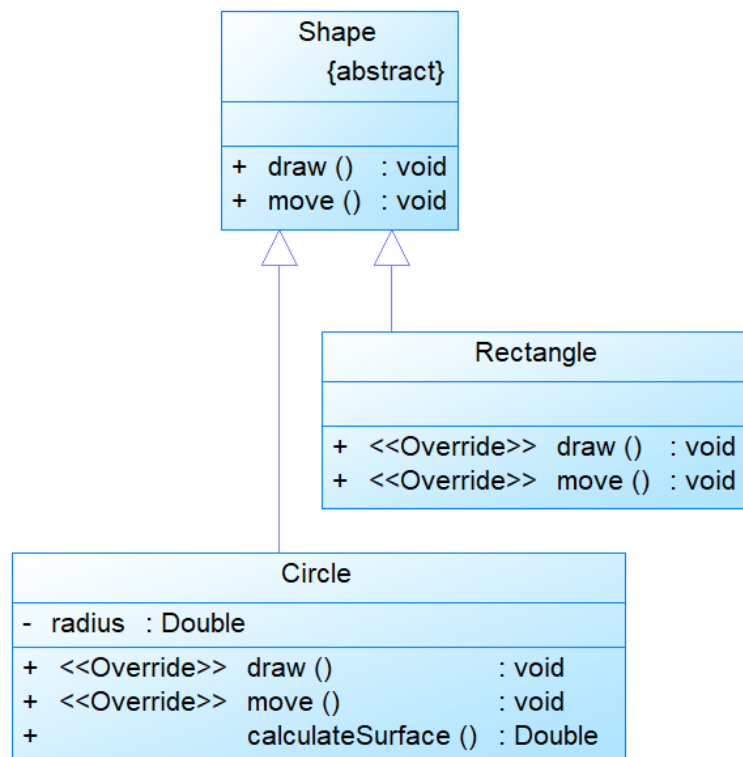


Class Diagram – dijagram klasa

1. Osnovni koncepti

Dijagrami klasa su strukturalni dijagrami u UML-u koji prikazuju skup klasa, interfejsa i njihovih veza. Strukturalni dijagrami su statički dijagrami koji opisuju nepromenljive logičke strukture u softveru. Dakle, akcenat je na strukturi sistema, a ne na njegovom ponašanju. Dijagrami klasa mogu prikazivati ceo ili samo deo strukture sistema.

Većina alata podržava generisanje skeleta koda iz dijagrama klasa.



2. Elementi dijagrama klasa

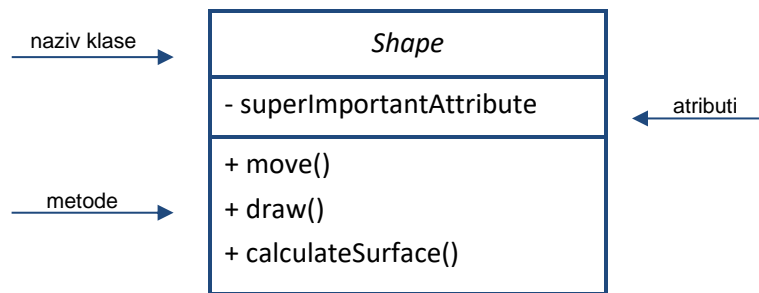
Elementi dijagrama klasa:

- Klase
- Interfejsi
- Paketi
- Objekti
- Veze: zavisnosti, generalizacije, asocijacije, realizacije

2.1. Klasa

Klasa je opis skupa objekata koji dele iste karakteristike, ograničenja i semantiku.

Simbol klase u UML-u je pravougaonik podeljen horizontalnim linijama u odeljke. Odeljci mogu biti prazni, što znači da atributi/metode nisu relevantni za dati dijagram. Apstraktna klasa se označava navođenjem **{abstract}** kod naziva klase ili se naziv klase napiše *italic*. Kardinalitet je osobina klase koja ograničava broj njenih instanci. Podrazumevani slučaj je proizvoljan broj instanci.

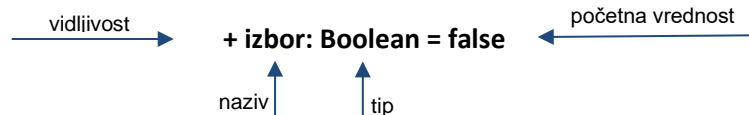


Klase mogu da opisuju apstrakcije:

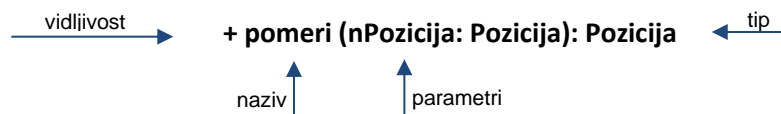
- Iz domena problema (za potrebe analize, kreiranja konceptualnog modela i slično)
- Iz domena rešenja (za potrebe implementacije)

2.1.1. Atributi i metode klase

Atributi su imenovana svojstva klase koja opisuju opsege vrednosti koje instance tog svojstva mogu sadržati.



Metode su servisi koji se mogu zahtevati od nekog objekta klase.








Vidljivost	Class	Package	Subclass (same pkg)	Subclass (diff pkg)	World
+ public	+	+	+	+	+
# protected	+	+	+	+	
~ package	+	+	+		
- private	+				

Tipovi podataka mogu biti:

- Primitivni: *int, bool, byte...*
- Enumeracije: definisan skup simboličkih imena
- Klase i interfejsi:
 - Ugrađene: *String, DateTime*
 - Kreirane: aktivni ili uvezani dijagram klasa

2.2. Veze zavisnosti, generalizacije, asocijacije i realizacije

Veze definišu saradnju klasa. Veze među strukturama sistema je mnogo lakše uočiti na dijagramu nego u kodu.

Zavisnost	Generalizacija	Asocijacija
		
		
		

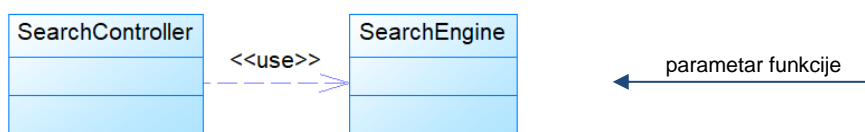
2.2.1. Veze zavisnosti

Veza zavisnosti se može povlačiti između svih elemenata UML modela. Povezuju se elementi modela kod kojih izmena nezavisnog elementa utiče na ponašanje zavisnog elementa (funkcionalnost ili implementacija jednog elementa modela zahteva prisustvo drugog elementa modela).

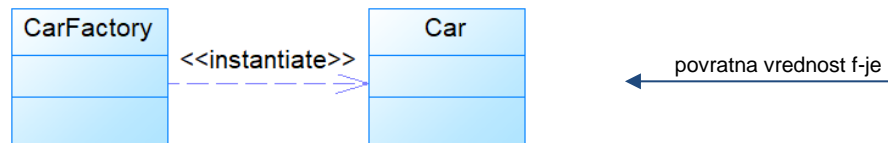
Element modela od koga polazi usmerena linija predstavlja zavisni element, dok element na drugom kraju veze predstavlja nezavisni element. Zavisni element koristi nezavisni element. Kao element modela na dijagramu klasa se može posmatrati klasa, interfejs ili paket.

Specijalan slučaj veze zavisnosti može biti predstavljen navođenjem odgovarajućeg stereotipa. Postoji niz stereotipa koji su definisani za vezu zavisnosti, a korisnik može definisati i svoje. Neki predefinisani stereotipovi su:

- *use* – funkcionalnosti ili implementacija jednog elementa modela zahteva prisustvo drugog elementa modela (npr. jedna klasa koristi objekat druge klase kao parametar svoje metode). Služi za naglašavanje osnovnog tipa veze zavisnosti u odnosu na sve ostale stereotipove



- *instantiate* – operacije nezavisne klase kreiraju instance zavisne klase



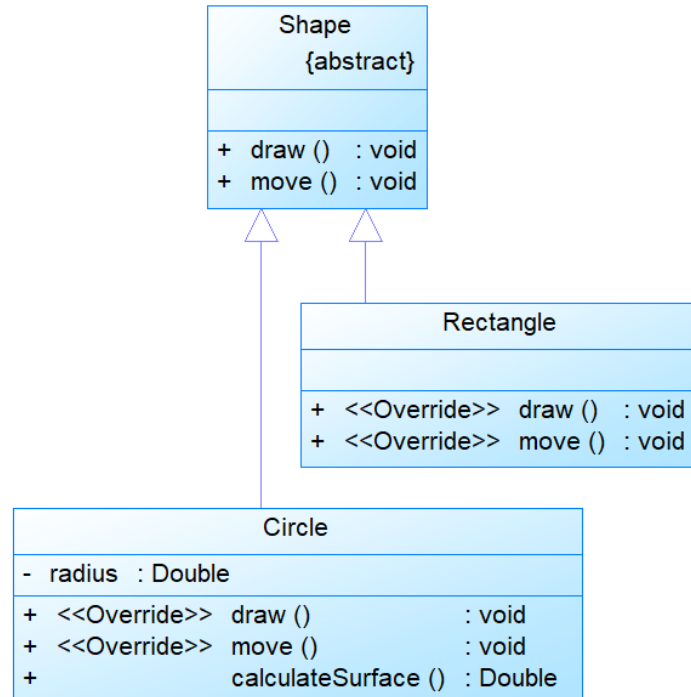
- *bind* – veza parametrizovane klase i klase koja daje konkretne vrednosti parametrima

Veze zavisnosti doprinose semantici modela, **ali ih treba koristiti selektivno** pošto ih ima puno.

2.2.2. Veze generalizacije

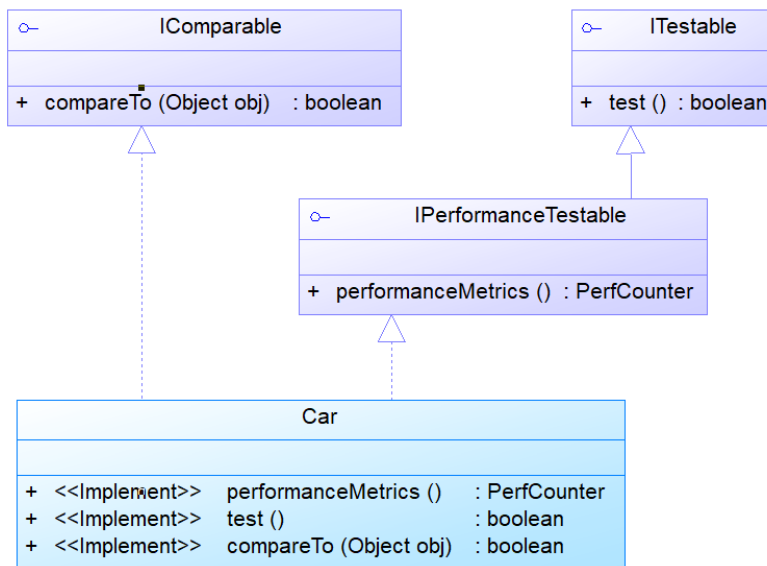
Veza generalizacije povezuje opštije elemente modela (superklasa, predak, roditelj) sa specijalizovanim (subklasa, naslednik, dete).

Generalizacija znači da se objekti naslednika mogu pojaviti gde god se očekuje objekat pretka. Naslednici nasleđuju osobine svojih roditelja (atribute i metode). Metoda naslednika koja ima isti potpis kao metoda pretka redefiniše metodu pretka. Redefinisanje metode omogućava njeno polimorfno ponašanje. Klasa koja ima samo jednog pretka koristi jednostruko nasleđivanje. Klasa koja ima više predaka koristi višestruko nasleđivanje.



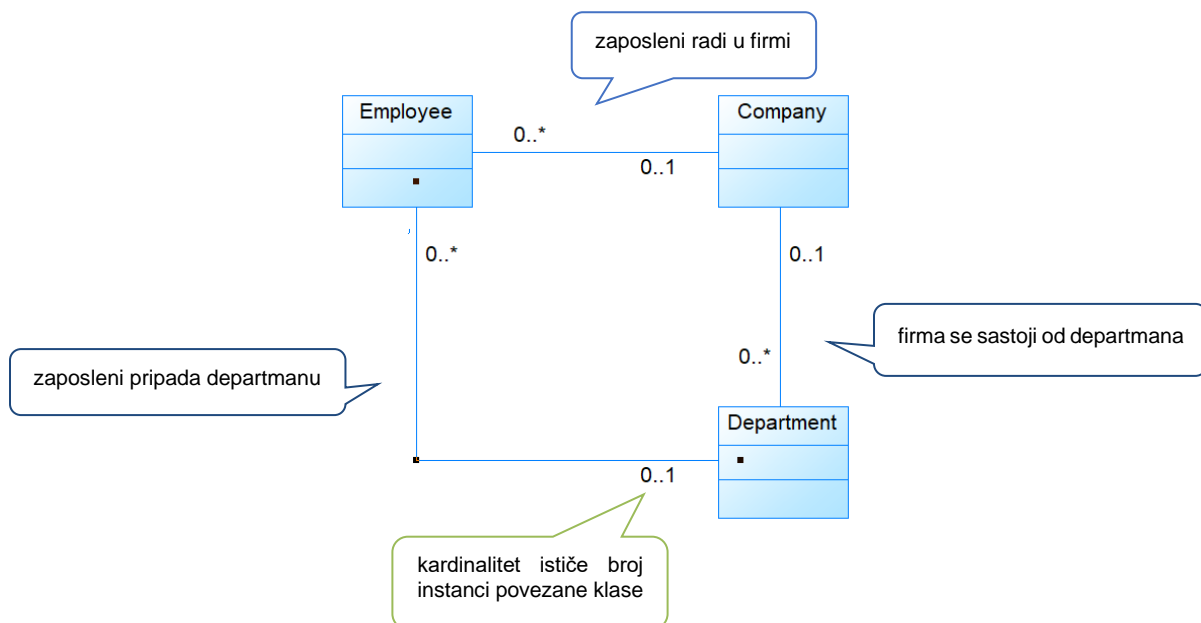
2.2.3. Veze realizacije

Veza realizacije se na dijagramu klasa koristi za označavanje veze između interfejsa i klase koja ga implementira.

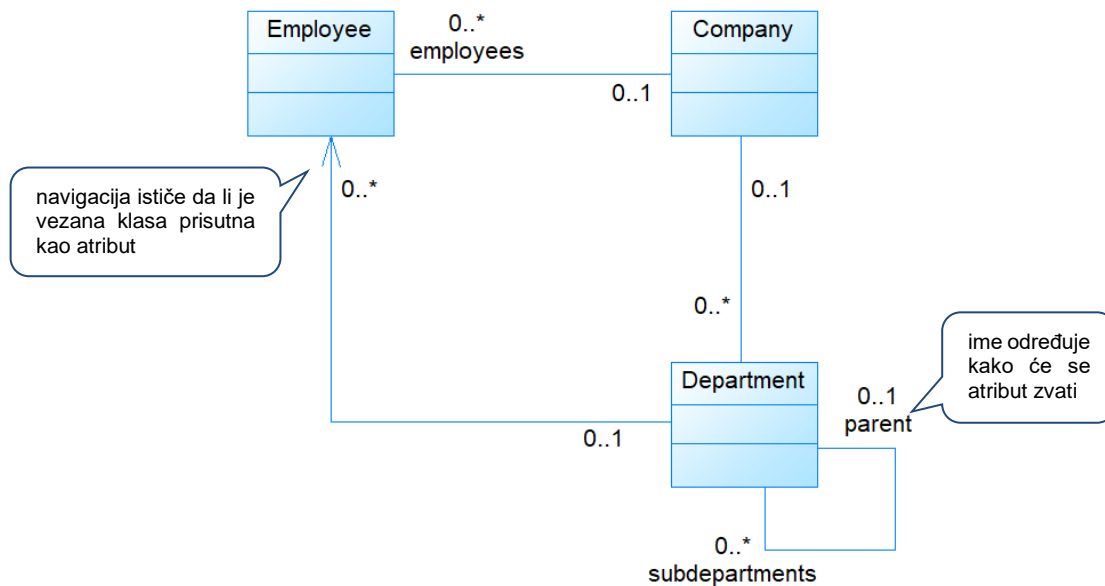


2.2.4. Veze asocijacije

Asocijacija je strukturna veza koja definiše odnos udruživanja i/ili posedovanja između instanci klasa. Preko asocijacije klase A sa klasom B instance klase A mogu doći do instanci klase B, što se obično implementira uvođenjem atributa (reference na B) u klasi A.



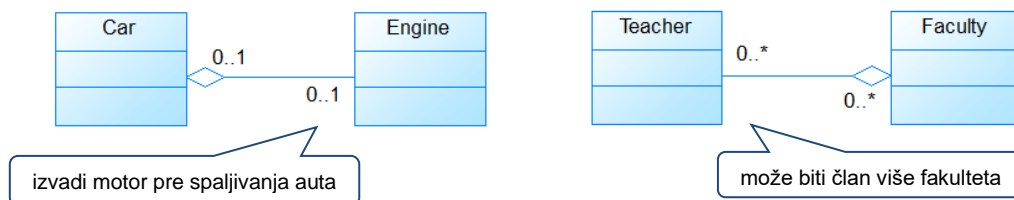
Ako se drugačije ne kaže, asocijacija predstavlja bidirekionalnu vezu (dvosmerna navigabilnost). Dozvoljena je i asocijacija sa samim sobom (postoje veze između objekata iste klase). Strelica na vezi asocijacije označava navigabilnost u naznačenom pravcu, a ukoliko je veza bez oznake navigabilnosti, onda se smatra da je navigabilna na obe strane.



Uobičajeno je da asocijacija povezuje dve klase (binarna asocijacija). Moguće je da asocijacija povezuje i više klasa (n-arna asocijacija), ali to je bolje modelovati preko klase asocijacije.

Postoji više vrsta asocijacija:

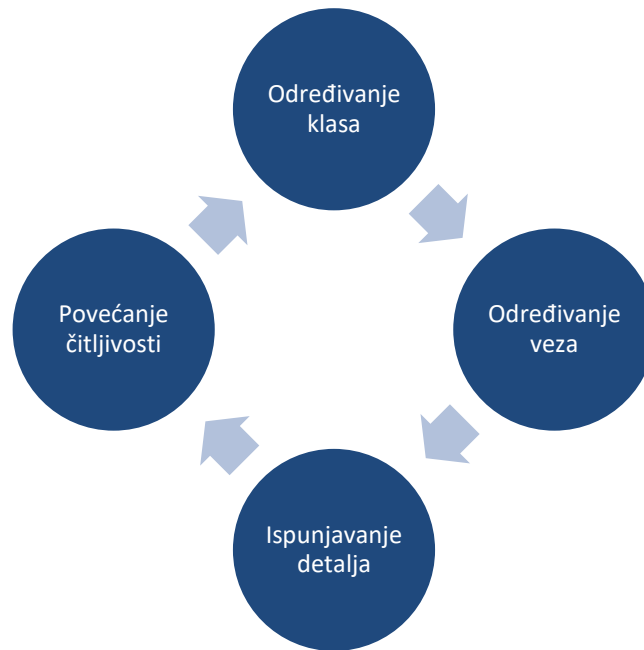
- Obična asocijacija je uopštena strukturna veza, koja se koristi za modelovanje udruživanja bilo koje dve klase.
- Agregacija je specijalna vrsta asocijacije koja modeluje odnos "celina-delovi", pri čemu su delovi samostalne klase čije instance mogu da "žive" i nezavisno od klase koja ih agregira. Deo u agregaciji može biti zajednički deo više celina.



- Kompozicija je specijalna vrsta agregacije kod koje su delovi slabi objekti koji ne mogu samostalno da postoje bez osnovne klase. Deo može pripadati samo jednoj celini.



3. Crtanje dijagrama



- Određivanje klasa – potrebno je uočiti bitne domenske objekte
- Određivanje veza – uočiti odnose među objektima, koje su njihove sličnosti, postoji li mogućnost apstrakcije
- Ispunjavanje detalja – atributi i metode klasa, detalji veza
- Povećanje čitljivosti – da li je moguće izvršiti paketiranje, reorganizaciju, bojenje, ubacivanje beleški

Zadatak 1. – koristeći *power designer* alat nacrtati dijagram klasa za proizvoljan sistem ili deo sistema (npr.sistem fakulteta)

Zadatak 2. – nacrtati dijagram klasa za deo projektnog zadatka