

1. Opis i sadržaj vežbe

- Upoznavanje sa osnovnim konceptima objektno orijentisanog programiranja u Javi,
 - Modelovanje entiteta iz realnog sveta sa klasama,
 - Definisanje strukture klasa, atributi i metode, podrazumevane vrednosti atributa, ključne reči *null* i *this*
 - Definisanje veze između dva objekata (1-1, 1-N)
 - modifikatori pristupa, enkapsulacija podataka, get/set metode, read/write, read-only i write-only svojstvi,
 - Konstruktori - Inicijalizacija objekata (rad sa konstruktorima klase)
- Nizovi i objekti modelovanih entiteta, *ArrayList* i Objekti modelovanih entiteta
 - Niz koji sadrži reference ka objektima modelovanih entiteta,
 - *ArrayList*-a koja sadrži reference ka objektima modelovanih entiteta,
 - Poređenje
- Klasa *Object*, metode *toString* i *equals*,

Dodatni materijal:

- Plitka i duboka kopija
- Relacije između objekata,
 - Definisanje mogućih relacija između entiteta,
 - UML, dijagram klasa i simboli za predstavu relacija,
 - Pretvaranje relacije iz modela klasa u Java kod.
- *Immutable Objects*,
 - Šta su i šta omogućavaju
 - Primeri
 - Kreiranje immutable objekata iz korisničkih klasa čiji se sadržaj ne može menjati posle kreiranja.

2. Literatura

materijali/slajdoviTermin3.pptx

3. Primeri

primer01 - Klase, atributi, metode, predefinisane vrednosti atributa, *null* literal

primer02 - Veza između dve klase, atributi klase koji nisu primitivni tipovi

tj. atributi koji sadrže referencu ka drugim objektima, Liste koje sadrže reference ka objektima modelovanih klase

primer03 - Prava pristupa, get i set metode

primer04 - Konstruktori

primer05 – Kompleksni primer, Detaljan rad sa klasom *ArrayList*

Primeri dodatno:

primerDodatnoImmutableObjects – Modelovanje Java klase čije su instance immutable objekti.

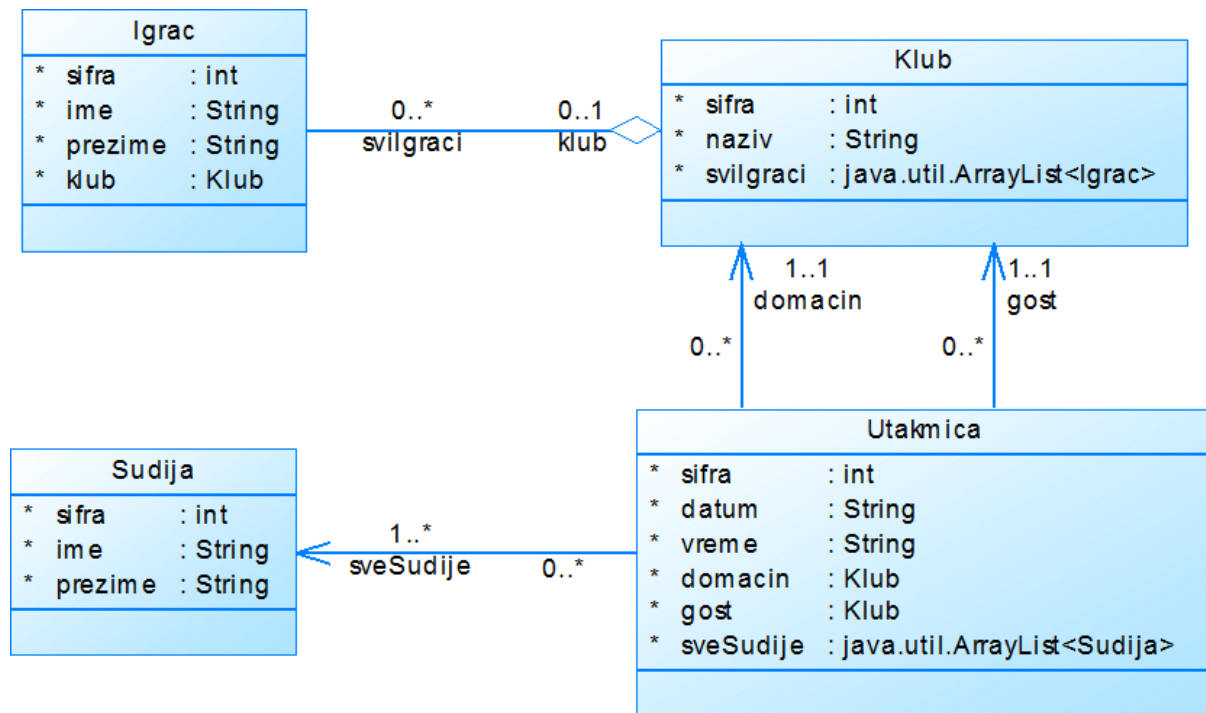
4. Zadaci na času

----- PODSEĆANJE SMERNICA ZA MODELOVANJE:

1. Ako u tekstu zadatka identifikujemo varijaciju reči više znamo da možemo imati atribut koji će biti Set/Lista/Mapa i koji će sadržati reference ka više objekata.
2. Ako u tekstu zadatka identifikujemo reči jedan znamo da možemo imati atribut koji će sadržati referencu ka jednom objektu.

Zad01 – Modelira se košarkaški savez. Sve što je crvenom

- **Klub** ima svoju *šifru* i *naziv*. U klubu igra **više** igrača, **a ne mora** da igra **ni jedan**. Prethodno na slici opisuje kardinalitet 0..* *sviIgrači*, što dovodi do kreiranja atributa *sviIgraci* u klasi Klub.
- **Igrač** ima svoju *šifru*, *ime* i *prezime*. Igrač **igra u jednom** klubu. Prethodno na slici opisuje kardinalitet 0..1 *klub*, što dovodi do kreiranja atributa *klub* u klasi Igrač.
- Kako u tekstu zadatka nigde ne stoji da igrači ne mogu da postoje bez kluba, relacija između ta dva entiteta je *agregacija*. Da igrači ne mogu da postoje bez kluba veza bi bila *kompozicija*.
- **Sudija** ima svoju *šifru*, *ime* i *prezime*. Za klasu Sudija nisu tražene nikakve relacije ka ostalim entitetima, te u klasi Sudija neće biti dodatnih atributa. Posledica je da ćemo teže dobiti informacije koje se odnose na sudiju a nalaze se u drugim entitetima npr. koje je sve utakmice sudio određeni sudija.
- Klub igra **utakmicu** sa drugim klubom. Za svaku utakmicu **se zna datum** i **vreme** odigravanja, klub koji je bio *domaćin* (na slici opisuje kardinalitet 1..1 *domacin*, što dovodi do kreiranja atributa *domacin* u klasi Utakmica), klub koji je bio *gost* (na slici opisuje kardinalitet 1..1 *gost*, što dovodi do kreiranja atributa *gost* u klasi Utakmica), *broj koševa koje su postigli domaćin i gost* i *sudije* koje su sudile utakmicu. Svaka utakmica između dva kluba ima svoju *šifru*.
 - Jednu utakmicu može da sudi više sudija, a mora da sudi bar jedan. Prethodno na slici opisuje kardinalitet 1..* *sveSudije*, što dovodi do kreiranja atributa *sveSudije* u klasi Utakmica.
 - Klub kao domaćin može da igra utakmicu sa više klubova, a ne mora ni sa jednim. Prethodno na slici opisuje kardinalitet 0..*. Višak informacija za zadatak.
 - Klub kao gost može da igra sa više klubova, a ne mora da igra ni sa jednim. Prethodno na slici opisuje kardinalitet 0..*. Višak informacija za zadatak.



Zad02

Na osnovu teksta zadatka u fajlu `studentska_sluzba_dijagram_klasa.pdf` potrebno je po ugledu na klasu `Student` kreirati klasu `IspitniRok` i smestiti je u paket `prime05` tako da:

- Ispitni rok sadrži (identifikator), naziv, početak i kraj. Za sada početak i kraj predstavite tipom `String`.
- Je potrebno obezbediti:
 - da su svi atributi zaštićeni
 - da postoje više konstruktora (odlučite sami koji su vam potrebni),
 - da postoje metode `vratiTekstualnuReprezentacijuZaIspis()`, `isti(IspitniRok ir)` i `set/get` metode za attribute klase.

Izmeniti test klasu `Test` u okviru koje je potrebno da se kreira lista ispitnih rokova i da se ona ispisuje.

Izmeniti test klasu `TestStudentskaSluzba` tako da ona podrži rad sa ispitnim rokovima.

Test podatke preuzeti (prekopirati) iz fajla `ispitni_rokovi.csv`.

5. Domaći zadaci

Dom01

Modelovati sve preostale entitete iz Studentske Službe koje su prikazane u fajlu „studentska_sluzba_diagram.pdf“. Sve modelovane entitete smestiti je u paket `prime05`.

Dom02

Sve entitete studentske službe pokušati uključiti u test klasu `TestStudentskaSluzba`. Napisati neophodne funkcionalnosti (koje su već date za `Studenta` i `Predmet`).

Dom03

Uradite što više zadataka iz fajla „Dodatni zadaci vežba 3 modelovanje. pdf“.