



VEZA ZAVISNOSTI INSTANCE

Munir Šabanović¹, Momčilo Vujičić²

Rezime: *Objektno orijentisani jezici odlikuju se komunikacijom između objekata odnosno instanci klasa. U vezi zavisnosti instance opisuje se veza između specifikacionog elementa instance i same instance. Takođe, ovde se prezerentira različit pogled na objekat.*

Ključne reči: *Instanca, klasa.*

RELATION DEPENDENCY INSTANCE

Summary: *The main characteristic of object-oriented programming languages is communication between objects, more exactly classes' level. Related to the level's dependency, a link between the specified level element and the level itself is described. Different point of view to the object is presented here as well.*

Key words: *Instanca, class.*

1. UVOD

Centralno mesto u objektnom programiranju zauzima klasa, odnosno instanca klase - objekat. Klasama se definišu gradivni elementi kao i međusobne veze modela koji se opisuje. Među relevantnim stručnjacima iz objektnog programiranja ne postoji saglasnost kada je u pitanju pogled na objekat pa se zato pribeglo empirijskoj definiciji objekta. Po mišljenju relevantnih autora (G. Booch) svaki objekat se definiše pomoću identiteta, stanja i ponašanja.

Identitet objekta je nešto po čemu se objekti iste ili različitih klasa razlikuju jedan od drugog. Identitet je unikatan tako da dva objekta u istom memorijskom prostoru ne mogu da imaju isti identitet iako mogu da se nalaze u istom stanju i da imaju isto ponašanje.

Stanje objekta se opisuje pomoću konkretnih vrednosti podataka članova klase i predstavlja rezultat operacija koje su vršene nad objektom u prošlosti i određuje ponašanje objekta u sadašnjosti i budućnosti. Na primer kod steka se ne može izvršiti operacija čitanja ako nema elemenata u steku ali se može izvršiti operacija upisivanja.

Ponašanje objekta se opisuje pomoću operacija, a realizacija operacije nosi naziv "metoda".

¹ Munir Šabanović, dipl. ing. elektrotehnike, asistent na Fakultetu za informatiku i informacione tehnologije, Internacionalnog univerziteta u Novom Pazaru, e-mail: munirsabanovic@yahoo.com

² Dr Momčilo Vujičić, vanredni profesor, Tehnički fakultet, Svetog Save 65, Čačak, e-mail: vujicic@tfe.kg.ac.yu, vujicic_momcilo@yahoo.com

Klasa može imati jednu operaciju, više njih ili nijednu operaciju. Na primer, u biblioteci za rad sa prozorima, kao što je slučaj kod Javinog paketa awt, svi objekti klase Rectangle mogu se pomerati, može im se menjati veličina ili im se mogu ispitivati osobine. Obično (ali ne i uvek), poziv operacije nad objektom menja njegove podatke ili stanje. Grafički u UML-u operacije se navode u odeljku ispod atributa slika 1.2. Ponašanje objekta, kao reakcija na aktiviranje neke operacije (pobudu, poruku) zavisi od tri faktora, a to su:

- sama pobuda
- stanje datog objekta
- stanje drugih objekata, iste ili druge klase, koji nisu objekti-članovi, ali ostvaruju uticaj u sklopu pobude. Znači može da konsultuje neku metodu ili stanje nekog trećeg objekta.

Takođe je neophodno napomenuti da objekat može da bude aktivni i pasivni. Ako ima ugrađeno ponašanje radi se o aktivnom objektu za razliku od sloga koji je pasivan jer nema metode koje su ugrađene u slogu. Prema tome ponašanje objekta određeno je operacijama, pri čemu konkretni obrazac ponašanja zavisi kako od same operacije tako i od tekućeg (zatečenog) stanja. Aktiviranje operacije vrši se tzv. porukom.

2. VEZA ZAVISNOSTI INSTANCE

Instanca je konkretan primerak klase sa sopstvenim identitetom, stanjem i ponašanjem i predstavlja sinonim za objekat. Objekat može da se posmatra u tri domena i to na konceptualnom nivou, domenu modela i domenu realizacije. Na konceptualnom nivou najčešće se koristi drugi termin tj. entitet, dok se u druga dva domena koristi termin objekat iako u domenu modela objekat zapravo predstavlja uprošćeni model entiteta iz konceptualnog nivoa dok u domenu realizacije objekat zauzima konkretan memorijski prostor.

Entitet ima mnoštvo osobina, ali se uzimaju u obzir bitne osobine za dati domen problema i one nose naziv atributi, dok se druge misaono odbacuju za razliku od modela gde se nebitne osobine potpuno odbacuju. Takođe, atribut poseduje ime i predstavlja određenu osobinu sredstva koji se modeluje kao na primer ime studenta, broj indeksa itd., i koja je zajednička svim objektima klase. Na primer svaki zid ima visinu, širinu i debljinu; kupce možemo modelovati da svaki ima ime, adresu, telefonski broj i datum rođenja. Atribut je na taj način, apstrakcija vrste podataka ili stanja koje objekat klase može sadržati [2].

Jedna od bitnih osobina atributa je njegova semantička nesamostalnost, na primer atribut ime sam po sebi nema nikakvo značenje a dobija ga ako se dovede u vezu sa nekom klasom entiteta iz datog domena problema (klasa čovek—čovek ima ime, klasa pas-konkretan pas ima ime, auto-konkretno auto ima ime, klasa životinja). Domen atributa se definiše na nivou klase i isti je za sve objekte date klase. Atribut se može definisati na sledeći način:

Atribut predstavlja imenovanu deskriptivnu osobinu entiteta koja je bitna u datom domenu problema ili nekom od njegovih subdomena [1].

Atributi se posmatraju u domenu problema dok se u domenu realizacije koristi drugi naziv i to podatak-član. Atribut može istovremeno da bude i podatak član kao na primer koordinate x i y u dvodimenzionalnom koordinatnom sistemu, ali i ne moraju da reprezentuju istu stvar kao na primer kod semafora gde su boje crvena, žuta i zelena atributi dok podatak član u zavisnosti od skalarne vrednosti reprezentuje određenu boju.

Pored opisnih osobina u definisanju klase učestvuju i procedurene osobine i to one koje su bitne za dati domen problema. Na osnovu rečenog može se zaključiti da su entiteti realni, klasa entiteta već ima nivo apstraktnog.

Pojam entiteta definisan je međunarodnim standardom kao bilo koja konkretna (golf 3 plave boje koji dodirujemo-postoji, konkretan lav) ili apstraktna stvar (pas, auto, lav) ("thing") koja postoji, koja je postojala (Napoleon) ili je mogla postojati, uključujući i veze između ovih stvari.

Delovi instance klase koji su tipa klase nazivaju se objekti članovi. Podaci članovi i objekti članovi su promenljive objekta. Dakle objekti i njihovi objekti-članovi sa svojim članovima čine strukturu objekta.

Prema tome sastavni delovi objekta su podaci članovi, objekti članovi i operacije koje se mogu izvršiti nad objektom (tzv. funkcije-članice ili metode).

Dakle, U klasi se specifikuje forma i ponašanje za skup instanci sa istim osobinama. Instance imaju identitet i vrednosti koje su konsistentne sa specifikacijom u opisu objekata tj. klasom.

Drugim rečima, na različitim tačkama u vremenu tokom izvršavanja u sistemu, instance može biti identifikovana istim imenom iako stanje instance može biti promenjeno. U bilo kom trenutku instance imaju neko stanje izraženo u obliku vrednosti podatka-članova i referenci do drugih instanci. Identitet podatka je povezan sa njegovom dužinom trajanja. Osim toga identitet i stanje, za svaku instancu imaju opis koja ograničava moguća stanja koje instance može imati. Svrha modela je da opiše moguće vrednosti koje mogu da uzmu podaci članovi u instancama modela. Objekat je instance klase a link je instance od veze. Postoje bezimene instance koje egzistiraju u nekom prelaznom procesu.

Model opisuje moguće vrednosti objekta i njegovo ponašanje u napredovanju od vrednosti do vrednosti tokom izvršavanja. Stanje opisuju vrednosti podataka članova objekata kao i njihovih objekata članova u modelu.

Direktna instance. Objekat je direktna instance date klase ako pripada toj klasi i ne pripada nijednoj klasi koja je nasleđuje.

Primer:

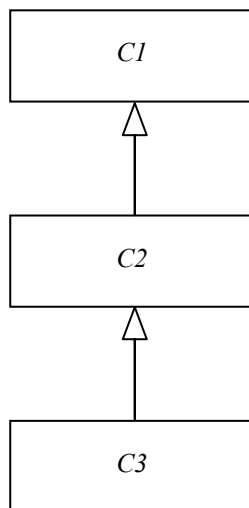
Recimo da imamo klasu C1. Klasa C2 nasleđuje klasu C1. Klasa C3 nasleđuje klasu C2. Na sledećoj slici je grafički prikazana hijerarhija klasa C1, C2 i C3.

Definišemo li promenljive (objekte) I1, I2 i I3:

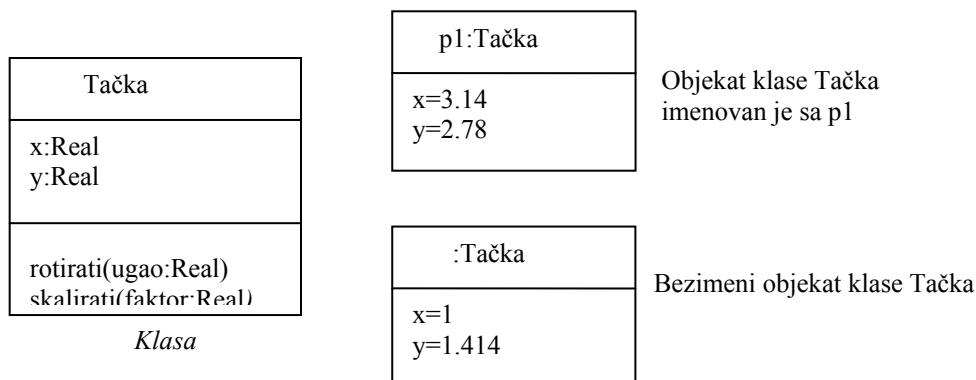
- C1 I1; /* I1 je objekat (tipa) klase C1 i samim tim je instance C1 */
- C2 I2; /* I2 je objekat (tipa) klase C2 i samim tim je instance C2 */
- C3 I3; /* I3 je objekat (tipa) klase C3 i samim tim je instance C3 */
- I1 je direktna instance klase C1 jer je instance klase C1, a neke koja je nasledila C1 (u ovom slučaju C2 i C3).
- I2 i I3 su indirektno instance klase C1 jer su instance klase koje nasleđuju C1.

Slično i za ostale:

- I2 je direktna instance klase C2.
- I3 je indirektna instance klase C2.
- I3 je direktna instance klase C3.

**Slika 1**

Na slici 2 prikazana je klasa Tačka koja opisuje tačku u dvodimenzionalnom koordinatnom sistemu i dva objekta klase Tačka. Jedan objekat je imenovan sa p1 a drugi je bezimeni. Oba objekta imaju konkretne vrednosti podataka članova x i y.



Instance (dva objekta) su oličene stanjima. Nije potrebno prikazati fiksiran deo za operacije za svaki objekat, već se operacije koriste za sve instance.

Slika 2

3. ZAKLJUČAK

Ovde sam proučio vezu instance koja se bavi instanciranjem objekata i to kako direktnih tako i indirektnih.

4. LITERATURA

- [1] Dušan Malbaški: *Objekti i objektno programiranje kroz programske jezike C++ i pascal*, 2006.
- [2] Meyer B.: *Object-Oriented Software Construction*, Prentice Hall, 1988
- [3] James Rumbaugh, Ivar Jacobson, Grady Booch: *The Unified Modeling Language, Reference Manual*;
- [4] Ivana Stanojević, Dušan Surla: *UML, Uvod u objedinjeni jezik modeliranja*
- [5] *UML Distilled Second Edition A Brief Guide to the Standard Object Modeling Language*
- [6] Kraus L.: *Programski jezik C++ sa rešenim zadacima*, Mikroknjiga, Beograd, 1994.
- [7] Branko Milosavljević, Milan Vidaković: *Java i internet programiranje*, Novi Sad 2002.
- [8] <http://www.parlezuml.com/tutorials/umlforjava>
- [9] Dušan Malbaški, *Internet programiranje-deo 1: objekti i objektno programiranje kroz programske jezik Java*, Tehnički fakultet Mihajlo Pupin Zrenjanin, 2007.