

# Termin5 zadaci

## 1 ZADATAK

Kreirati novi projekat koji će imati naziv Termin5Zadaci. Napraviti paket zadatak5 u src folderu. U paketu zadatak5 napraviti podpakete:

- a) model
- b) zadaci

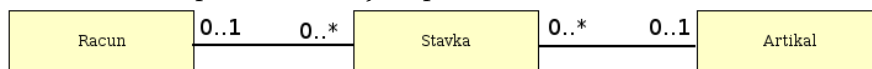
Direktno u folderu projekta napraviti novi folder lib i u njega prekopirati jar fajlove:

- a) sqlite-jdbc-3.8.11.2.jar
- b) ormlite-jdbc-5.0.jar
- c) ormlite-core-5.0.jar

iz projekta Termin5Primeri. Iz Intellij okruženja pritisnuti desnim klikom miša na svaki od jar fajlova i otići na opciju Add as Library, na dnu padajućeg menija, kako bi bili prepoznati prilikom pokretanja zadataka.

## 2 ZADATAK PRODAVNICA

Napisati klase za entitete prodavnice koji su prikazani na slici 2.1.



Slika 2.1: Entiteti prodavnice

U projektu Termin5Zadaci u folderu zadaci napraviti pet klasa:

- a) Zadatak1KreiranjeTabela
- b) Zadatak2DodavanjeBrisanjeIzmenaVrednosti
- c) Zadatak3BidirekcionaVeza
- d) Zadatak4PretragaVrednosti
- e) Zadatak5BrisanjeRacunaStavkiTransakcija

Unutar svake klase napisati main metodu:

```
public static void main(String[] args)
```

Unutar main metoda napisati potrebne naredbe za konekciju sa bazom i njih obuhvatiti try catch blokom naredbi.

## 2.1 KREIRANJE KLASA ZA ENTITETE

### 2.1.1 ZADATAK KLASA ARTIKAL

U paketu model napraviti novu klasu Artikal. Za ovu klasu definisati sledeće attribute:

- a) id koji je tipa int
- b) naziv koji je tipa String
- c) opis koji je tipa String
- d) cena koji je tipa double

Svi atributi treba da imaju modifikator pristupa postavljen na private. Za klasu Artikal treba dodati i statičke attribute koji takođe imaju postavljenu i ključnu reč final. Treba dodati sledeće statičke atributa i njihove vrednosti:

- a) POLJE\_NAZIV vrednost je: "naziv"
- b) POLJE\_OPIS vrednost je: "opis"
- c) POLJE\_CENA vrednost je: "cena"

Za klasu Artikal dodati anotaciju DatabaseTable i postaviti vrednost atributa tableName, anotacije DatabaseTable, na "artikal". Za attribute klase Artikal dodati anotacije DatabaseField i postaviti za vrednost atributa columnName, anotacije DatabaseField, odgovarajuće vrednosti statičkih atributa. Za atribut canBeNull, anotacije DatabaseField, postaviti vrednost false, da bi vrednosti svih atributa klase Artikal morale biti unete. Za atribut id klase Artikal postaviti da mu se automatski generiše vrednost primarnog ključa postavljajući vrednost atributa generatedId, anotacije DatabaseField, na true. Klasa Artikal se nalazi na više kraju veze Stavke i Artikal klasa. Potrebno je dodati atribut stavke koji je tipa ForeignCollection<Stavka> kao odgovarajući atribut za taj kraj veze. Za atribut stavke treba dodati i anotaciju ForeignCollectionField i za vrednost atribut foreignFieldName te anotacije postaviti "artikal". Za klasu Artikal napisati i konstruktor bez parametara i konstruktor koji očekuje parametre naziv, opis i cena. Za odgovarajuće attribute dodati i get i set metode. Za statičke attribute ne treba da se dodaju get i set metode. Redefinisati toString metodu, tako da prikazuje vrednosti atributa id, naziv, opis i cena.

### 2.1.2 ZADATAK KLASA RACUN

U paketu model napraviti novu klasu Racun. Za ovu klasu definisati sledeće attribute:

- a) id koji je tipa int
- b) oznaka koji je tipa String
- c) datum koji je tipa Date. Date klasa se nalazi u paketu java.util i potrebno je navesti import naredbu:  
import java.util.Date;  
pre početka klase Racun.

Svi atributi treba da imaju modifikator pristupa postavljen na private. Za klasu Racun treba dodati i statičke attribute koji takođe imaju postavljenu i ključnu reč final. Treba dodati sledeće statičke atributa i njihove vrednosti:

- a) POLJE\_OZNAKA vrednost je: "oznaka"
- b) POLJE\_DATUM vrednost je: "datum"

Za klasu Racun dodati anotaciju DatabaseTable i postaviti vrednost atributa tableName, anotacije DatabaseTable, na "racun". Za attribute klase Racun dodati anotacije DatabaseField i postaviti za vrednost atributa columnName, anotacije DatabaseField, odgovarajuće vrednosti statičkih atributa. Za atribut canBeNull, anotacije DatabaseField, postaviti vrednost false, da bi vrednosti svih atributa klase Racun morale biti unete. Za atribut id klase Racun postaviti da mu se automatski generiše vrednost primarnog ključa postavljajući vrednost atributa generatedId, anotacije DatabaseField, na true. Klasa Racun se nalazi na više kraju veze Stavke i Racun klasa. Potrebno je dodati atribut stavke koji je tipa ForeignCollection<Stavka> kao odgovarajući atribut za taj kraj veze. Za atribut stavke treba dodati i anotaciju ForeignCollectionField i za vrednost atribut foreignFieldName te anotacije postaviti "racun". Za klasu Racun napisati i konstruktor bez parametara i konstruktor koji očekuje parametre oznaka i datum. Za odgovarajuće attribute dodati i get i set metode. Za statičke attribute ne treba da se dodaju get i set metode. Redefinisati toString metodu, tako da prikazuje vrednosti atributa id, oznaka i datum.

### 2.1.3 ZADATAK KLASA STAVKA

U paketu model napraviti novu klasu Stavka. Za ovu klasu definisati sledeće attribute:

- a) id koji je tipa int
- b) kolicina koji je tipa int

Svi atributi treba da imaju modifikator pristupa postavljen na private. Za klasu Stavka treba dodati i statički atribut koji takođe ima postavljenu i ključnu reč final. Treba dodati sledeći statički atribut i njegovu vrednost:

- a) POLJE\_KOLICINA vrednost je: "kolicina"

Za klasu Stavka dodati anotaciju DatabaseTable i postaviti vrednost atributa tableName, anotacije DatabaseTable, na "stavka". Za attribute klase Racun dodati anotacije DatabaseField i postaviti za vrednost atributa columnName, anotacije DatabaseField, odgovarajuću vrednost statičkog atributa. Za atribut canBeNull, anotacije DatabaseField, postaviti vrednost false, da bi vrednost atributa kolicina klase Stavka morala biti uneta. Za atribut id klase Stavka postaviti da mu se automatski generiše vrednost primarnog ključa postavljajući vrednost atributa generatedId, anotacije DatabaseField, na true. Klasa Stavka se nalazi na jediničnom kraju veze Stavke i Racun klasa i Stavka i Artikal klasa. Potrebno je dodati attribute:

- a) racun koji je tipa Racun

b) artikal koji je tipa Artikal

kao odgovarajuće attribute za taj kraj veze. Za ove attribute postaviti anotaciju DatabaseField i za vrednosti atributa foreign, anotacije DatabaseField, postaviti vrednost true, da bi odredili da su ovo atributi koji odgovaraju koloni stranog ključa. Za vrednost atributa foreignAutoRefresh, anotacije DatabaseField, postaviti vrednost true, da bi se prilikom učitavanja stavki iz baze učitale i vrednosti za račune i artikle. Za vrednost atributa columnName, anotacije DatabaseField, postaviti vrednosti "racun\_id" i "artikal\_id" u zavisnosti za koji atribut je postavljena anotacija. Za klasu Stavka napisati i konstruktor bez parametara i konstruktor koji očekuje parametre kolicina, racun i artikal. Za odgovarajuće attribute dodati i get i set metode. Za statički atribut ne treba dodavati get i set metode. Redefinisati toString metodu, tako da prikazuje vrednosti atributa id i kolicina.

## 2.2 ZADATAK KREIRANJE TABELA

U klasi Zadatak1KreiranjeTabela, unutar main metode, pozvati statičke metode klase TableUtils za brisanje tabela dropTable i metode za kreiranje tabela createTable i proslediti im klase u odgovarajućem redosledu. Za brisanje tabela redosled je:

- Stavka
- Artikal
- Racun

Za kreiranje tabela redosled je:

- Artikal
- Racun
- Stavka

## 2.3 ZADATAK DODAVANJE, BRISANJE I IZMENA VREDNOSTI

U klasi Zadatak2DodavanjeBrisanjeIzmeneVrednosti kreirati tri statička atributa:

- artikalDao
- racunDao
- stavkaDao

ovi atributi za tip imaju Dao generičku klasu, kojoj za prvu vrednost treba postaviti klasu odgovarajućeg entiteta, a za drugu vrednost klasu kojeg tipa je primarni ključ. Ostatak funkcionalnosti implementirati unutar main metode. Instancirati odgovarajuće Dao objekte pozivom createDao statičke metode klase DaoManager. Pozvati metodu clearTable, klase TableUtils, da se obrišu sve vrednosti tabela. Za brisanje vrednosti iz tabela redosled je:

- Stavka
- Artikal
- Racun

### 2.3.1 ZADATAK DODAVANJE VREDNOSTI

Instancirati objekte klase Artikal sa sledećim vrednostima atributa:

- a) naziv: "Mleko", opis: "Mleko u flasi 1L" i cena: 40.99
- b) naziv: "Beli hleb", opis: "400g" i cena: 50
- c) naziv: "Crni hleb", opis: "Crni razeni hleb" i cena: 60
- d) naziv: "Jogurt", opis: "Jogurt u tetrapaku 1L" i cena: 90.99

Sve instancirane objekte klase Artikal upisati u bazu pozivom metode create odgovarajućeg Dao objekta.

Instancirati objekte klase Racun sa sledećim vrednostima atributa:

- a) oznaka: "Racun1", datum: new Date()
- b) oznaka: "Racun2", datum: new Date()

Sve instancirane objekte klase Racun upisati u bazu pozivom metode create odgovarajućeg Dao objekta.

Instancirati objekte klase Stavka sa sledećim vrednostima atributa:

- a) kolicina: 1, artikal: objekat Artikal "Mleko" racun: objekat Racun "Racun1"
- b) kolicina: 2, artikal: objekat Artikal "Beli hleb" racun: objekat Racun "Racun1"
- c) kolicina: 3, artikal: objekat Artikal "Crni hleb" racun: objekat Racun "Racun2"

Sve instancirane objekte klase Stavka upisati u bazu pozivom metode create odgovarajućeg Dao objekta.

### 2.3.2 ZADATAK IZMENA VREDNOSTI

Za objekat Racun sa oznakom "Racun1" promeniti oznaku u "RacunPrvi" i sačuvati izmene u bazi pozivom metode update odgovarajućeg Dao objekta.

### 2.3.3 ZADATAK BRISANJE VREDNOSTI

Instancirati objekte klase Artikal sa sledećim vrednostima atributa:

- a) naziv: "Voda", opis: "Flasa vode od 1,5L" i cena: 70

Objekat klase Artikal upisati u bazu pozivom metode create odgovarajućeg Dao objekta. Pozivom metode queryForAll odgovarajućeg Dao objekta vratiti sve artikle iz baze i koristeći for petlju prikazati sve Artikle. Pozivom metode queryById pronaći dodati objekat klase Artikal sa nazvom "Voda" i obrisati pozivom metode delete odgovarajuće Dao klase. Pozivom metode queryForAll odgovarajućeg Dao objekta vratiti sve artikle iz baze i koristeći for petlju prikazati sve Artikle. Proveriti da se više ne prikazuje Artikal sa nazivom "Voda".

### 2.4 ZADATAK PRIKAZ SVIH STAVKI RACUNA

U klasi Zadatak3BidirekcionaVeza kreirati tri statička atributa:

- artikalDao
- racunDao
- stavkaDao

ovi atributi za tip imaju Dao generičku klasu, kojoj za prvu vrednost treba postaviti klasu odgovarajućeg entiteta, a za drugu vrednost klasu kojeg tipa je primarni ključ. Ostatak funkcionalnosti implementirati unutar main metode. Instancirati odgovarajuće Dao objekte pozivom createDao statičke metode klase DaoManager.

Metodom queryForAll odgovarajuće Dao klase vratiti sve račune iz baze. Koristeći for petlju prikazati sve vrednosti. Pozivom metode queryForEq nad odgovarajućim Dao objektom pronaći sve račune koji za vrednost atributa oznaka imaju postavljenu vrednost "Racun2". Iz vraćene kolekcije preuzeti element sa indeksom 0 i preuzeti njegove stavke koristeći odgovarajuću getStavke metodu. Za kolekciju ForeignCollection<Stavka> vratiti objekat CloseableIterator<Stavka> i koristeći hasNext i next metode proći kroz sve stavke i prikazati stavku i artikal stavke.

### 2.5 PRETRAGA VREDNOSTI

U klasi Zadatak4PretragaVrednosti kreirati tri statička atributa:

- artikalDao
- racunDao
- stavkaDao

ovi atributi za tip imaju Dao generičku klasu, kojoj za prvu vrednost treba postaviti klasu odgovarajućeg entiteta, a za drugu vrednost klasu kojeg tipa je primarni ključ. Ostatak funkcionalnosti implementirati unutar main metode. Instancirati odgovarajuće Dao objekte pozivom createDao statičke metode klase DaoManager.

### 2.5.1 SELECT IZRAZ

Koristeći QueryBuilder napraviti select iskaz koji vraća listu artikala i pronalazi samo one artikle u bazi koji za vrednost kolone naziv imaju tekst hleb.

Koristiti like metodu Where objekta i porediti sa "%hleb%" šablonom.

### 2.5.2 SELECT IZRAZ SA SELECTARG

Koristeći QueryBuilder napraviti select iskaz koji vraća listu artikala i pronalazi samo one artikle u bazi koji za vrednost kolone opis imaju šablon koji će biti prosleđen sa SelectArg objektom. Za SelectArg objekat prvo postaviti da se traži šablon "%1L%" a za drugu pretragu postaviti da se traži šablon "%u%".

Koristiti like metodu Where objekta za poređenjem sa šablonom.

### 2.5.3 SELECT IZRAZ SA JOIN OPERATOROM

Napisati select izraz koji vraća račune iz baze. Select izraz treba napraviti koristeći join operator i spajanjem vrednosti iz računa sa stavkama i proveravanjem da li je vrednost kolone kolicina u stavkama jednaka vrednosti 2.

Koristi eq metodu za poređenje vrednosti kolone kolicina sa vrednosti 2.

## 2.6 TRANSAKCIJE

U klasi Zadatak5BrisanjeRacunaStavkiTransakcija kreirati tri statička atributa:

- artikalDao
- racunDao
- stavkaDao

ovi atributi za tip imaju Dao generičku klasu, kojoj za prvu vrednost treba postaviti klasu odgovarajućeg entiteta, a za drugu vrednost klasu kojeg tipa je primarni ključ. Ostatak funkcionalnosti implementirati unutar main metode. Instancirati odgovarajuće Dao objekte pozivom createDao statičke metode klase DaoManager.

Koristeći kod iz klase Zadatak3BidirekcionalnaVeza koji prolazi kroz sve stavke računa sa oznakom "Racun2", sve stavke ubaciti u posebnu listu za brisanje. Pozvati statičku metodu callInTransaction klase TransactionManager kojoj treba proslediti anonimnu klasu kreiranu za Callable<Void>. U kreiranoj anonimnoj klasi Callable<Void> unutar metode public Void call() throws Exception

ubaciti da se prvo obrišu sve stavke sačuvane u listi za brisanje, a zatim obrisati i račun sa oznakom "Racun2" unutar iste metode.