

Objedinjeni jezik modelovanja

Objedinjeni jezik modelovanja ili UML (Unified Modeling Language) je grafički jezik za vizuelno predstavljanje, specifikaciju, konstrukciju i dokumentovanje elemenata softverskih sistema koji je postavljen kao standard od OMG-a (Object Modeling Group).

UML nije programski jezik, on se više može smatrati grafičkim jezikom. UML se može primeniti i u drugim disciplinama osim razvoja softvera. UML predstavlja kolekciju najboljih uspešnih inženjerskih metoda kojima je, iskustveno dokazano, pojednostavljeno modelovanje velikih i složenih sistema.

UML

Postoji sedam ciljeva kojima UML kao jezik teži:

1. Pružiti korisniku brz jezik za vizuelno modelovanje kojim će moći u relativno kratkom vremenu napraviti i razmenjivati modele sa određenim značenjem.
2. Pružiti korisniku mogućnost proširenja i stvaranja specijalizovanih delova.
3. Biti nezavisan od programskih jezika i razvojnih procesa
4. Pružiti formalne osnove za razumevanje jezika za modelovanje

UML

5. Podsticanje rasta i razvoja objektno orijentisanih programskih jezika
6. Podrška visoko pozicioniranih razvojnih pojmova kao što su saradnja, okvirni rad, uzorci i komponente
7. Integrisanje i nadopunjavanje praktičnim iskustvom.

UML je razvijen sa ciljem da pojednostavi veliki broj objektno orijentisanih razvojnih metoda.

Osnovni elementi modela

Osnovni pojmovi modelovanja:

Entitet – jedinica posmatranja (proces, pojava, sistem, biće, predmet...)

Koncept – pojam ili misao koja se dodeljuje entitetima (stvarnim ili apstraktnim)

Domen problema – oblast interesovanja i okvir za definisanje skupa objekata posmatranog problema

Osnovni elementi modela

1. Objekat

Objekat opisuje pojedinačni entitet sa definisanom ulogom u domenu problema koji je određen svojim stanjem, ponašanjem i identitetom.

Stanje objekta obuhvata sve osobine posmatranog objekta i sve vrednosti tih osobina (promenljivih, parametara) kao i veze sa drugim objektima. Nad svakim objektom se može izvršiti skup operacija koje mogu menjati stanje objekta. Redosled izvršavanja operacija je bitan jer određuje stanje objekta, tj. utiče na ponašanje objekta.

Osnovni elementi modela

Ponašanje objekta zavisi stanja objekta i operacije koja se nad njim vrši. Svaka osobina objekta je predstavljena nekom vrednošću koja može biti jednostavna ili složena (objekat). Ako se zadata vrednost osobine ne menja tokom vremena onda je ona statička. U suprotnom osobina ima dinamičku vrednost. Osobine objekta ne obezbeđuju jedinstvenu identifikaciju objekta.

Identitet objekta jednoznačno određuje objekat. Identitet objekta odražava njegovo postojanje i ostaje nepromenjen bez obzira na promene stanja objekta. Identitet služi i za povezivanje objekata.

Osnovni elementi modela

Struktura objekta je skup svih njegovih atributa i operacija. Objekat se može posmatrati i kao pojava nekog koncepta. Objekat može predstavljati različite koncepte jer može imati različite uloge u sistemu.

Primer: Koncept Student. Svaki student fakulteta predstavlja konkretnu pojavu koncepta Student. Student doktorskih studija u isto vreme može biti i asistent, što znači da on istovremeno predstavlja i koncept Asistent. Za svakog studenta postoje podaci o njegovom imenu i prezimenu, broju indeksa, godini upisa, smeru, godini studija, broju položenih ispita i prosek ocena. Stanje objekta čine vrednosti za sve prethodno navedene podatke.

Osnovni elementi modela

Konkretan objekat koncepta student može biti student sa imenom Petar Petrović sa brojem indeksa 1234 koji je upisao fakultet 2009-te godine na smeru dizajner medija u obrazovanju treća godina, koji je položio 18 ispita sa prosekom ocena 7.58. Statičke vrednosti su npr. Godina upisa i broj indeksa, a dinamičke vrednosti su godina studija, broj položenih ispita i prosek. U ovom primeru smer može biti takođe objekat sa osobinama ime smera i nastavni plan smera. Navedene vrednosti određuju stanje objekta. Operacija nad ovim objektom može biti npr. upis godine. Da bi upis mogao da bude izvršen potrebno je da bude ispunjen uslov u vidu broja položenih ispita.

Osnovni elementi modela

Ponašanje sistema se posmatra kroz interakciju objekata koji postoje u tom sistemu. Stanje sistema predstavlja skup objekata sistema, njihovih stanja i njihovih međusobnih veza u određenom trenutku vremena. Veza između dva objekta nosi informaciju koju objekti imaju jedan o drugom recimo o operacijama koje objekti mogu da izvrše međusobno. Vezama se opisuju zakonitosti i ograničenja koja važe između objekata koji se nalaze u sistemu.

Objekat koji predstavlja profesora sa imenom i prezimenom Milan Milanović i zvanjem docenta može biti povezan sa studentom Petrom Petrovićem na način da Petar Petrović sluša predavanja kod profesora Milana Milanovića.

Osnovni elementi modela

Objekat koji je učesnik u vezi može da ima ulogu klijenta, servera ili agenta. Uloga klijenta podrazumeva da objekat može da zahteva izvršavanje operacija drugih objekata a da pritom nijedan objekat ne može da zahteva izvršavanje operacija klijenta. Server ne može da zahteva izvršavanje operacija drugih objekata već drugi objekti mogu da traže izvršavanje njegovih operacija. Agent može da zahteva od drugih objekata da vrše operacije ali i drugi objekti mogu da zahtevaju od njega da izvrši svoje operacije.

Osnovni elementi modela

2. Klasa

Klasa je pojam iz objektno orijentisanog programiranja i ona predstavlja skup objekata koji imaju zajedničke atribute, operacije, semantiku i veze sa drugim objektima. Objekat je konkretna pojava posmatrane klase i ima određenu ulogu u sistemu.

Primer: Klasa student je definisana podacima o imenu i prezimenu, broju indeksa, godini upisa, smeru, godini studija, broju položenih ispita i proseku ocena. Operacija ove klase može biti upis godine. Objekti koji predstavljaju konkretne studente su podaci tipa Student sa zadatim vrednostima za sve prethodno navedene atribute.

Osnovni elementi modela

Primer: Skup svih profesora nekog fakulteta je predstavljen sa skupom objekata klase Profesor koja ima attribute ime i prezime i zvanje. Između klase Profesor i klase Student postoji veza kojom se opisuje da studenti slušaju predavanja kod profesora.

Osim klasa postoje i metaklase, parametrizovane klase i potklase. Metaklasa je klasa čiji su objekti takođe klase. Parametrizovana klasa je uzorak za druge klase i konkretizuje se zamenom svojih parametara konkretnim klasama, objektima ili operacijama. Potklasa je klasa koja nasleđuje drugu klasu i dodaje neke svoje attribute i operacije.

Osnovni elementi modela

Klase imaju svoj spoljašnji i unutrašnji pogled. Spoljašnji pogled skriva strukturu klase i implementaciju operacija. Spoljašnji pogled prikazuje samo deklaracije operacija i atributa klase. Unutrašnji pogled daje uvid u implementaciju klase tj. svih operacija definisanih u spoljašnjem pogledu klase. Spoljašnji pogled klase može biti podeljen u tri dela i to:

- Javni
- Zaštićeni
- Privatni

Osnovni elementi modela

Javni deo predstavlja skup deklaracija koje su dostupne objektima drugih klasa. Zaštićeni deo sadrži deklaracije koje su dostupne samoj klasi i njenim podklasama. Deklaracije iz privatnog dela su dostupne samo toj klasi.

Izmena stanja objekta posmatrane klase ili preuzimanje vrednosti atributa posmatranog objekta se vrši pomoću operacija iz klasa. Jedino operacije pridružene nekoj klasi objekata mogu da pristupe osobinama objekata radi ažuriranja stanja ili preuzimanja vrednosti atributa.

Osnovni elementi modela

Objektima se pristupa preko skupa operacija bez poznavanja načina predstavljanja osobina objekata i načina implementacije posmatranih operacija. Razdvajanjem javnog i privatnog dela klase se uvodi **princip skrivanja informacija** koji podrazumeva da je način implementacije objekta skriven od drugih objekata u sistemu.

Na ovaj način je obezbeđeno da objekat jedne klase ne može da vidi implementacioni deo objekta druge klase, već ga samo može koristiti pozivajući ga pomoću poruka. **Poruka** predstavlja zahtev upućen objektu klase, da izvrši neku od svojih operacija. Objekti međusobno komuniciraju putem poruka.

Osnovni elementi modela

Osnovna karakteristika komunikacije među objektima je da objekat, koji prosleđuje poruku, ne mora posedovati informaciju o klasi objekta kome je poruka prosleđena. Iz tog razloga se ista operacija može implementirati na različite načine u različitim klasama pomoću različitih metoda. Ovaj koncept se naziva **polimorfizam**.

Prema načinu manipulisanja osobinama objekata operacije se mogu podeliti na:

- modifikatore,
- selektore,
- iteratore,
- konstruktore i
- destruktore

Osnovni elementi modela

Modifikatori su operacije koje menjaju osobinu objekta. Selektori su operacije koje pristupaju atributima objekta ali ih ne menjaju. Iteratori su operacije koje dozvoljavaju da sve osobine objekta budu dostupne ali u nekom dobro definisanom redosledu. Konstruktor je operacija koja kreira objekat, a može i da inicijalizuje njegovo stanje. Destruktor oslobađa memorijski prostor koji zauzima posmatrani objekat, tj. briše ga.

Objekti u sistemu se dele na aktivne ili pasivne objekte. **Aktivni objekti** imaju svoj tok kontrolne za razliku od **pasivnih objekata**.

Osnovni elementi modela

Aktivni objekti mogu da izvode svoje aktivnosti bez potrebe za akcijom od strane nekog drugog objekta nad njim. U slučaju kada sistem sadrži višestruke tokove kontrole, tada postoji više aktivnih objekata. Sa druge strane, pasivni objekti menjaju svoje stanje samo kada neki drugi objekti izvršavaju operacije nad njima.

Aktivni objekat predstavlja koren kontrole u programu. U okviru jednog toka kontrole se sekvencijalno izvršava niz akcija. U jednom procesu se više tokova kontrole može paralelno izvršavati. Tako je obezbeđeno da vreme CPU (Central Processing Unit) bude podeljeno i upotrebljeno za izvršavanje više različitih zadataka odjednom.

Osnovni elementi modela

Klase obično ne postoje izolovano, već su u interakciji sa drugim klasama u sistemu. Veze između klasa omogućavaju da povezane klase dele neke zajedničke osobine (nasleđivanje, agregacija) ili neku semantičku vezu (asocijacija). Veze između klasa su:

- Asocijacija
- Nasleđivanje
- Agregacija
- Zavisnost

Osnovni elementi modela

Asocijacija je veza koja označava semantičku zavisnost između klasa. Specifikacija ove veze podrazumeva navođenje imena veze, klasa koje učestvuju u vezi, uloge posmatranih klasa i kardinalnost preslikavanja. Kardinalnost preslikavanja predstavlja odnos pojavljivanja objekata klase u datoj vezi. Obično se koriste sledeće tri vrste kardinalnosti preslikavanja:

- Jedan prema jedan
- Jedan prema više
- Više prema više

Osnovni elementi modela

Primer: Veza asocijacije je veza između klasa Student i Profesor kojom se predstavlja veza da student sluša predavanja kod profesora, kao i da profesor drži predavanja studentima. Kako jedan profesor ima više studenata na predavanju a i studenti takođe idu kod više profesora na predavanja, ovo je primer veze više prema više. U ovoj vezi profesor ima ulogu predavača.

Sa druge strane, za izvođenje nastave iz jednog predmeta je obično zadužen samo jedan profesor, a jedan profesor može držati predavanja iz više predmeta. To znači da je veza između klasa Predmet i Profesor kardinalnosti jedan prema više.

Osnovni elementi modela

Nasleđivanje je veza u kojoj jedna klasa deli strukturu ili ponašanje definisano u jednoj ili više drugih klasa (jednostruko ili višestruko nasleđivanje). Klasa koja nasleđuje je podklasa, a klasa od koje se nasleđuje je superklasa ili nadređena klasa. Podklasa može povećavati ili ograničavati strukturu ili ponašanje nadređene klase.

Ne moraju sve klase imati konkretne objekte da bi učestvovala u nasleđivanju. Klase koje imaju objekte nazivamo **konkretnim klasama** a klase koje nemaju objekte **apstraktnim klasama**.

Osnovni elementi modela

Klasa koja ima najopštiju strukturu i ponašanje u stablu nasleđivanja se naziva osnovnom klasom. Vezom nasleđivanje se modeluje generalizacija odnosno specijalizacija.

Primer: Klasa vozilo se može specijalizovati na dve potklase PutničkoVozilo i TeretnoVozilo. Za potklasu Putničko vozilo se može definisati dodatni atribut maksPutnika. Za teretno vozilo se može definisati dodatni atribut maksTežine ili maksDimenzije.

Osnovni elementi modela

Agregacija predstavlja vezu između celine i dela. Vrsta agregacije **kompozicija** znači da objekat koji predstavlja deo nekog drugog objekta ne postoji nezavisno od celine čiji je deo. Životni vek objekta koji predstavlja deo zavisi od životnog veka objekta koji predstavlja celinu. Postoji i manje direktna vrsta agregacije koja predstavlja sadržavanje po referenci i ovde životni vek celine i dela nije u tako velikoj meri povezan. To znači da je kreiranje i uništavanje objekata ovih klasa nezavisno. U ovom slučaju deo se može koristiti u više celina i može postojati nezavisno od objekta koji predstavlja celinu.

Osnovni elementi modela

Primer: Ako posmatramo klase Fakultet i Smer, možemo reći da na jednom fakultetu postoji više smerova. Gašenjem fakulteta se automatski gase i svi smerovi na njemu što znači da je veza između klasa Fakultet i Smer veza kompozicije.

Primer: Grupa za vežbe se može modelirati kao klasa koja predstavlja agregaciju po referenci skupa objekata klase Student. Jedan student može učestvovati u više grupa za vežbe. Postojanje objekta klase Student je nezavisno od postojanja objekta klase GrupaZaVežbe.

Osnovni elementi modela

Veza zavisnosti opisuje zavisnost između klasa i drugih elemenata modela. Zavisnost ima značenje da prisustvo jednog elementa modela uslovljava i prisustvo elemenata modela sa kojima je u vezi. Veza zavisnosti se koristi kada jedna klasa ima referencu na drugu klasu prilikom izvršavanja svoje operacije.

Primer: Klasa prodavac obično uslovljava postojanje i klase kupac ili artikl kako bi objekat klase prodavac mogao da vrši svoju funkciju.

Osnovni elementi modela

3. Interfejs

Interfejs definiše skup operacija koje definišu neko ponašanje. Jedan interfejs može biti implementiran od strane više klasa a takođe i jedna klasa može implementirati više različitih interfejsa. Interfejs definiše spoljašnje (vidljive) operacije neke klase i ne sadrži implementaciju, attribute i stanje. Interfejs možemo posmatrati kao apstraktnu klasu bez atributa za koju se može definisati veza nasleđivaja.

Osnovni elementi modela

4. Paket

Modeli mogu biti organizovani u više logičkih celina međusobno povezanih vezama zavisnosti nasleđivanja ili kompozicije, kako bi se pojednostavio sistem. Te logičke celine nazivamo paketima i oni predstavljaju podskup modela koji se sastoji od elemenata modela i veza između njih. Paket određuje oblast definisanosti klase sistema.

5. Podsistem

Podsistem je izdvojena celina sistema čiji elementi imaju zajedničko ponašanje.

Proširenje osnovnih elemenata modela

UML obezbeđuje mehanizme koji omogućavaju definisanje novih elemenata modela na osnovu već postojećih. Proširenje osnovnih elemenata modela se vrši pomoću stereotipa, ograničenja i označene vrednosti.

Stereotip definiše novi element na osnovu definicije postojećeg elementa modela dodavanjem odgovarajuće semantike, čime se proširuje semantika osnovnog elementa modela. Novi element modela se označava imenom (stereotipom) koji se navodi između znakova « i ». Na ovaj način je moguće definisati nove elemente modela bez menjanja notacije, odnosno samog jezika modelovanja.

Proširenje osnovnih elemenata modela

Ograničenje je mehanizam koji omogućava definisanje ograničenja nad elementom modela, čime se menja njegovo značenje ili način korišćenja. Ograničenjem se određuju uslovi koji moraju biti zadovoljeni. Postoji skup predefinisanih ograničenja koji može biti proširen od strane korisnika. Ograničenje se navodi u okviru vitičastih zagrada.

Označene vrednosti omogućavaju definisanje različitih osobina elemenata modela. Osobina se zadaje preko svog imena i vrednosti i može biti definisana za klase, attribute, operacije, vezu asocijacije ili bilo koji element modela. Ovako se omogućava dodavanje bilo koje informacije u model.

Kontrolna pitanja

57. Šta je UML?
58. Koje uloge može da ima objekat koji je učesnik u vezi?
59. Šta je klasa?
60. Na koje delove je podeljen spoljašnji pogled klase?
61. Kako se mogu podeliti operacije prema načinu manipulisanja sa objektima?
62. Šta su aktivni objekti?
63. Koje veze između klasa postoje?
64. Opišite vezu asocijacije.
65. Opišite vezu nasleđivanja.
66. Opišite vezu agregacije.