

## StudentskiDom

- ✓ Ovo je kontejnerska klasa za ovaj projekat
- ✓ Cilj ove klase je da na jednom mjestu čuvamo sve podatke u cilju jednostavnijeg i lakšeg povezivanja našeg sistema u jednu cjelinu
- ✓ Atributi:
  - ❖ studenti (List<Student>)
  - ❖ uprava (Uprava)
  - ❖ restoran (Restoran)
  - ❖ zahtjevi (List<Zahtjev>)
  - ❖ paviljoni (List<Paviljon>)
- ✓ Metode ove klase su:
  - ❖ geteri i seteri
  - ❖ upisiStudenta - ova metoda se poziva prilikom odobravanja zahtjeva za upis studenta. U ovoj metodi dodajemo studenta u bazu podataka, te vršimo ažuriranje smještajnih kapaciteta.
  - ❖ brisiStudenta - ova metoda se poziva prilikom ispisivanja studenta iz studentskog doma
  - ❖ dodajZahtjev - ova metoda dodaje zahtjev proslijeđen kao parametar u listu zahtjeva
  - ❖ obradiZahtjev - ova metoda obrađuje zahtjev koji je proslijeđen kao parametar i briše ga iz liste zahtjeva
  - ❖ daLiImaMjesta - ova metoda se poziva da bismo provjerili da li ima mjesta za još studenata u studentskom domu
  - ❖ nadjiStudentaPoIDu - pronalazi studenta iz liste studenata koji ima odgovarajući ID
- ✓ Ova klasa sadrži konstruktor bez parametara koji prilikom instanciranja ove klase iz baze podataka kupi potrebne podatke
- ✓ Ovo vjerovatno nisu sve metode koje sadrži klasa StudentskiDom, te zbog toga postoji mogućnost proširivanja ove klase

## Korisnik

- ✓ Apstraktna klasa
- ✓ Svi akteri su izvedeni iz ove klase
- ✓ Atributi:
  - ❖ username (string)
  - ❖ password (string)
- ✓ Od metoda ova klasa posjeduje samo getere i setere za svoje metode
- ✓ Ova klasa ima konstruktor sa dva parametra, kao i konstruktor bez parametara koji je namijenjen za gosta koji još nema username i password

## Student

- ✓ Izvedena iz klase Korisnik (generalizacija)
- ✓ Pored atributa iz klase Korisnik, klasa Student sadrži i sljedeće attribute:
  - ❖ id (int)
  - ❖ licniPodaci (LicniPodaci)
  - ❖ prebivaliste (PrebivalisteInfo)
  - ❖ skolovanje (SkolovanjeInfo)
  - ❖ soba (Soba)
  - ❖ brojRucaka(int)
  - ❖ brojVecera(int)
- ✓ Ova klasa ima sljedeće metode:
  - ❖ getere i setera za svaki atribut
  - ❖ azurirajLicnePodatke - ova metoda služi za promjenu ličnih podataka o studentu
  - ❖ azurirajPrebivaliste - ova metoda služi za promjenu informacija o prebivalištu studenta

- ✓ Ova klasa ima defaultni konstruktor sa svim parametrima koji odgovaraju atributima

### **PrebivalisteInfo**

- ✓ Ova klasa nam služi za čuvanje informacija o prebivalištu studenata
- ✓ Atributi ove klase su:
  - ❖ adresa (String)
  - ❖ kanton (String)
  - ❖ općina (String)
- ✓ Ova klasa od metoda sadrži samo getere i setere za attribute ove klase
- ✓ Ova klasa ima defaultni konstruktor sa svim parametrima koji odgovaraju atributima

### **SkolovanjeInfo**

- ✓ Ova klasa nam služi za čuvanje informacija o fakultetu studenata
- ✓ Atributi ove klase su:
  - ❖ fakultet (String)
  - ❖ brojIndeksa (int)
  - ❖ ciklusStudija (int)
  - ❖ godinaStudija (int)
- ✓ Ova klasa od metoda sadrži samo getere i setere za attribute ove klase
- ✓ Ova klasa ima defaultni konstruktor sa svim parametrima koji odgovaraju atributima

**Pol (ENUM) {MUSKO, ZENSKO}**

## Uprava

- ✓ Ova klasa je izvedena iz klase Korisnik (generalizacija)
- ✓ Atributi ove klase:
  - ❖ blagajna (Blagajna)
- ✓ Metode ove klase su getter i setter za blagajnu
- ✓ Ova klasa ima konstruktor bez parametara koji kreira instancu klase Blagajna

## Restoran

- ✓ Ova klasa je izvedena iz klase Korisnik (generalizacija)
- ✓ Ova klasa implementira interfejs PregledStanjaBonova, kao i interfejs AzurirajStanjeBonova
- ✓ Atributi ove klase su:
  - ❖ dnevniMeni (DnevniMeni)
  - ❖ idTrenutnogStudenta (int)
- ✓ Metode ove klase su:
  - ❖ azurirajDnevniMeni - ova metoda se poziva u slučaju da želimo vršiti ažuriranje dnevnog menija
  - ❖ dodajRucak
  - ❖ dodajVeceru
  - ❖ izbaciRucak
  - ❖ izbaciVeceru
- ✓ Ova klasa ima konstruktor bez parametara u kojem se samo kreira instanca klase DnevniMeni

## Zahtjev

- ✓ Ovo je apstraktna klasa
- ✓ Ova klasa nam služi da bismo iz nje izveli ostale zahtjeve
- ✓ Sadrži attribute:
  - ❖ id (int)
  - ❖ podnosilacZahtjeva (Korisnik)
  - ❖ datum (DateTime)
  - ❖ odobren (Bool)
- ✓ Metode ove klase:
  - ❖ posaljiZahtjev - zahtjev se dodaje u listu zahtjeva i čeka se na njegovo odobrenje
  - ❖ geteri i seteri
- ✓ Ova klasa sadrži konstruktor koji prima prva tri atribura, a odobren se automatski stavlja na false

## ZahtjevZaPremjestanje

- ✓ Ova klasa je izvedena iz klase Zahtjev (generalizacija)
- ✓ Atributi ove klase su:
  - ❖ trenutniPaviljon(Paviljon)
  - ❖ trenutnaSoba (Soba)
  - ❖ noviPaviljon (Paviljon)
  - ❖ novaSoba (Soba)
  - ❖ razlogPremjestanja (String)
- ✓ Ova klasa nema dodatnih metoda za razliku od bazne klase osim getera i setera

- ✓ Konstruktor ove klase ima za parametre pored svojih atributa, i attribute bazne klase. On vrši pozivanje konstruktora bazne klase.

## **ZahtjevZaUpis**

- ✓ Ova klasa je izvedena iz klase Zahtjev (generalizacija)
- ✓ Atributi ove klase su:
  - ❖ licniPodaci (LicniPodaci)
  - ❖ prebivaliste (PrebivalisteInfo)
  - ❖ skolovanje (SkolovanjeInfo)
- ✓ Ova klasa nema dodatnih metoda za razliku od bazne klase osim getera i setera
- ✓ Konstruktor ove klase ima za parametre pored svojih atributa, i attribute bazne klase. On vrši pozivanje konstruktora bazne klase.

## **ZahtjevZaCimeraj**

- ✓ Ova klasa je izvedena iz klase Zahtjev (generalizacija)
- ✓ Atributi ove klase su:
  - ❖ paviljon (Paviljon)
  - ❖ soba (Soba)
  - ❖ prviCimer (int)
  - ❖ drugiCimer (int)
  - ❖ dodatneNapomene (String)
- ✓ Ova klasa nema dodatnih metoda za razliku od bazne klase osim getera i setera
- ✓ Konstruktor ove klase ima za parametre pored svojih atributa, i attribute bazne klase. On vrši pozivanje konstruktora bazne klase.

## **ZahtjevZaNabavkuNamirnica**

- ✓ Ova klasa je izvedena iz klase Zahtjev (generalizacija)
- ✓ Atributi ove klase su:
  - ❖ listaNamirnica (List<String>)
  - ❖ narudzba (List<StavkaNarudzbe>)
- ✓ Ova klasa nema dodatnih metoda za razliku od bazne klase osim getera i setera
- ✓ Konstruktor ove klase ima za parametre pored svojih atributa, i attribute bazne klase. On vrši pozivanje konstruktora bazne klase.

## **StavkaNarudzbe**

- ✓ Ova klasa nam služi kao klasa koja nam povezuje namirnicu koju treba naručiti sa količinom koju treba naručiti
- ✓ Atributi:
  - ❖ namirnica (String)
  - ❖ kolicina (double)
- ✓ Metode ove klase su geteri i seteri za attribute ove klase
- ✓ Ova klasa ima defaultni konstruktor sa dva parametra

## Blagajna

- ✓ Ova klasa implementira interfejs AzurirajStanjeBonova
- ✓ Ova klasa se koristi za regulisanje uplata smještaja studenata studentskog doma i regulisanje stanja bonova
- ✓ Atributi ove klase su:
  - ❖ stanjeBudgeta (double)
  - ❖ trenutniStudent (Student) - ovaj atribut predstavlja studenta koji trenutno vrši uplatu
- ✓ Metode ove klase su:
  - ❖ provjeriID - provjerava ID studenta i na osnovu njega pronalazi studenta iz liste studenata
  - ❖ uplatiDomZaOdabraniMjesec - ova metoda ažurira stanje budžeta i poziva metode za ažuriranje stanja bonova studenta
  - ❖ ucitajStanjeBudzeta - čita stanje budžeta iz baze podataka
- ✓ Ova klasa ima konstruktor bez parametara unutar kojeg se poziva metoda ucitajStanjeBudzeta

## Paviljon

- ✓ Ova klasa nam služi kao kontejnerska klasa koja na jednom mjestu čuva sobe jednog paviljona
- ✓ Atributi ove klase su:
  - ❖ naziv (String)
  - ❖ sobe (List<Soba>)
  - ❖ kapacitet
  - ❖ brojStudenata
- ✓ Ova klasa od metoda ima samo getere i setere za attribute



- ❖ daLilmaMjesta - ova metoda se poziva da bi se utvrdilo da li ima mjesta u datom paviljonu
- ❖ brojSlobodnihMjesta - ova metoda nam služi da bismo odredili broj slobodnih mjesta u datom paviljonu
- ✓ Klasa posjeduje defaultni konstruktor sa četiri parametra

## **Soba**

- ✓ Atributi ove klase su:
  - ❖ brojSobe (int)
  - ❖ studenti (List<Student>)
  - ❖ kapacitet
- ✓ Metode ove klase:
  - ❖ geteri i seteri za attribute ove klase
  - ❖ dodajStudentaUSobu
  - ❖ izbaciStudentaIzSobe
  - ❖ daLilmaMjesta - ova metoda se poziva da bi se utvrdilo da li je određena soba popunjena
- ✓ Ova klasa posjeduje defaultni konstruktor sa tri parametra, kao i konstruktor sa dva parametra u slučaju da je soba prazna

## **DnevniMeni**

- ✓ Atributi ove klase su:
  - ❖ listaRucaka (List<String>)
  - ❖ listaVecera (List<String>)
- ✓ Metode ove klase su:
  - ❖ geteri i seteri

- ❖ dodajRucak
- ❖ dodajVeceru
- ❖ izbaciRucak
- ❖ izbaciVeceru
- ✓ Ova klasa pored konstruktora sa dva parametra posjeduje i konstruktor bez parametara koji služi za kreiranje praznog dnevnog menija

## **LicniPodaci**

- ✓ Cilj ove klase je čuvanje ličnih podataka na jednom mjestu, odnosno grupisanje podataka
- ✓ Atributi ove klase su:
  - ❖ prezime (String)
  - ❖ ime (String)
  - ❖ mjesto rođenja (String)
  - ❖ pol (Pol)
  - ❖ email (String)
  - ❖ JMBG (long)
  - ❖ datum rođenja (DateTime)
  - ❖ mobitel (int)
  - ❖ slika (String)
- ✓ Metode ove klase su seteri i geteri za atribute
- ✓ Ova klasa sadrži defaultni konstruktor sa svim parametrima koji predstavljaju odgovarajuće attribute

## **Interfejs: PregledStanjeBonova**

- ✓ Pomoću ovog interfejsa imamo uvid u stanje bonova određenog studenta

- ✓ Metode ove klase su:
  - ❖ dajBrojRucakaZaStudenta
  - ❖ dajBrojVeceraZaStudenta

## Interfejs:AzurirajStanjeBonova

- ✓ Pomoću ovog interfejsa vršimo ažuriranje stanja bonova određenog studenta
- ✓ Metode ove klase su:
  - ❖ azurirajStanjeRucaka
  - ❖ azurirajStanjeVecera

## Veze između klasa

- ✓ Klasa StudentskiDom ima veze sa sljedećim klasama:
  - ❖ Student - pošto StudentskiDom sadrži listu studenata kao atribut i pošto student može postojati bez studentskog doma, zaključujemo da je veza između ove dvije klase *agregacija*
  - ❖ Uprava - klasa StudentskiDom sadrži instancu klase Uprava kao atribut i kao što znamo da uprava ne može postojati bez studentskog doma, tako da je veza između ove dvije klase *kompozicija*
  - ❖ Restoran - klasa StudentskiDom sadrži instancu klase Restoran kao atribut i pošto je naša zamisao bila da restoran radi isključivo sa studentima, time smatramo da nema smisla njegovo postojanje bez studentskog doma. Iz tog razloga zaključujemo da je veza između ove dvije klase *kompozicija*

- ❖ Zahtjev - klasa StudentskiDom sadrži listu zahtjeva kao atribut i pošto zahtjevi za studentski dom nemaju smisla bez studentskog doma, zaključujemo da je veza između ove dvije klase *kompozicija*
- ❖ Paviljon - klasa StudentskiDom sadrži listu paviljona kao atribut i pošto postojanje paviljona nema smisla bez postojanja studentskog doma, zaključujemo da je veza između ove dvije klase *kompozicija*
- ✓ Klasa Korisnik ima veze sa sljedećim klasama:
  - ❖ Student
  - ❖ Uprava
  - ❖ Restoran
  - ❖ Sve ove tri klase nasljeđuju klasu Korisnik tako da je veza između ovih klasa sa klasom Korisnik veza *generalizacije*
- ✓ Klasa Student ima veze sa sljedećim klasama:
  - ❖ LicniPodaci - klasa Student sadrži kao atribut klasu LicniPodaci i klasa LicniPodaci nastavlja postojati i bez klase Student, tako da je veza između ove dvije klase *agregacija*
  - ❖ PrebivalisteInfo - klasa Student sadrži kao atribut klasu PrebivalisteInfo, te pošto podaci o prebivalištu imaju smisla i bez klase Student, tako da je veza između ove dvije klase *agregacija*
  - ❖ SkolovanjeInfo - klasa Student sadrži kao atribut klasu SkolovanjeInfo, te pošto se student školuje, nepostojanje klase Student, znači nepostojanje klase SkolovanjeInfo, te zbog toga je veza između ove dvije klase *kompozicija*
  - ❖ Soba - klasa Student sadrži kao atribut klasu Soba, i pošto klasa Soba može postojati i kada ne postoji klasa Student, zaključujemo da je veza između ove dvije klase *agregacija*
- ✓ Klasa Uprava ima vezu sa klasom Blagajna, jer ima instancu te klase kao svoj atribut, te pošto blagajna nema smisla bez uprave, veza između ove dvije klase je *kompozicija*
- ✓ Klasa Restoran ima veze sa sljedećim klasama:
  - ❖ DnevniMeni - jer posjeduje atribut tipa DnevniMeni, i pošto smo mi zamislili da DnevniMeni nema smisla bez klase Restoran, veza između ove dvije klase je *kompozicija*

- ❖ Student - pošto klasa Restoran ažurira broj bonova određenog studenta preko interfejsa, tako da je veza između ove dvije klase obična asocijacija
- ❖ Ova klasa realizuje dva interfejsa, PregledStanjaBonova i AzuriranjeStanjaBonova, tako da je veza između ovih klasa *realizacija*
- ✓ Klasa Zahtjev ima veze sa sljedećim klasama:
  - ❖ Korisnik - jer ima atribut tipa Korisnik. Veza je *agregacija* jer korisnik nastavlja postojati, iako ne postoji Zahtjev
  - ❖ ZahtjevZaPremjestanje
  - ❖ ZahtjevZaUpis
  - ❖ ZahtjevZaCimeraj
  - ❖ ZahtjevZaNabavkuNamirnica
  - ❖ Sve ove Zahtjev klase nasljeđuju klasu Zahtjev, tako da je veza između njih *generalizacija*
- ✓ Ove klase Zahtjeva imaju sa svojim atributima vezu *agregacije*
- ✓ Klasa Blagajna ima veze sa sljedećim klasama:
  - ❖ AzurirajStanjeBonova - Blagajna implementira ovaj interfejs, tako da je veza između ove dvije klase *realizacija*
  - ❖ Student - *agregacija*
- ✓ Klasa Paviljon ima vezu sa klasom Soba, tip veze je *agregacija*
- ✓ Klasa Soba ima vezu *agregacija* sa klasom Student
- ✓ Klasa DnevniMeni nema dodatnih veza sa drugim klasama
- ✓ Klasa LicniPodaci nema dodatnih veza sa drugim klasama
- ✓ Veze interfejsa sa drugim klasama su navedeni prije, tako da nema dodatnih veza