




1. Grafički simbol pravougaonik u SSA se koristi za prikaz:
  - a) skladišta podataka
  - b) procesa
  - c) interfejsa**
  - d) toka podataka
2. Grafički simbol  na PMOV-u predstavlja:
  - a) jak objekat
  - b) kardinalnost
  - c) agregaciju
  - d) vezu**
3. Za opisivanje dinamike slučaja korišćenja koristi se UML dijagram
  - a) konceptualni dijagram klasa
  - b) konačni dijagram klasa
  - c) objektni dijagram
  - d) dijagram sekvenci**
4. Za kreiranje konceptualnog modela koristi se UML dijagram
  - a) dijagram klasa**
  - b) dijagram sekvenci
  - c) dijagram prelaza stanja
  - d) dijagram komunikacije
5. PMOV služe za kreiranje
  - a) modela procesa
  - b) modela podataka**
  - c) modela zavisnosti između polja istog dokumenta
  - d) modela zavisnosti između polja različitih dokumenata
6. U UML 2.0 postoji generalizacija
  - a) samo aktera
  - b) samo slučajeva korišćenja
  - c) i aktera i slučajeva korišćenja**
  - d) ni aktera ni slučajeva korišćenja
7. Grafički simbol  na PMOV-u predstavlja
  - a) agregaciju**
  - b) jak objekat
  - c) slab objekat
  - d) objekat generalizacije
8. IDEF1X je standard koji se koristi za kreiranje
  - a) modela toka posla
  - b) modela podataka**
  - c) modela procesa
  - d) modela zavisnosti između polja istog dokumenta
9. Za opis strukture tokova i skladišta podataka u SSA koristi se
  - a) dijagram konteksta
  - b) dijagram toka podataka I nivoa
  - c) specifikacija logike primitivnih procesa
  - d) rečnik podataka**

10. Između dva slučaja korišćenja u UML 2.0 koriste se sledeće stereotipizovane veze:
- <<is>> <<has a>>
  - <<used>> i <<not used>>
  - <<extend>> i <<include>>**
  - <<produce>> i <<build>>
11. Dijagram sekvenci prikazuje komunikaciju između
- procesa
  - skladišta podataka
  - objekata**
  - atributa
12. Private vidljivost operacija u UML 2.0 se predstavlja
- znakom minus (-)**
  - znakom plus (+)
  - znakom tilda (~)
  - znakom sharp (#)
13. Na dijagramu slučajeva korišćenja u UML 2.0 dozvoljena je direktna asocijacija između:
- dva aktera
  - dva slučaja korišćenja
  - aktera i slučaja korišćenja**
  - sve navedeno
14. OCL @pre klauzula može se koristiti:
- u svim vrstama OCL ograničenja
  - isključivo u invariant ograničenjima
  - isključivo u precondition ograničenjima
  - isključivo u postcondition ograničenjima**
15. Grafički simbol elipsa u SSA se koristi za prikaz:
- skladišta podataka
  - procesa**
  - interfejsa
  - toka podataka
16. Grafički simbol na PMOV-u  na PMOV-u predstavlja:
- jak objekat
  - objekat specijalizacije
  - slab objekat**
  - objekat generalizacije
17. Dijagrami tokova podataka SSA služe za kreiranje:
- modela procesa**
  - modela podataka
  - modela zavisnosti između polja istog dokumenta
  - modela zavisnosti između polja različitih dokumenata
18. Grafički simbol strelica u SSA se koristi za prikaz:
- skladišta podataka
  - procesa
  - interfejsa
  - toka podataka**

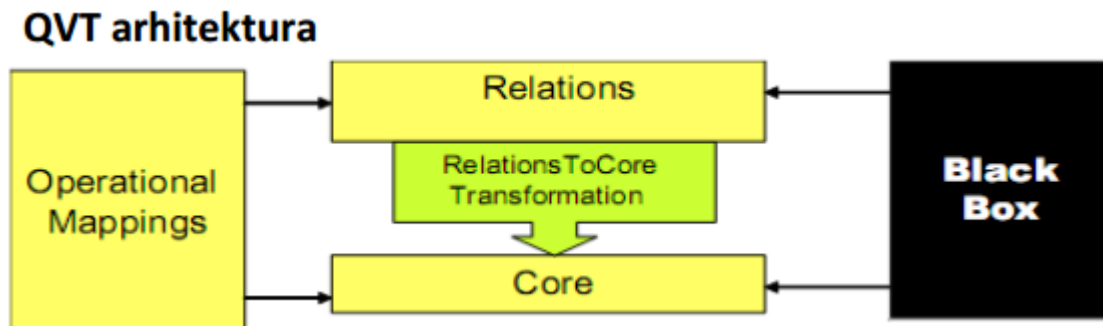
19. Public vidljivost operacija u UML 2.0 se predstavlja
- znakom minus (-)
  - znakom plus (+)**
  - znakom tilda (~)
  - znakom sharp (#)
20. U UML 2.0 između klase i interfejsa uspostavlja se veza:
- <<include>>
  - nasleđivanja
  - <<implementation>>**
  - <<derive>>
21. U SSA, funkcije koje se dalje ne dekomponuju nazivaju se:
- agregirane funkcije
  - funkcije dijagrama konteksta
  - nedekomponovane funkcije
  - primitivne funkcije**
22. U SSA, pored funkcija, mogu se još dekomponovati:
- samo tokovi
  - samo skladišta
  - i tokovi i skladišta**
  - ni tokovi ni skladišta
23. U UML 2.0 u dijagrame interakcije, pored dijagrama sekvenci, spada i:
- dijagram komunikacije**
  - dijagram klasa
  - dijagram prezaza stanja
  - dijagram aktivnosti
24. Na dijagramu slučajeva korišćenja u UML 2.0 dozvoljena je direktna asocijacija između:
- dva aktera
  - dva slučajeva korišćenja
  - aktera i slučaja korišćenja**
  - sve navedeno
25. OCL se koristi za definisanje ograničenja:
- isključivo nad dijagramom klasa
  - nad svim UML dijagramima**
  - isključivo nad dijagramom sekvenci
  - isključivo nad dijagramom klasa i dijagramom sekvenci
26. Protected vidljivost operacija u UML 2.0 se predstavlja
- znakom minus (-)
  - znakom plus (+)
  - znakom tilda (~)
  - znakom sharp (#)**
27. Neobavezna ekskluzivna specijalizacija na PMOV-u određena je kardinalnošću:
- 0,1**
  - 1,1
  - 0,M
  - 1,M

28. U SSA, dozvoljeno je da tok podataka direktno povezuje:
- a) **dva procesa**
  - b) dva interfejsa
  - c) interfejs i skladište
  - d) dva skladišta
29. U UML dijagramu klasa ukoliko ne istaknemo navigaciju podrazumeva se:
- a) jednosmerna veza
  - b) mora da se istekna navigacija
  - c) **dvosmerna veza**
  - d) veza od jake ka slaboj klasi
30. U četvoronivovskoj hijerarhiji, metamodeli se nalaze na nivou:
- a) M0
  - b) M1
  - c) **M2**
  - d) M3
31. Operacija mapiranja u QVT-u može biti:
- a) isključivo dvosmerna
  - b) **isključivo jednosmerna**
  - c) jednosmerna ili dvosmerna
  - d) ništa od navedenog
32. Model za opis strukture sistema je:
- a) DTP strukturne systemske analize
  - b) **dijagram klasa**
  - c) dijagram promene stanja
  - d) dijagram slučajeva korišćenja
33. U rečniku podataka [ ] je oznaka za:
- a) agregaciju
  - b) skup
  - c) neekskluzivnu specijalizaciju
  - d) **ekskluzivnu specijalizaciju**
34. U rečniku podataka / / je oznaka za:
- a) agregaciju
  - b) skup
  - c) **neekskluzivnu specijalizaciju**
  - d) ekskluzivnu specijalizaciju
35. Composition part – of se u UML dijagramu klasa koristi za opis:
- a) metaklase
  - b) **jake posesivnosti**
  - c) skupa klasa
  - d) skupa veza

### 10. Šta je XMI? Šta je QVT? Prikazati arhitekturu QVT-a i dati opis svih komponenti.

-XMI (XML Metadata Interchange), definiše XML tagove preko kojih se može opisati serijalizovan model baziran na MOFu. Služi prvenstveno za razmenu modela.

-QVT (Query/View/Transformation), predstavlja standard za transformacije modela definisan od strane OMGa (Object Management Group).



-Relations jezik- deklarativna specifikacija veza između MOF modela, poseduje grafičku sintaksu

-Core jezik- deklarativni jezik, mali model/jezik koji podržava mapiranje među promenljivim kroz evoluciju uslova definisanih nad ovim promenljivim u odnosu na skup modela.

-Operational Mappings jezik je imperativan jezik koji proširuje relations i core jezike. Oslanja se na OCL.

-Black box implementacija dozvoljava implementaciju složenih algoritama u bilo kom programskom jeziku.

### 10. Objasniti pojam „apstrakcije“ kao opšteg metodološkog pristupa za savladavanje složenosti sistema. Objasniti sledeće apstrakcije: 1) tipizacije, 2) generalizacije i 3) agregacije

- Apstrakcija je kontrolisano uključivanje detalja, "sakrivanje" detalja, odnosno "izvlačenje" opštih karakteristika u opisivanju nekog sistema.

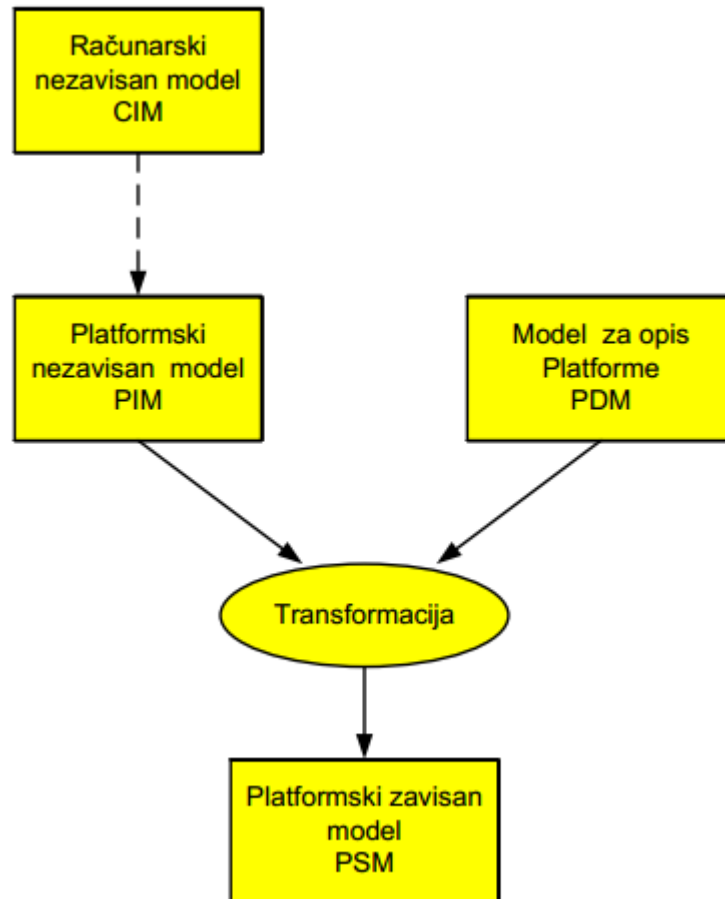
-Klasifikacija ili tipizacija je apstrakcija u kojoj se skup sličnih objekata pretstavlja jednom klasom objekata, odnosno svaki objekat iz posmatranog skupa odgovarajućim tipom objekta.

-Generalizacija je apstrakcija u kojoj se skup sličnih tipova objekata pretstavlja opštijim generičkim tipom (nadtipom). Pod sličnim tipovima objekata ovde se mogu tretirati tipovi objekata koji imaju jedan broj istih (zajedničkih) atributa, tipova veza sa drugim objektima i operacija.

-Agregacija je apstrakcija u kojoj se skup tipova objekata i njihovih veza tretira kao jedinstveni agregirani tip objekta.

**10. Nabrojati i ukratko opisati vrste modela u Modelu vođenom razvoju (MDA/MDD)? Kako su ovi modeli međusobno povezani.**

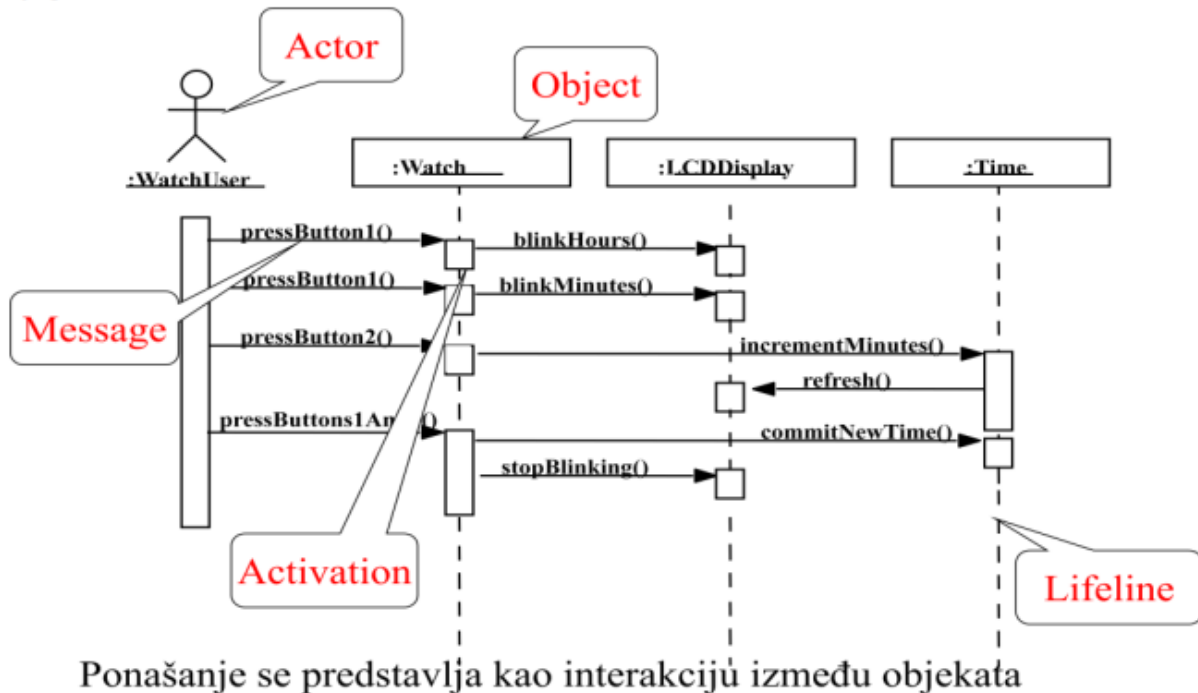
-Model driven zato što se u svim fazama razvoja koriste modeli. Arhitektura nekog sistema, uopšte, je specifikacija njegovih delova i pravila koja definišu način komunikacije između delova. U MDA pristupu definišu se modeli koji se koriste u razvoju softvera i pravila njihovog povezivanja.



- 1) CIM – model odgovarajućeg domena, zajednički rečnik za korisnika i projektanta.
- 2) PIM – model IS nezavisan od implementacione platforme. Specifikacija sistema.
- 3) PDM – model implementacione platforme
- 4) PSM – model IS implementiran u datom okruženju.

## 10. Objasniti sve koncepte UML dijagrama sekvenci

-Uz dijagram komunikacije služi za modelovanje interakcija, prikazujući vremenski redosled poruka.



## 10. Nabrojati sve UML modele koji se koriste za opis dinamike. Dati osnovne koncepte jednog od njih.

- To su dijagrami: sekvenci, kolaboracije, promene stanja, aktivnosti.

// uzeti opis iz predhodnog pitanja.

## 10. Objasniti osnove objektnog pristupa. Objasniti pojam tipa i klase objekata. Kako se ovi pojmovi predstavljaju u UML-u?

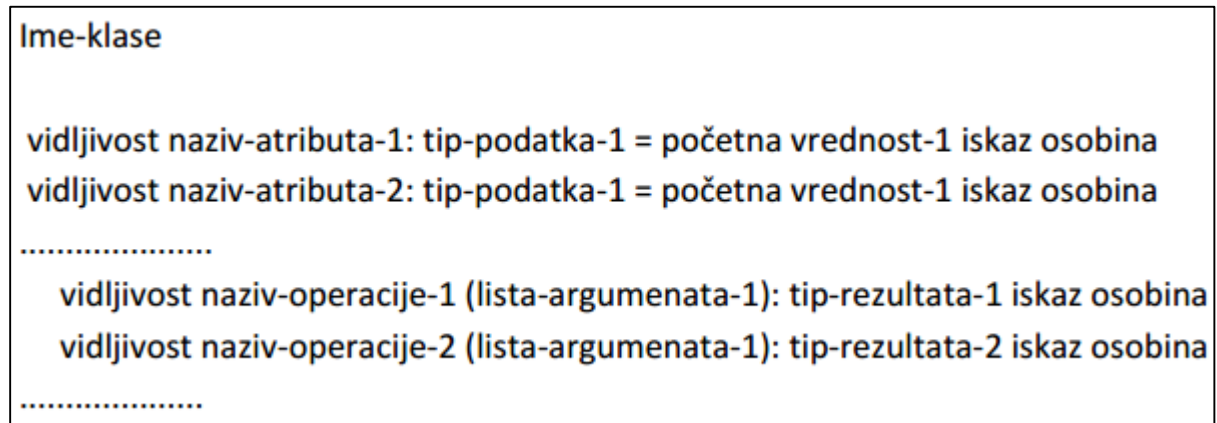
-Sistem je skup povezanih objekata. Pod objektom se podrazumeva entitet koji je sposoban da čuva svoja stanja i koji stavlja na raspolaganje okolini skup operacija preko kojih se ta stanja prikazuju ili menjaju.

-Bitna karakteristika je enkapsulacija (učaurenje), sakrivanje informacija. U strogo objektnim pristupima vidljivi deo objekta je samo specifikacija operacija, a ne i njihova implementacija.

-Pod tipom objekata se podrazumeva kategorija objekata koji imaju isti skup stanja. Neki konkretan objekat se tretira kao pojavljivanje tipa.

-Svaka operacija ima pozvani objekat kao implicitni argument. Njeno ponašanje zavisi od klase kojoj pozvani objekat pripada. Osobina da objekat poziva neku operaciju drugog objekta ne znajući kojoj klasi ovaj objekat pripada naziva se polimorfizam.

-Standardni opis klasa u UML-u:



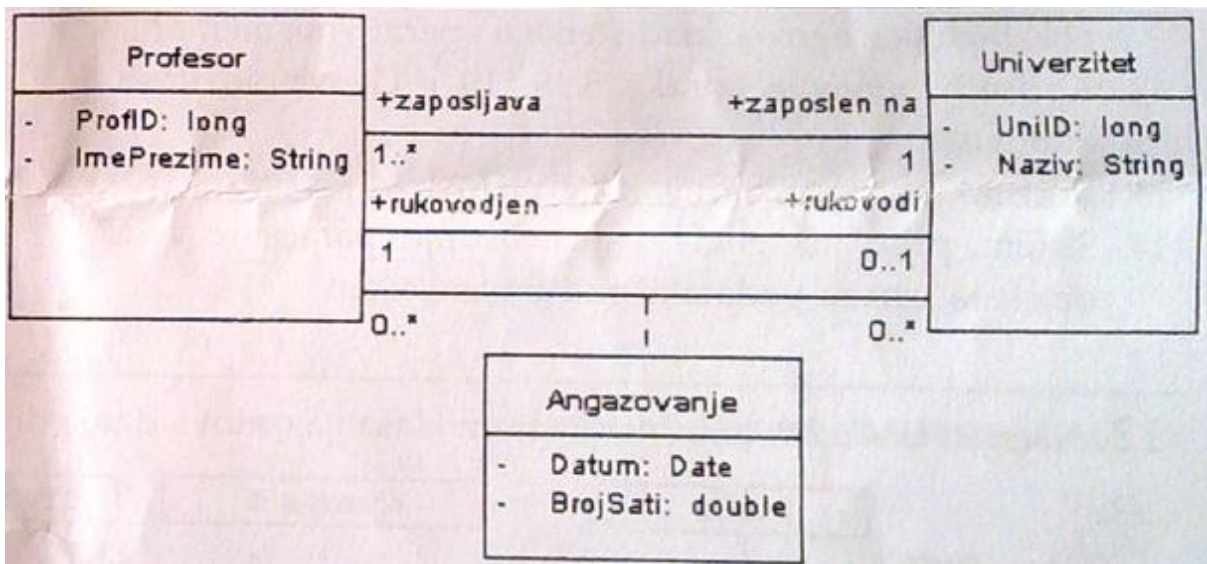
10. Kreirati UML profil kojim se uvode nova proširenja u dijagramu slučajeva korišćenja. Dodaju se dve nove vrste interakcija (asocijacija) između aktera korišćenja „testira“ i „administrira“. Definisano je i ograničenje kojim akter koji ima ulogu „testira“ za konkretni slučaj korišćenja, bilo direktno ili indirektno. Dati i jedan primer dijagrama slučaja korišćenja u kojem se koristi definisani profil.

-69ta i 70ta str u skripti

10. Za konceptualni model prikazan na slici (zad.8) definisati sledeća OCL ograničenja:

-broj sati angazovanja profesora na univerzitetu mora biti veći od 0

-u svakom trenutku, univerzitet mora zapošljavati bar jednog profesora



-77ma – 82ga strana u skripti



## ***METAMODELI***

- 10. Nacrtati metamodel XML dokumenta upotrebom UML dijagrama klasa
- 10. Nacrtati metamodel SSA korišćenjem UML dijagrama klasa.
- 10. Nacrtati metamodel PMOV korišćenjem UML dijagrama klasa
- 10. Nacrtati metamodel UML dijagrama aktivnosti korišćenjem UML dijagrama klasa.
- 10. Nacrtati metamodel UML dijagrama slučajeva korišćenja upotrebom UML dijagrama klasa
- 10. Nacrtati metamodel relacionog modela upotrebom UML dijagrama klasa.
- 10. Nacrtati metamodel UML dijagrama komunikacije korišćenjem UML dijagrama klasa
- 10. Nacrtati metamodel tipova poslovnih dokumenata koji predstavljaju uzor analize „Poslovni dokument“. Na dijagramu predstaviti metamodel strukture tipova dokumenata i model pojavljivanja dokumenta.