uposlenika i korisnika “unutar” same aplikacije, odvija se putem poruka
* atribut obavijesti davit će informacije o popustima, hitnom slucaju
* Atributi
* *registrovan(bool)*
* *poruke (List<Poruka>)*
* *obavijesti (List<Obavijest>)*
* Konstruktor
* Metode
* *void dodajPoruku(Poruka poruka)*
* *void dodajObavijest(Obavijest obavijest)*
* *geteri*
* *seteri*

**Klasa Tretman**

* sadrži podatke o tretmanu, njegovom nazivu, cijeni, trajanju
* jedan od atributa je i atribut “opis” koji predstavlja opis samog tretmana, tj. njegove značajne karateristike
* kako Klijent ima mogućnost uvida u izvršene i zakazane tretmane, potreban je jedan atribut koji će davati informacije o tome. Defultna vrijednost atributa “izvrsen” je false.
* Atributi
* *naziv(string)*
* *cijena(double)*
* *opis(string)*
* *kategorija(Kategorija)*
* *trajanje (int)*
* *izvrsen(bool)*
* Konstruktori
* Metode
* *Geteri*
* *seteri*

**Klasa Kategorija**

* klasa koja je namijenjena da da opis/naziv kategorije tretmana
* daje nam informacije kojoj skupini tretmana pripada tretman koji se dodaje (tretman lica, kože, kose itd.)
* Atributi
* *naziv(string)*
* Konstruktori
* Metode
* *geteri*
* *seteri*

**Klasa Rezervacija**

* sadrži osnovne podatke rezervacije (koji tretman se vrši, klijent koji je rezervisao tretman, vrijeme zakazanog tretmana i uposlenik koji će vršiti dati tretman)
* Atributi
* *tretman (Tretman)*
* *datumVrijeme (Local Date)* 
  + *klijent (Klijent)*
  + *uposlenik (Uposlenik)*
* Konstruktori
* Metode
* geteri
* seteri

**Klasa Obavijest**

* obavijesti mogu slati samo uposlenici klijentima
* mogu se slati obavijesti o ostvarenim popustu, hitnom slučaju
* Atributi
* *posiljalac(String)*
* *primalac(String)*
* *obavijest(String)*
* *datumSlanja(LocalDate)*
* *tipObavijesti(TipObavijesti)*
* Metode
* *geteri*
* *seteri*
* Konstruktori

**Enum TipObavijesti**

* daje nam tip/kategoriju obavijesti koja može biti poslana
* HitanSlucaj
* Popust

**Klasa Poruka**

* klasa za interakciju izmedju klijenta i uposlenika
* prilikom slanja poruke, atributi posiljalac i primalac u sebi sadrže korisnička imena uposlenika i/ili klijenta
* Atributi
* *posiljalac (String)*
* *primalac (String)*
* *poruka (String)*
* *datumSlanja (LocalDate)*
* Metode
* *geteri*
* *seteri*
* Konstruktori

**Klasa KozmetičkiSalon:**

* klasa koja predstavlja“stub” aplikacije
* u sebi sadrži sve informacije o registrovanim klijentima, uposlenicima, tretmanima, kategorijama tretmana
* Implementira interfejse Prijava, Registracija, HitanSLucaj, PošaljiObavijest, PošaljiPoruku
* Atribute
* *rezervacije(List<Rezervacija>)*
* *tretmani(List<Tretman>)*
* *klijenti(List<Klijent>)*
* *uposlenici (List<Uposlenik>)*
* *kategorije (List<Kategorija>)*
* Konstruktori
* Metode
  + *Geteri*
  + *Seteri*
  + *void dodajNovuKategoriju (Kategorija kategorija)*
  + *void obrisiKategoriju (Kategorija kategorija)*
  + *void dodajNoviTretman (Tretman tretman)*
  + *void obrisiTretman (Tretman tretman)*
  + *void urediTretman (Tretman tretman)*
  + *void obrisiKlijenta (Klijent klijent)*
  + *void dodajNovogUposlenika (Uposlenik uposlenik)*
  + *void obrisiUposlenika (Uposlenik uposlenik)*
  + *void urediUposlenika (Uposlenik uposlenik)*
  + *void promijeniRezervaciju (Rezervacija rezervacija)*

**Interfejs** **Registracija**

* interfejs kojim se dodaju novi klijenti
* Metode
* *void registracija(String ime, String prezime, String korisnickoIme, String lozinka, String e-mail)*

**Interfejs Prijava**

* Metode
* *void prijava (String korisnickoIme, String lozinka)*

**Interfejs PošaljiPoruku**

* Metode
* *void posaljiPoruku (String posiljalac, String primalac, Poruka poruka)*

**Interfejs PošaljiObavijest**

* Metode
* void posaljiObavijest (String posiljalac, String primalac, Obavijest obavijest)

**Interfejs HitanSlucaj**

* unutar metode aktivirajHitanSlucaj pozivat će se metoda posaljiObavijest (String posiljalac, String primalac, Obavijest obavijest)
* Metode
  + *void aktivirajHitanSlucaj (Klijent klijent, Uposlenik uposlenik)*

**Veze:**

* Iz klase *Osoba* su izvedene klase *Klijent*, *Administator* i *Uposlenik* i to je **generalizacija**.
* Klasa *KozmetičkiSalon* u sebi sadrži kolekcije *Tretman*, *Rezervacija, Uposlenik* , *Klijent* i *Kategorija* i to je **kompozicija** jer ove klase ne mogu postojati bez klase KozmetickiSalon
* Klasa *KozmetičkiSalon* implementira interfejse *Prijava*, *Registracija*, *PosaljiPoruku*, *PosaljiObavijest* i *HitanSlucaj* i to je **realizacija**
* Klasa *Tretman* sadrži atribut tipa klase *Kategorija* i ako se klasa *Tretman* obriše, *Kategorija* više nema smisla da postoji pa je to **kompozicija**
* Klasa *Rezervacija* ima atribute tipa *Tretman*, *Klijent* i *Uposlenik*  i ako se klasa *Rezervacija* obriše, ove klase mogu nezavisno postojati i to je **agregacija**
* Kalsa *Uposlenik* ima atribut tipa *Poruka* i ako klasa *Uposlenik* prestane postojati, klasa *Poruka* nema smisla više, pa je to **kompozicija**
* Klasa *Klijent* ima atribut tipa *Obavijest* i *Poruka* i to je **kompozicija** jer klase mogu postojati neovisno jedna od druge