

ANALIZA SISTEMA

1. KORISNIK SISTEMA

-apstraktna klasa

Atributi:

- ime (string)
- prezime (string)
- datumRodjenja (DateTime)
- password (string)

Metode:

- Konstruktor sa svim atributima
- Getteri
- Setteri

2. KORISNIK SA NALOGOM

-izvedena klasa iz klase korisnik sistema

Atributi:

- email (string)
- bankovniRacun (BankovniRacun)
- rezervisaneKarte (list<Karta>)
- imaPravoNaPopust (bool)

Metode:

- Konstruktor sa svim atributima, osim atributa imaPravoNaPopust
- Getteri
- Setteri
- rezervisiKartu (Karta)
- otkaziRezervaciju (Karta)

3. ADMINISTRATOR

-izvedena iz klase korisnik sistema

Nema dodatnih atributa

Metode:

- editujFilm (Film)
- obrisiFilm (Film)
- dodajFilm (Film)
- dodajProjekciju (Projekcija)
- obrisiProjekciju (Projekcija)

4. FILM

Atributi:

- naziv (String)
- zanr (list<Zanr>)
- trajanje (int)
- sinopsis (string)
- glumci (list<String>)
- reziser (String)
- trailer (String)
- slika (String)

Metode:

- Konstruktor sa svim atributima
- Getteri
- Setteri

5. KARTA

Atributi:

- IdKarte (int)
- Projekcija (Projekcija)
- cijenaKarte (double)
- sala (Sala)
- sjediste (String)

Metode:

- uracunajPopust (bool)
- Konstruktor sa svim atributima
- Getteri
- Setteri

6. SALA

Atributi:

- idSale (String)
- slobodnaSjedista (list<String>)
- brojSjedista (int)
- slikaSale (String)

Metode:

- prazan konstruktor
- konstruktor sa idSale, slikaSale
- Getteri i Setteri
- zauzmiSjediste (String)

7. PROJEKCIJA

Atributi:

- Film (Film)
- DatumIVrijemePrikazivanja (DateTime)
- sala (Sala)

Metode:

- Konstruktor sa svim atributima
- Getteri
- Setteri

8. ZANR

-pobrojani tip zanrova (enum klasa)

9. BANKOVNIRACUN

Atributi:

- BrojRacuna (long)
- datumIsteka (DateTime)
- CSC (int)
- stanjeRacuna (double)

Metode:

- Konstruktor sa svim atributima
- Getteri
- Setteri

10.CINEMAGIC

Atributi:

- Filmovi (list<Film>)
- Sale (list<Sala>)
- Projekcije (list<Projekcija>)
- Korisnici (list<KorisnikSaNalogom>)
- Administratori (list<Administrator>)

Metode:

- Default konstruktor
- Getteri i setteri
- dodajFilm (Film)
- editujFilm (Film, Film) – stari i novi film
- obrisiFilm (Film)
- dodajAdministradora(Administrator)
- obrisiAdministradora(Administrator)
- dodajProjekciju (Projekcija)
- obrisiProjekciju (Projekcija)
- dodajKorisnika (KorisnikSaNalogom)
- obrisiNalog (KorisnikSaNalogom)
- urediNalog (KorisnikSaNalogom, KorisnikSaNalogom) – stari i novi nalog
- filtrirajFilmove – filtrira filmove po odredjenim parametrima
- dajProjekcijePoDanu (String)
- rezervisiKartu (Karta)
- otkaziRezervaciju (Karta)
- zauzmiSjedisteUSali (Sala, String)

11.SISTEM ZA KARTIČNO POSLOVANJE

Atributi:

- BankovniRacuni (list<BankovniRacun>)

Metode:

- validirajBankovniRacun (BankovniRacun)
- skiniNovacSaRacuna (double)
- uplatiNovacNaRacun (double)

12.BAZA

Baza podataka će čuvati sve podatke o filmovima, projekcijama, korisnicima i bankovnim računima

Veze između klasa:

- Imamo vezu između klase CineMagic i KorisnikSaNalogom. Veza između ove dvije klase je kompozicija, jer korisnik sa nalogom ne može postojati bez CineMagica, a isto vrijedi i za Administratora, Projekciju i Korisnika sistema.
- Veza između klase KorisnikSistema i klasa KorisnikSaNalogom i Administrator je generalizacija, jer se radi o nasljeđivanu klase KorisnikSistema.
- Veza između klasa Film i CineMagic je agregacija, jer Film može postojati i bez CineMagica (CineMagic sadrži listu filmova kao atribut).
- Veza između klasa Karta i KorisnikSaNalogom je agregacija, jer Karta može postojati i bez klase KorisnikSaNalogom (KorisnikSaNalogom sadrži listu karata kao atribut).
- Veza između klasa Film i Projekcija je ovisnost, jer izmjena Filma dovodi do izmjena u Projekciji koja je vezana za taj Film.

SOLID principi:

- Single responsibility principe je ispoštovan jer svaka klasa radi ono za šta je i namijenjena. Jedina klasa koja bi potencijalno mogla kršiti ovaj princip je CineMagic.
- Open/Closed Principle je narušen jer klase imaju kao attribute druge klase.
- Liskov Substitution Principle je ispoštovan jer gdje god koristimo klasu Korisnik sistema možemo koristiti i klasu Administrator ili Korisnik sa nalogom, koje su izvedene iz bazne klase Korisnik sistema.
- Interface Segregation Principle – interfejsi nisu korišteni
- Dependency Inversion Principle je ispoštovan jer je bazna klasa ujedno i apstraktna klasa