



Analiza sistema

Objektno orijentisana analiza i dizajn

Projekat: Aplikacija za studentski dom

Grupa: SeKEmin

Studenti: Kemal Lazović

Semir Suljević

Emin Šiljak

Mali uvod o projektu

Ovaj projekat je zamišljen da pomogne kako studentima stanarima studentskog doma, tako i upraviteljima studentskog doma. Aplikacija je dostupna svima, tj. svi imaju pravo poslati zahtjev za zaduživanje međutim slanje samog zahtjeva nije dovoljno da se dobije mjesto u ovom studentskom domu. Upravitelj je taj koji stoji između studenta (potencijalnog stanara) i njegovog mjesta u studentskom domu. Krenimo onda sa imenovanjem naših potencijalnih klasa.

Osoba

- Apstraktna klasa
- Svi akteri izvedeni iz ove klase
- Atributi:
 - ime (string)
 - prezime (string)
 - grad iz kojeg dolazi (string)
 - datum rođenja (date)
- Od metoda osoba ima samo getere i setere za svoje atribute
- Klasa sadrži jedan konstruktor sa svim parametrima

Student

- Izvedena iz klase Osoba (generalizacija)
- Pored atributa iz klase Osoba sadrži i sljedeće atribute:
 - fakultet (string)
 - godina studija (int)
 - korisničko ime (string)
 - password (string)
 - količina bonova (int) - nakon svakog zaduživanja se postavlja na 50, na kraju mjeseca ukoliko se student nije zadužio za naredni mjesec ovaj iznos automatski spada na 0
 - id (int)
 - cimer id (string)
- Cimer id predstavlja unikatni string radi pojednostavljenja zaduživanja
- Ovaj id dobijemo kao id iz baze + prvo slovo imena + prvo slovo prezimena
- Od metoda klasa pored getera i setera sadrži:
 - posaljiZahtjevZaZaduzenje – u bazu upisuje informacije o studentu kao i to da je isti poslao zahtjev na zaduženje koji čeka na obradu od strane upravitelja

- posaljiZahtjevZaRazduzivanje - slična metoda prethodnoj jedino što student u ovom slučaju mora već biti zadužen
- rezervisiTerminUSobiZaZabavu – metoda koja šalje zahtjev studenta za rezervaciju sobe u terminu koji je proslijeđen kao parametar funkcije
- posaljiZahtjevZaPodizanjeKnjige – metoda koja šalje zahtjev studenta za podizanje knjige koja je proslijeđena kao parametar
- potrosiBonoveZaHranu – metoda koja studentu smanjuje broj bonova za onoliko koliko vrijedi proslijeđena hrana (ukoliko student ima dovoljno bonova I ukoliko taj dan zajedno s tim obrokom nije potrošio više od 5 bonova)
- Što se tiče konstruktora ova klasa pored baznog konstruktora sadrži i konstruktor sa svim parametrima

Knjiga

- Ova klasa služi kao “pomagalo” upravitelju za biblioteku
- Od atributa ova klasa ima:
 - naziv djela (string)
 - žanr (string)
 - ime pisca (string)
 - broj izdanja (int)
 - trenutni vlasnik (Student)
- Što se tiče metoda klasa pored getera i setera sadrži i sljedeće metode:
 - podigniKnjigu – metoda koja u slučaju da knjiga nije podignuta postavlja njenog vlasnika na onog studenta koji je proslijeđen kao parametar, u suprotnom metoda baca izuzetak
 - vratiKnjigu – metoda koja postavlja vlasnika knjige na null ukoliko nije null, a u suprotnom baca izuzetak
- Klasa od konstruktora ima samo konstruktor sa svim parametrima

Zahtjev

- Apstraktna klasa iz koje su naslijeđene sve klase za slanje zahtjeva u bazu
- Od atributa klasa ima samo svoj id koji predstavlja id itema u bazi, kao i referencu na studenta koji šalje zahtjev
- Od konstruktora ova klasa posjeduje samo konstruktor koji prima referencu na studenta

Zahtjev za promjenu stanovanja

- Klasa naslijeđena iz klase Zahtjev (generalizacija)
- Klasa se upotrebljava kod slanja zahtjeva za zaduživanje, kao i kod zahtjeva za razduživanje

- Od atributa klasa sadrži samo one iz bazne klase
- Od metoda klasa sadrži samo getere i setere
- Od konstruktora klasa sadrži samo bazni konstruktor

Zahtjev za podizanje knjige

- Klasa naslijeđena iz klase Zahtjev (generalizacija)
- Klasa se upotrebljava kod slanja zahtjeva za podizanje knjige od strane studenta
- Od atributa pored onih iz bazne klase sadrži i referencu na knjigu koju student želi podići
- Od metoda klasa sadrži samo getere i setere
- Od konstruktora pored baznog konstruktora sadrži i konstruktor koji pored reference na studenta prima i referencu na knjigu

Zahtjev za rezervaciju sobe

- Klasa naslijeđena iz klase Zahtjev (generalizacija)
- Klasa se upotrebljava kod slanja zahtjeva za rezervaciju sobe za zabavu od strane studenta
- Od atributa klasa pored onih iz bazne klase sadrži i datum za koji se rezerviše termin (datum sadrži i vrijeme) kao i double vrijednost koja predstavlja dužinu trajanja ostanka u sobi izraženu u satima (ne može biti više od 3 jer je 3 mjesečno sata u sobi maksimum)
- Od metoda klasa sadrži samo getere i setere za svoje attribute
- Od konstruktora pored baznog posjeduje i konstruktor koji prima datum rezervacije kao i dužinu rezervacije

Studentski dom

- Kontejnerska klasa sistema
- Od atributa sadrži listu studenata korisnika studentskog doma, liste upravitelja (za različite vrste upravitelja različite liste), listu zahtjeva (različite liste za različite vrste zahtjeva)
- Student svojim slanjem zahtjeva (npr. slanje zahtjeva za podizanje knjige) taj zahtjev se upisuje u listu zahtjeva pomoću metode dodajZahtjev
- Upravitelji u zavisnosti koji je zahtjev mogu odobriti ili odbiti zahtjev koji se brise iz liste zahtjeva te ukoliko je odobren izvršava se akcija za koju je zahtjev poslan
- Od konstruktora ova klasa sadrži samo konstruktor bez parametara u kojem su upravitelji već postavljeni, dok su ostali atributi na podrazumijevanim vrijednostima
- U njoj se nalaze sve metode za obradu/potvrdu i odbijanje zahtjeva od strane upravitelja
- Sadrži sve metode koje su u prethodnoj verziji bile u klasi upravitelja soba, upravitelja bibliotekom i upravitelja hranom.

Upravitelj za zaduživanje soba

Uposlenik studentskog doma s najvećim privilegijama. Ima uvid na sve zahtjeve za zaduživanje sobe, produžavanja boravka kao i razduživanja sobe.

Upravitelj može odbiti ili prihvatiti zahtjev.

- Klasa naslijeđena iz apstraktne klase Osoba (generalizacija)
- Dodatni atributi ove klase su:
 - korisničko ime (string)
 - password (string)
 - id (int)
- Ova klasa od metoda sadrži standardne gettere i settere.
- Konstruktor sa svim parametrima

Upravitelj bibliotekom

Uposlenik doma koji je zadužen za sve aktivnosti povezane s bibliotekom. To uključuje vođenje evidencije o knjigama, zahtjevima za podizanje, zahtjevima za vraćanje. Ima zadatak da svim studentima omogući podizanje knjige s tim da se vodi računa da sve prethodne knjige moraju biti vraćene. Ima mogućnosti unosa novih knjiga, kao i brisanja.

- Klasa naslijeđena iz apstraktne klase osoba (generalizacija)
- Dodatni atributi ove klasu su:
 - korisničko ime (string)
 - id (int)
- Ova klasa će imati metode gettere I setter.
- Konstruktor sa svim parametrima.

Opis klasa vezanih za upravitelja hranom i sobom za zabavu

U ovome slučaju imamo 2 osnovna aktera: upravitelja hranom i upravitelja sobom za zabavu. To su ujedno i 2 klase koje će nam biti potrebne. S obzirom da će ove 2 klase posjedovati nekoliko istih atributa kao i ostale upravitelj klase (ime, prezime itd), one će nasljeđivati apstraktnu klasu osoba. Tako da u tom slučaju imamo vezu generalizacije sa apstraktnom klasom Osoba

Klasa upravitelj hranom:

Atributi:

Atributi naslijeđeni iz apstraktne klase Osoba

Menu

Evidencija zaposlenih radnika

Metode:

Dodavanje jela (Jelo jelo)

Uklanjanje jela sa menua (Jelo jelo)

Zapošljavanje radnika (Radnik radnik)

Otpuštanje radnika (Radnik radnik)

Atribut evidencija zaposlenih radnika u klasi Upravitelj hranom je zapravo nova, kontejnerska klasa. Ona služi za čuvanje svih zaposlenih u sebi jer donekle ne bi bilo smisleno čuvati listu zaposlenih Radnika u klasi Upravitelja hranom. Također, u potrebi za proširenjem aplikacije (npr. dodavanjem evidentiranja zaposlenih u nekoj drugoj klasi), samo je potrebno dodati kao atribut klasu Evidencija zaposlenih radnika u željenu klasu.

Također, bitno je napomenuti da je veza između ove klase i klasa „Menu“ i „Evidencija zaposlenih radnika“ veza asocijacije (tačnije agregacije) jer obje ove klase mogu nastaviti postojati ako obrišemo klasu Upravitelj hranom.

Klasa Evidencija zaposlenih radnika:

Atributi:

Lista objekata tipa Radnik

Metode:

Zaposli radnik (Radnik radnik)

Otpusti radnika (Radnik radnik)

Zamijeni radnika (Radnik stariRadnik, Radnik noviradnik)

Ova klasa se u datom trenutku nalazi u sklopu klase Upravitelj hranom, ali po potrebi se lagano može uključiti u bilo koju drugu „Upravitelj“ klasu.

Također, veza između ove klase i klase „Radnik“ je veza kompozicije jer u slučaju brisanja klase Radnik, klasa „Evidencija zaposlenih radnika“ ne može postojati, odnosno nema smisla.

Klasa Menu:

Atributi:

Lista objekata tipa Jelo

Metode:

Dodaj jelo (Jelo jelo)

Obriši jelo (Jelo jelo)

Ažuriraj jelo (Jelo staroJelo, Jelo novoJelo)

Slično kao kod klasa Evidencija zaposlenih radnika i Radnik, veza između klasa Menu i jelo je veza asocijacije (tačnije kompozicije) jer ako obrišemo klasu Jelo, klasa Menu nema smisla.

Klasa Jelo:

Atributi:

Naziv jela

Lista objekata tipa String (sastojci)

Cijena jela (u studentskim bonovima)

Metode:

Klasični geteri i seteri za sve atribute

Klasa Upravitelj sobom za zabavu:

Atributi:

Atributi naslijeđeni iz apstraktne klase Osoba

Evidencija termina u sobi

Metode:

Evidentiraj termin (Termin traženiTermin)

Otkazi termin (Termin termin)

Što se tiče atributi Evidencija termina u sobi, on je ekvivalent atributu Evidencija zaposlenih radnika. Napravljena je posebna klasa u tu svrhu iz više razloga. Kao prvo, ne bi imalo toliko smisla čuvati listu termina u klasi „Upravitelj sobom za zabavu“. Kao drugo, na ovaj način je istu klasu moguće koristiti u različitim klasama (sobama). U slučaju dodavanja neke nove prostorije u kojoj je potrebno rezervisati termine, veoma lagano možemo istu klasu ponovno upotrijebiti.

.

Klasa Evidencija termina u sobi:

Atributi:

Lista objekata tipa Termin

Metode:

Dodaj termin (Termin noviTermin)

Otkazi termin (Termin termin)

Pomjeri termin (Termin, Date/Calendar novoVrijeme)

Svrha i cilj ove klase su već objašnjeni. Potrebno je samo još dodati da veza između ove klase i klase Termin je, ponovno, veza asocijacije, samo u ovom slučaju je to kompozicija. Brisanjem klase Termin nije smisleno da postoji klasa Evidencija termina u sobi.

Klasa Termin:

Atributi:

Student (koji je zakazao termin)

Vrijeme termina (Date ili neki sličan tip podataka)

Metode:

Standardni geteri i seteri sa navedene attribute

Bitna napomena: U svim klasama se podrazumijevaju geteri i seteri, te konstruktor sa svim parametrima, osim u nekim specijalnim slučajevima (koji će biti naglašeni). Dodatno, klase u kojim je eksplicitno navedeno postojanje getera i setera su one najjednostavnije klase koje nemaju nekih drugih posebnih metoda.

Veze klasa

asocijacija – veza samo preko metoda

agregacija - može opstati bez klase s kojom je vezana (nije samo veza preko metoda)

kompozicija – ne može opstati bez klase s kojom je vezana

- Osoba – sve klase akteri izvedeni iz ove klase pa su sljedeće klase njena generalizacija:
 - Student
 - Upravitelj zaduzivanjem
 - Upravitelj hranom
 - Upravitelj sobom za zabavu
 - Upravitelj bibliotekom
- Student
 - Atributi - /
 - Operacije – Knjiga (asocijacija)
- Knjiga
 - Atributi – Student (agregacija, jača je veza od asocijacije pa je veza između studenta i knjige agregacija)
 - Operacije - /
- Zahtjev – klase zahtjeva su izvedeni iz ove klase pa su sljedeće klase njena generalizacija:
 - Zahtjev za promjenu stanovanja
 - Zahtjev za podizanje knjige
 - Zahtjev za rezervaciju sobe
 - Atributi – Student (agregacija)
 - Operacije - /
- Zahtjev za promjenu stanovanja

- Atributi - /
- Operacije - /
- Zahtjev za podizanje knjige
 - Atributi – Knjiga (agregacija)
 - Operacije - /
- Zahtjev za rezervaciju sobe
 - Atributi - /
 - Operacije - /
- Studentski dom
 - Atributi – Student (agregacija), Upravitelj zaduzivanja (kompozicija), Upravitelj bibliotekom (kompozicija), Upravitelj sobom za zabavu (kompozicija), Upravitelj hranom (kompozicija), Zahtjev (kompozicija)
 - Operacije - /

Između klase Upravitelj hranom i klasa Menu i Evidencija zaposlenih radnika se javlja veza asocijacije (tačnije agregacije) jer obje ove klase mogu nastaviti postojati ako obrišemo klasu Upravitelj hranom.

Veza između klase Evidencija zaposlenih radnika i klase Radnik je veza kompozicije jer u slučaju brisanja klase Radnik, klasa Evidencija zaposlenih radnika ne može postojati, odnosno nema smisla da ta klasa više postoji.

Slično kao kod klase Evidencija zaposlenih radnika i Radnik, veza između klase Menu i jelo je veza asocijacije (tačnije kompozicije) jer ako obrišemo klasu Jelo, klasa Menu nema smisla.

Između klase Upravitelj sobom za zabavu i Evidencija termina u sobi imamo vezu asocijacije (tačnije agregacije) jer brisanjem klase Evidencija termina u sobi i dalje je smisleno da klasa Upravitelj sobom za zabavu nastavi postojati.

Između klase Evidencija termina u sobi i Termin, slično kao kod klase Evidencija zaposlenih radnika, imamo vezu kompozicije jer brisanjem klase Termin nema smisla da klasa Evidencija termina u sobi nastavi postojati.

Bitno je još napomenuti da između klase Upravitelj hranom, Upravitelj sobom za zabavu i aspraktne klase Osoba imamo vezu generalizacije. To je organizirano na taj način zbog postojanje nekoliko atributa koje sve „Upravitelj“ klase imaju (npr. ime, prezime i neki drugi lični podaci).