

Dijagrami

Sadržaj:

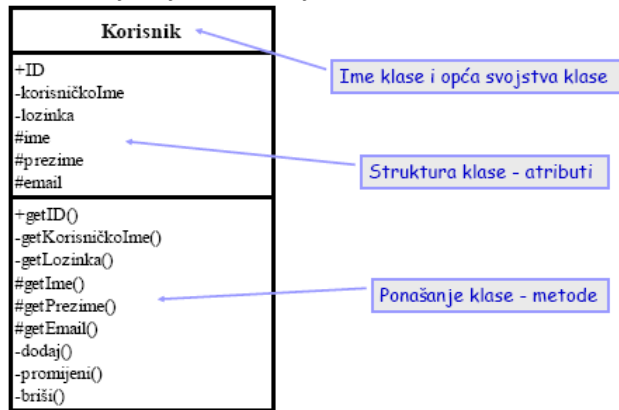
Dijagram klasa	2
Dijagram objekata.....	4
Dijagram stanja.....	5
Dijagram aktivnosti	6
Dijagram komponenti	8
Kolaboracijski dijagram / Dijagram međudjelovanja	10

Dijagram klasa

= statički pregled sustava pokazujući njegove klase i odnose među tim klasama

Razred / Klasa

- plan po kojem se stvara objekt
- svaki objekt je instanca jedne klase



Atribut - svojstvo klase. Naziv atributa se sastoji od jedne riječi napisane malim slovima. Ako se naziv atributa sastoji od više riječi, te riječi su spojene i svaka riječ osim prve počinje s velikim slovom.

Metoda je nešto što klasa može raditi ili što vi (ili druga klasa) možete raditi toj klasi. Za naziv metode vrijede pravila kao i za naziv atributa.

Atributi i metode mogu se označiti na sljedeći način:

Zaštićena – vidljiva je samo podklasama određene klase #

Privatna – nije vidljiva izvan klase -

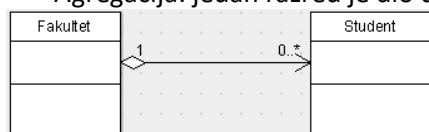
Javna – vidljiva je svima +

Veze opisuju odnose između pojedinaca (instanci) razreda

- **Mnogostrukost** veze određuje broj primjeraka jedne klase u odnosu na drugu klasu (1, 0..*, 1..*, 4..6)

- Vrste

- Pridruživanje: jednosmjerno ➡ dvosmjerno ➡
- Agregacija: jedan razred je dio drugog razreda



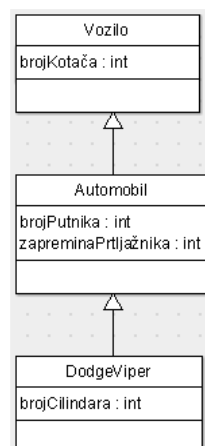
Student je dio fakulteta

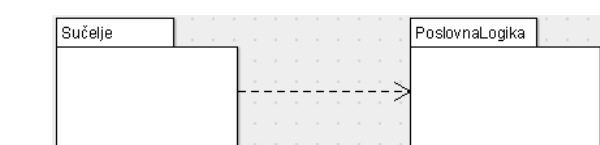
- Kompozicija: jedan razred je dio drugog razreda, ali kod uništavanja objekta uništavaju se i pojedinci razreda koji su dio tog objekta



Student je dio fakulteta, uništavanjem fakulteta će se uništiti i student

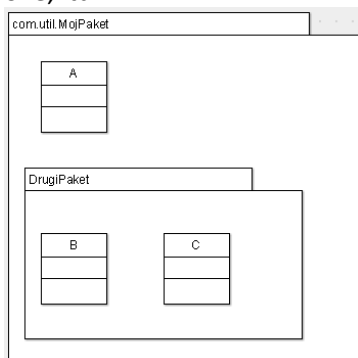
- Generalizacija - omogućuje stvaranje nadklase koja objedinjuje strukturu i ponašanje zajedničko za nekoliko klasa
- Specijalizacija – omogućuje stvaranje podklase koja predstavlja podklasu uvijek ima više ili jednak broj svojstava u odnosu na nadklasu
- **Ovisnost** jednog paketa ili razreda o drugom



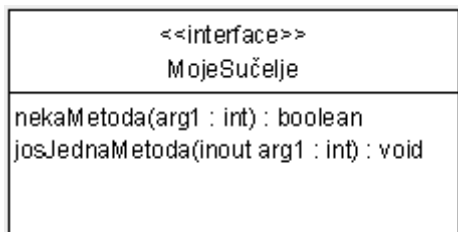


Sučelje ovisi o poslovnoj logici

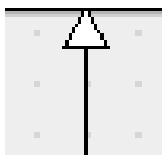
Paket - je skup različitih objekata. Može sadržavati druge pakete, objekte, razrede, komponente, UC-ove, itd.



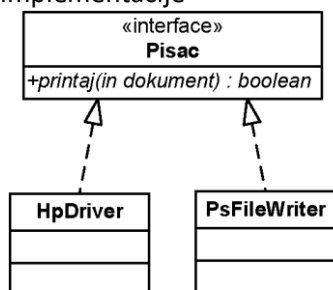
Sučelje - je razred, ali bez atributa i sve operacije imaju samo tijelo, bez implementacije



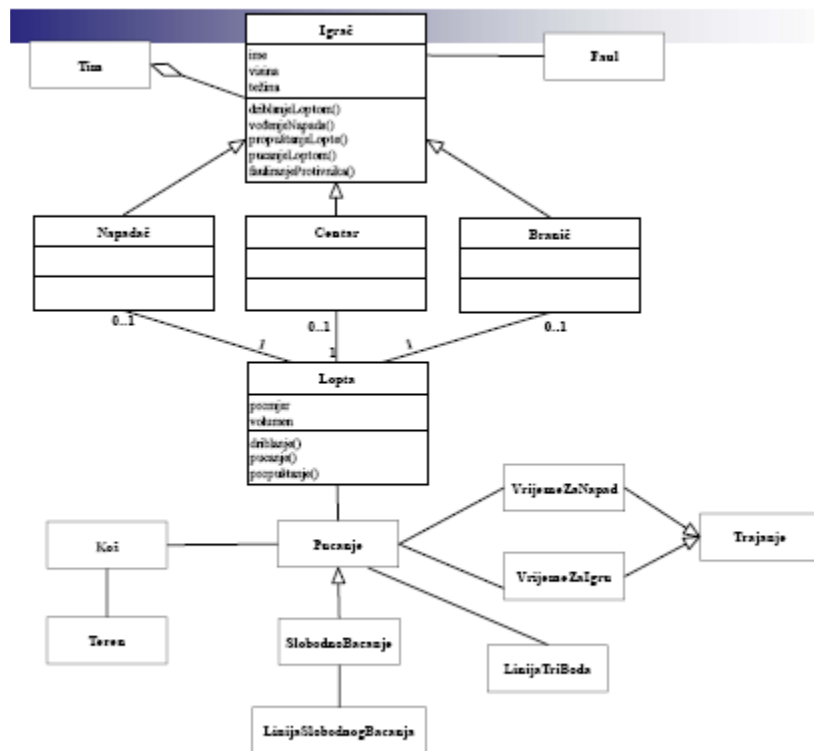
Nasljeđivanje - mehanizam stvaranja novih klasa koje nasljeđuju atribute, relacije i operacije definirane u njenim temeljnim klasama, mogu redefinirati atribute i dodati nove vlastite atribute te mogu imati svoju implementaciju operacija koje je naslijedila



Realizacija – slična nasljeđivanju samo što se ovdje nasljeđuju samo operacije s parametrima, a bez implementacije



Primjer:

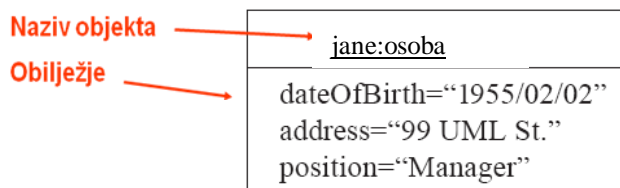


Dijagram objekata

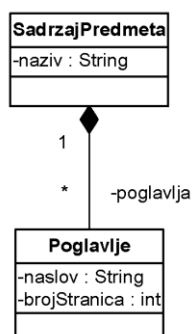
= grafički prikaz primjeraka uključujući objekte i pripadne im vrijednosti
 = opisuju stvarne podatke koji bi se mogli nalaziti u objektima u jednom vremenskom trenutku

Objekt

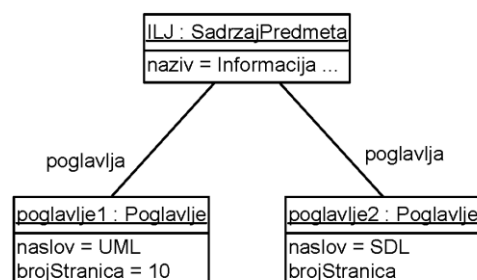
- entitet koji ima stanje (skup objekta) i definirani skup metoda koje djeluju na tom stanju
- Naziv objekta počinje malim slovom i prethodi dvotočki koja prethodi nazivu klase i cijeli naziv je podcrtan.



dijagram klasa



dijagram objekata

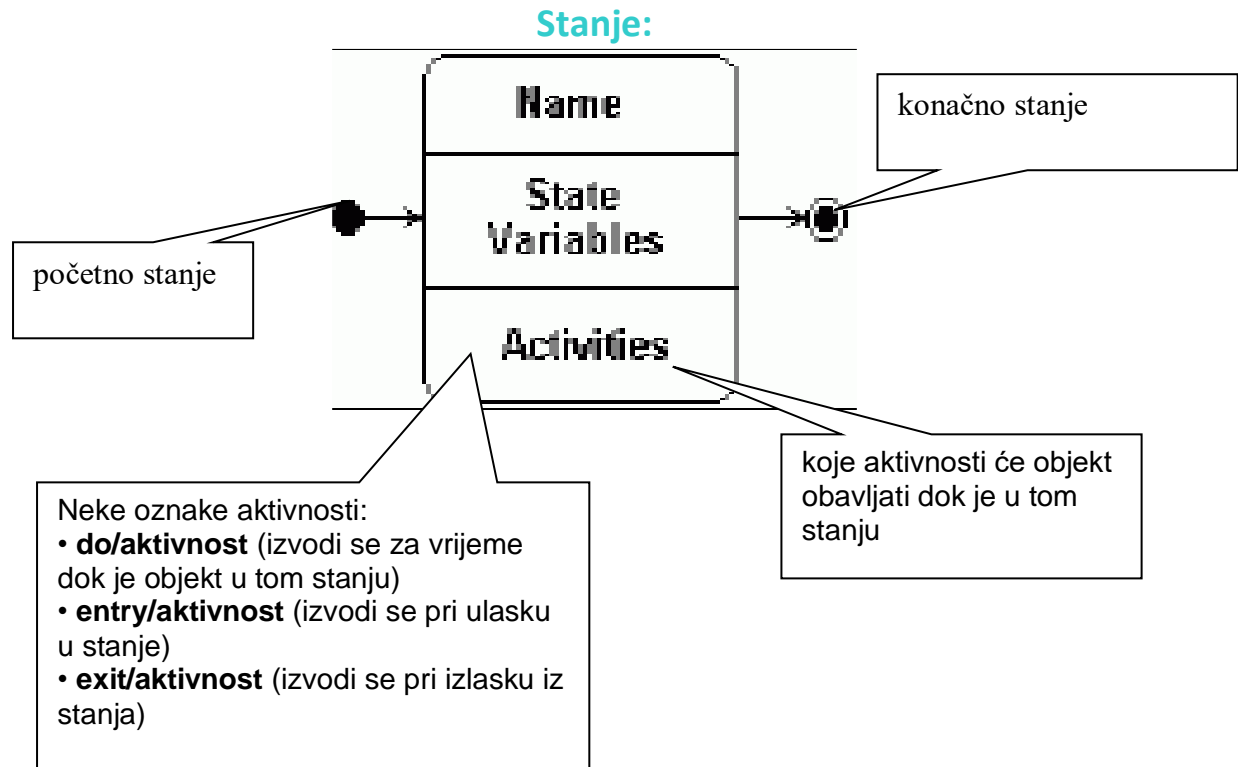


Dijagram stanja

= mehanizam za modeliranje promjena

= predstavlja stanja u kojima objekt može biti kao i prijelaze između tih stanja te pokazuje početnu i krajnju točku u nizu promjena stanja

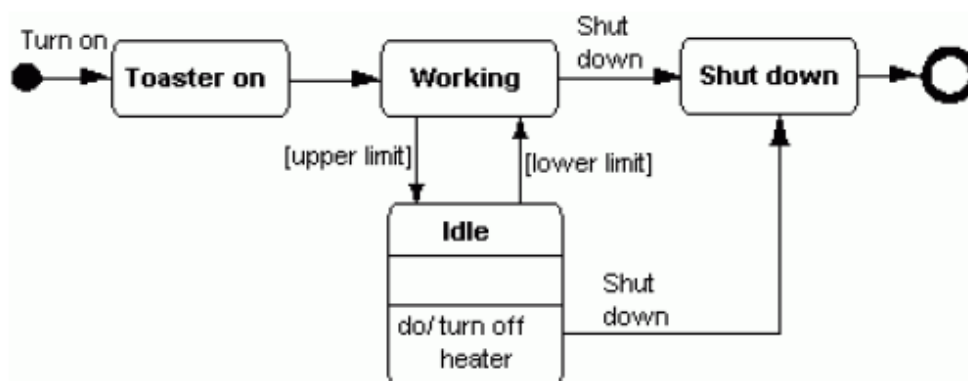
= nisu potrebni svim klasama



Prijelazi:

- predstavljaju promjenu stanja iz trenutnog u novo (novo stanje može biti to isto stanje)
- inicirani su događajima i uvjetima, a opisuju se kao **događaj [uvjet] / akcija**

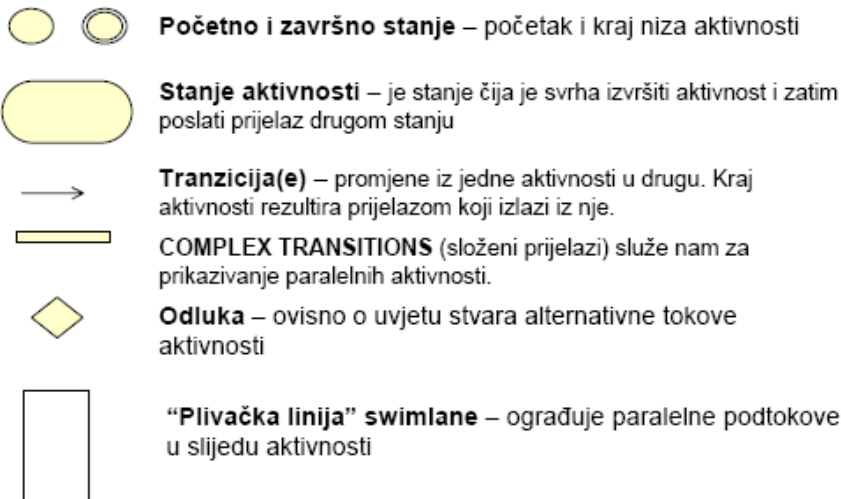
Primjer:



Dijagram aktivnosti

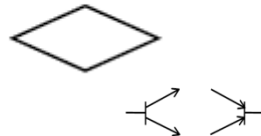
= proširenje dijagrama stanja. Dijagram stanja pokazuje stanje objekta, a aktivnosti prikazuje kao strelice koje povezuju stanje, dok dijagram aktivnosti ističe baš aktivnosti.

Elementi dijagrama aktivnosti



ODLUKE

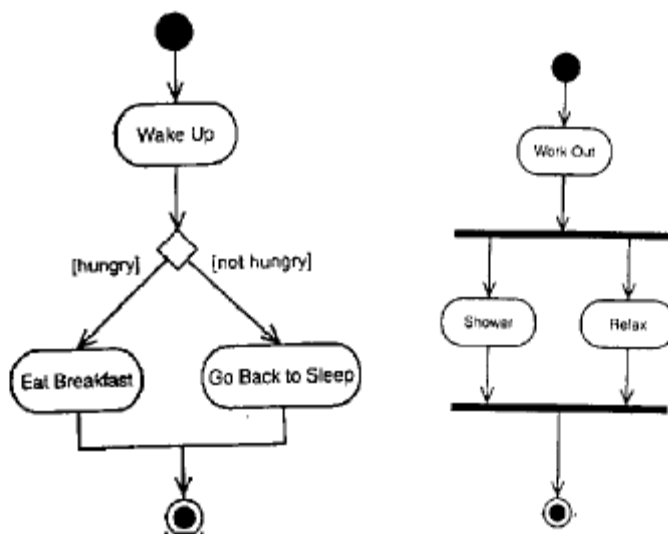
grananje i spajanje

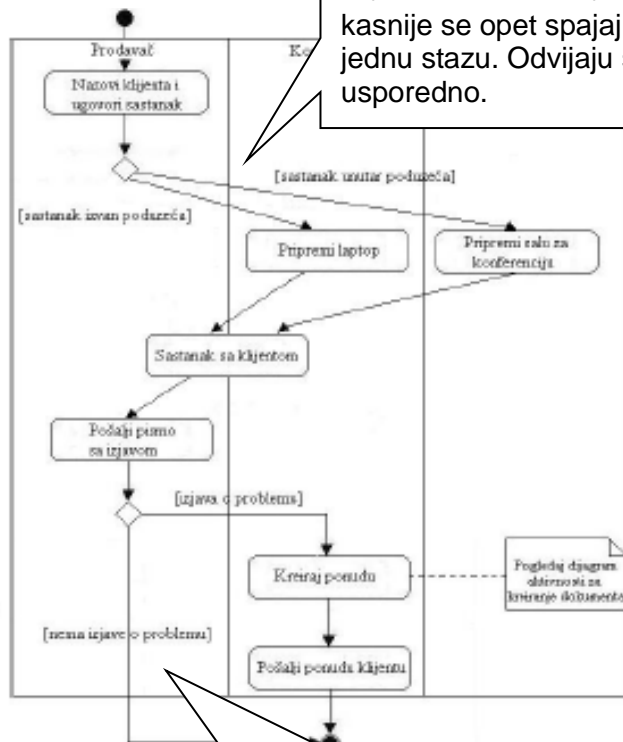
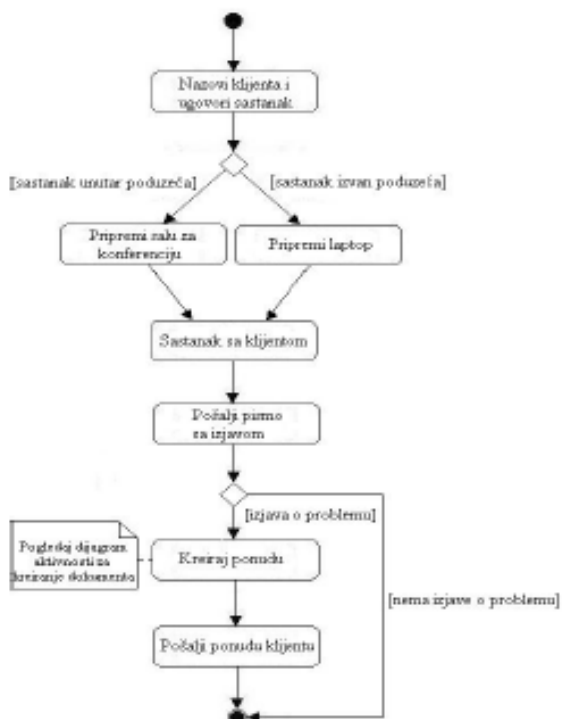


sinkronizacija



Primjer:





Usporedne staze su staze koje kreću u isto vrijeme, a kasnije se opet spajaju u jednu stazu. Odvijaju se usporedno.

Swimlane pokazuje ime uloge na vrhu i prikazuje aktivnost te uloge.

Dijagram komponenti

= prikazuje fizičku strukturu, organizaciju i ovisnosti između komponentata sustava

Sadrži:

- Pakete komponenti
- Komponente
- Sučelja
- Relaciju zavisnosti

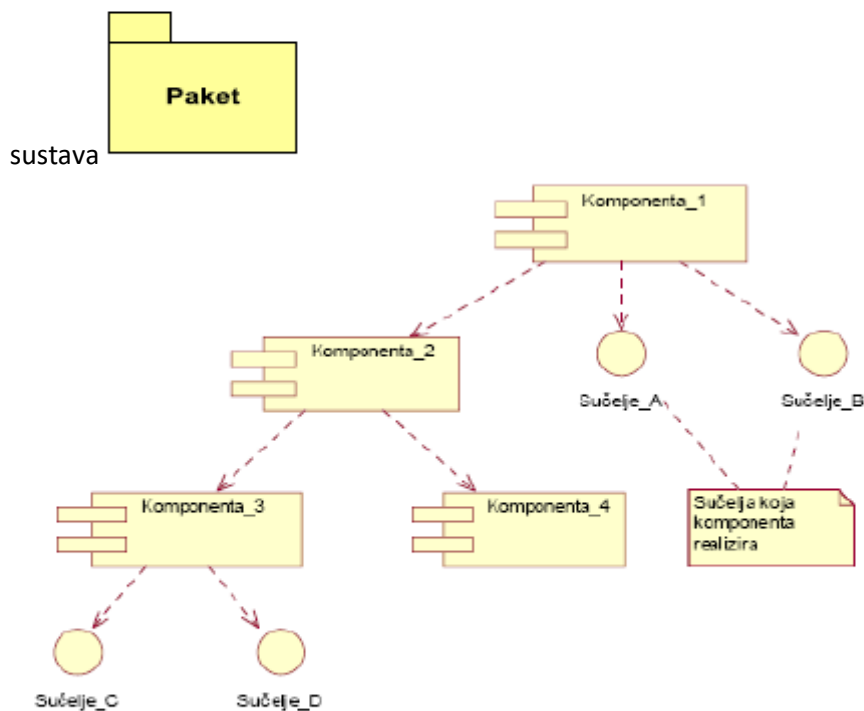
Komponenta:

- izvorna komponenta – kod tijekom kompajliranja
- binarna komponenta: objektni kod (statičke ili dinamičke biblioteke)
- izvršna komponenta - .exe program

Vrste komponentata:

- izvršne (executables)
- biblioteke (library)
- tablice (tables)
- datoteka (file)
- dokument (document)

Paket komponenti - skupina logički povezanih komponenti, služe za podjelu fizičkog modela



Stereotipovi komponenti: se koriste da bi se razlikovali tipovi komponenti u sustavu

- glavni program
-
- The diagram shows a UML component diagram for a main program. It consists of a yellow rectangle with a dark grey header bar. Above the rectangle is the text 'NewMainSubprog'.



- **zadaca**
(paket s nezavisnim nizom upravljackih kontrola)



- **exe**



- **ddl**

Veze: Generalizacija, realizacija, zavisnost, asocijacija

Kolaboracijski dijagram / Dijagram međudjelovanja

= pokazuje koji objekti surađuju u sustavu

Vremenski redoslijed je prikazan brojevima na porukama

Pokazuje:

- Objekte koji sudjeluju u interakciji
- Slijed razmijenjenih poruka

Svaka poruka ima broj slijeda. Poruka na najvišem nivou je označena s 1. Poruke na istom nivou (one koje su poslone tijekom istog poziva) imaju isti decimalni prefiks, a sufixi su im 1, 2, 3, itd. prema slijedu pojavljivanja.

Primjer:

