# Vežbanje za Kolokvijum 1

**Kreirati UML projekat.** Projektu i paketu dodelite naziv: **ImePrezimeBrojIndeksa** (primer: PetarPetrovic2015300100). Sve zadatke je potrebno uraditi u kreiranom projektu.

1. **Kreirati dijagram klasa za aplikaciju "e-Knjižara":**

* **Klijenti** ima atribute: KlijentID, Username, Password, email.
* Klase **Porudžbina** i **Faktura** su generalizovane u klasu **Dokumenta**, gde se nalaze sledeći zajednički atributi: BrDokID, Datum, Iznos i TipDokumenta.
* **StavkeDokumenta** sadrže atribute: RbrStavke, PoručenaKoličina i Popust.
* **e-Knjige** sadrže sledeće atribute i metode: eKnjigaID, Naziv, Cena, StanjeNaZalihama, getStanje(e-knjigaID) i setCena().
* **Izdavač** sadrži atribute: IzdavačID i NazivIzdavača.

Potrebno je adekvatno povezati klase i **dopuniti klase sa stranim ključevima!**

Potrebno je za svaku relaciju prikazati odgovarajući tip veze tj. **multiplikativnost**.

1. **Kreirati dijagram klasa za aplikaciju "Upis na kurs stranih jezika":**

* **Klijenti** koji se upisuju na kurseve mogu biti Fizička ili Pravna lica. Zajednički atributi su: KlijentID, Adresa, Telefon i e-mail.
  + **Pravna lica** imaju atribute: Naziv i PIB.
  + **Fizička lica** imaju atribute: Ime, prezime i JMBG.
* **Kurs** ima atribute: KursID i Opis. Metod je upisNaKurs().
* Usled velikog broja polaznika kursa, isti kurs može da sadrži više grupa. **Grupa** ima atribute: BrGrupeID, UkupanBrojPolaznika.
* **RasporedKlijenta** sadrži atribute: TerminKursa i evidencijaPrisustva.
* Klijent može da ima minimum jedan, a maksimalno više rasporeda.
* Jedan kurs može da ima više rasporeda.
* **Jezik** sadrži atribute JezikID i NazivJezika.
* Jedan jezik može da se drži na više kurseva.

Potrebno je adekvatno povezati klase i **dopuniti klase sa stranim ključevima!**

Potrebno je za svaku relaciju prikazati odgovarajući tip veze tj. **multiplikativnost**.

1. **Kreirati dijagram klasa za aplikaciju "Prijavljivanje ispita na fakultetu":**

* Studenti fakulteta mogu biti redovni ili zaposleni. Zaposleni studenti pored atributa broj indeksa, ime, prezime i email imaju i naziv kompanije gde rade.
* Studenti prijavljuju ispite u određenim ispitnim rokovima. Ispitni rok ima atribute: ispitniRokID, nazivIspitnogRoka, datumOd, datumDo.
* PrijavaIspita sadrži atribute: PrijavaID, DatumPrijave, Iznos i Ocena.
* Jedna prijava odnosi se samo na jedan ispit. Atributi ispita su: IspitID, datumIspita, VremeIspita, Prostorija.
* Ispit se odnosi na jedan predmet i jednog profesora koji je nosilac predmeta.
* Aributi predmeta su: PredmetID, nazivPredmeta i fondČasova.
* Atibuti profesora su: profID, ime, prezime i zvanje.

Potrebno je adekvatno povezati klase i **dopuniti klase sa stranim ključevima!**

Potrebno je za svaku relaciju prikazati odgovarajući tip veze tj. **multiplikativnost**.

1. **Kreirati dijagram stanja za klasu “Online Klijent”:**

* Kada se korisnik registruje na sajtu postaje Potencijalni klijent, prilikom čega mu se generiše UserID. Tokom ovog stanja, korisnikova kretanja se prate (tracking).
* Kada korisnik postavi svoju prvu porudžbinu postaje aktivan korisnik (setActive(true)) i prelazi u stanje Aktivan klijent.
* Tokom stanja Aktivan klijent, klijent se obaveštava o aktuelnim promocijama.
* Kada su porudžbine klijenta veće od 10.000 din, klijent postaje VIP (setVIP(true)) i prelazi u stanje VIP klijent.
* Ulaskom u stanje VIP klijent, korisniku se kreira VIP kartica i tokom ovog stanja dobija izveštaj o posebnim popustima.
* Ukoliko je klijent tokom poslednjih 6 meseci imao porudžbine manje ili jednako 10.000 din, prelazi iz stanja VIP u Aktivan klijent (setVIP(false)).
* Ukoliko u poslednjih 6 meseci klijent nije imao porudžbina iz stanja Aktivan prelazi u stanje Pasivan klijent (setPassive(true)).
* Tokom stanja Pasivan klijent, prikazuju mu se atraktivne promocije. Ukoliko je od poslednjeg ažuriranja prošlo godinu dana, klijent se arhivira (setArchive). Kada klijent poruči, iz pasivnog stanja prelazi u aktivno (setPassive(false)).
* Kada je u sistemu uočena neka malverzacija klijenta (pokušaj zloupotrebe podataka, transakcija ili sl.), klijent se postavlja na crnu listu (setBlackListUser) i iz aktivnog prelazi u stanje Klijenata na Crnoj listi.

1. **Primeniti odgovarajući dizajn patern za klasu Porudžbina:**

* Potrebno je kreirati klasu Porudžbina za sistem poručivanja robe. U zavisnosti od stanja porudžbine izvršavaće se određene metode: AddOrderLine(), Confirm(), Pay(), Ship() i Cancel(). Porudžbina može biti u stanju Inicijalna porudžbina, Potvrđena porudžbina, Plaćena porudžbina, Isporučena porudžbina i Otkazana porudžbina.

1. **Primeniti odgovarajući dizajn patern na aplikaciju za upravljanje projektima.**

* U zavisnosti da li je projekat kompanije (CompanyProject) ili projekat Univerziteta (UniversityProject) kreira se odgovarajuća instanca klase bilo Akademski projekat (AcademicProject) ili Komercijalni projekat (CommercialProject).