

## 16. MODELOVANJE SOFTVERA - UML

**UML - Unified Modeling Language, nastao je 1997. godine udruživanjem različitih pristupa objektno-orijentisanom modelovanju**, predstavlja prvenstveno grafički jezik modelovanja u svim fazama razvoja softvera, počev od specifikacije zahteva, systemske analize, dizajna softvera i arhitekture, dokumentovanja itd. **Autori: Grady Booch, James Rumbaugh i Ivor Jacobsen.** Od 1997. godine je **UML jezik prihvaćen kao standard za objektno orijentisani razvoj softvera, tj. projektovanje i dizajniranje objekata.** Nije namenjen samo za modelovanje softvera, već je ekspresivan dovoljno da bi se primenio i za modelovanje drugih sistema, npr. hardvera, organizacionih sistema i njihovih radnih tokova i sl.

Prva verzija UML 1.0 uključuje devet vrsta dijagrama: **use case dijagram, class dijagram, object dijagram, sequence dijagram, collaboration dijagram, activity dijagram, statechart dijagram, dijagram komponenti (component), dijagram razmeštaja (deployment).**

Druga verzija UML 2.0 iz 2005. godine predstavlja unapređenje u preciznijim pravilima sintakse i semantike i podrške modelovanju kompleksnih sistema. Uključuje dodatne dijagrame: communication dijagram, package dijagram, interaction overview, composite structure dijagram, timing dijagram.

Verzija UML 2.5 je izdata 2015. godine. UML modeli softverske arhitekture razvijaju pre implementacije. UML predstavlja notaciju za opis rezultata objektno-orijentisane analize i dizajna razvijenih pomoću bilo koje metodologije.

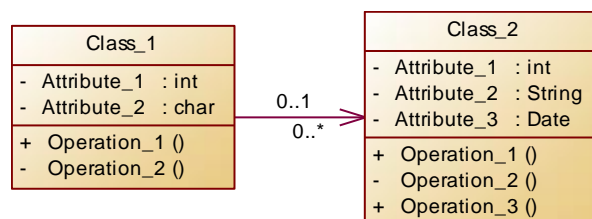
**UML je nezavistan od implementacione platforme.**

**U okviru dizajna softvera može se piše i specifikacija različitih elemenata modela,** ali umesto specifikacije programa može se kreirati i uraditi niz međusobno povezanih UML dijagrama koji ne sadrže nikakvu specifikaciju.

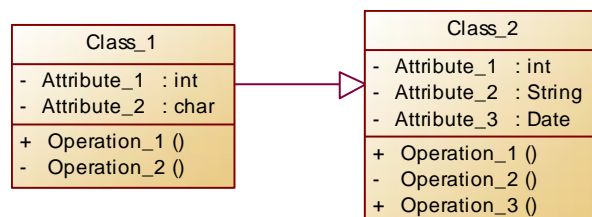
**UML dijagrami su:**

1) **CLASS DIAGRAM – dijagram klasa** predstavlja sredstvo prikaza statičkih odnosno strukturalnih osobina sistema. Sastoji se od simbola klasa, interfejsa i relacija.

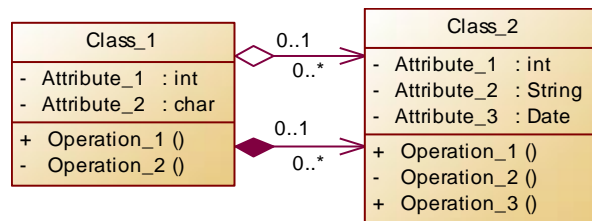
Asocijacije su relacije između klasa koje prikazuju sa koliko objekata druge klase je neki objekat povezan:



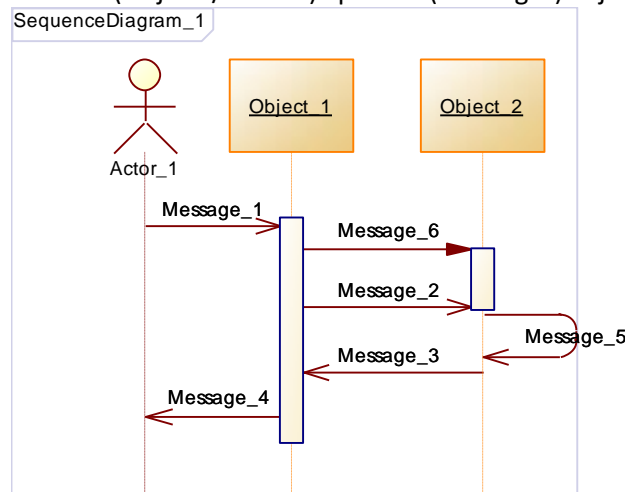
Generalizacija prikazuju kako je neka podklasa povezana sa nadklasom od koje nasleđuje atribute i operacije:



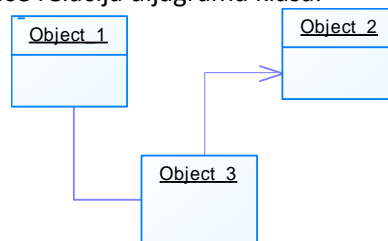
Agregacija (na slici gore) i kompozicija (na slici dole) su odnosi deo-celina:



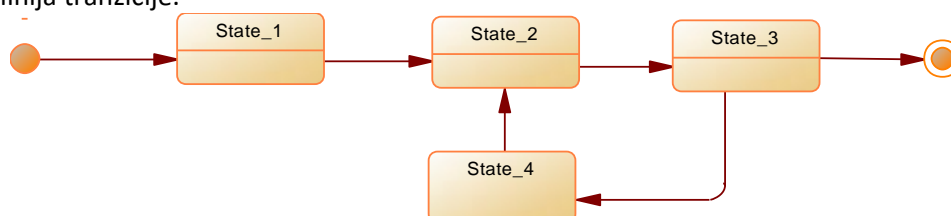
2) SEQUENCE DIAGRAM - **dijagram sekvenci** služi za predstavljanje dinamičkih osobina sistema ili njegovih delova. Glavni fokus dijagrama je prikaz vremenskog sleda slanja i prijema poruka između objekata u cilju ostvarenja zajedničkog zadatka. Elementi dijagrama sekvenci su korisnici (Actors/Use Case), objekti kao instance klasa (Objects/Classes) i poruke (Messages) koje razmenjuju objekti.



3) OBJECT DIAGRAM - **dijagram objekata** sadrži objekte, kao instance pojedinih klasa (pravougaonici) i linkove (pune linije), kao instance relacija dijagrama klasa.

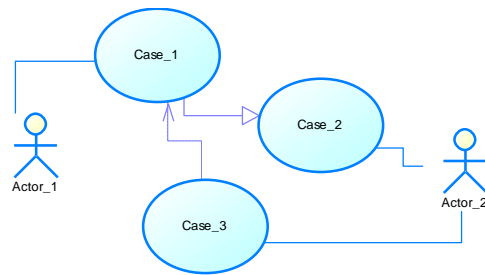


4) STATECHART DIAGRAM - **dijagram stanja** prikazuje promenu stanja objekata, uslovljeni izvršavanjem akcija koje su pokrenute od strane događaja. Prevođenje jednog stanja u drugo prikazuje linija tranzicije.

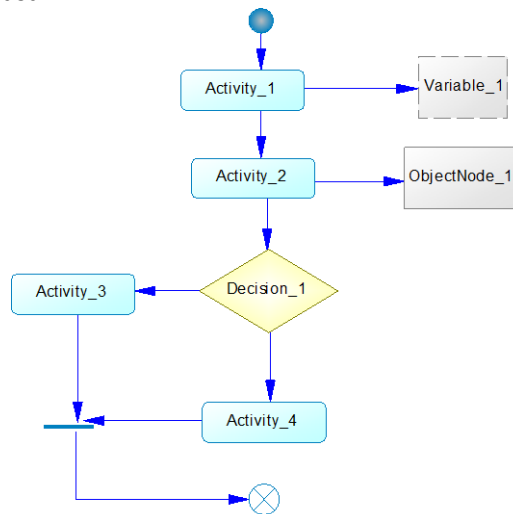


5) COLLABORATION DIAGRAM – **dijagram saradnje**, služi za prikaz dinamičkih osobina sistema. Prikazuje objekata koji šalju i primaju poruke, u cilju izvršavanja zajedničkog zadatka - realizacije slučaja korišćenja ili nekog ponašanja.

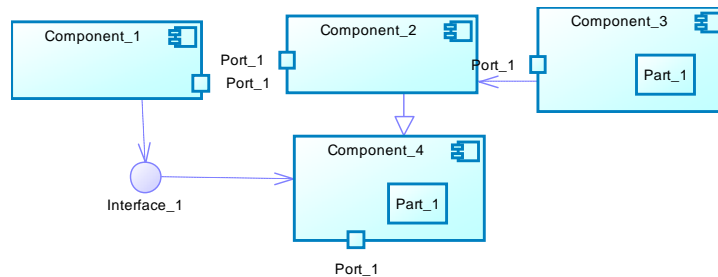
6) USE CASE DIAGRAM – **dijagram slučajeva korišćenja** se sastoji od više slučajeva korišćenja, učesnika i relacija.



7) ACTIVITY DIAGRAM – **dijagram aktivnosti** koji predstavlja tok kontrole aktivnosti, algoritma, operacija/ponašanja klasa.



8) COMPONENT DIAGRAM – **dijagram komponenti**, zasniva pre svega na elementima kao što su softverske komponente i relacije između njih (os, drajveri, servisi, aplikacije, podaci).



9) DEPLOYMENT DIAGRAM – **dijagram razmeštaja** se zasniva pre svega na elementima kao što su procesni čvorovi i relacije između njih sa komponentama (hardver, računari, i/o uređaji, mrežna oprema).

