

Projektovanje softvera

Dijagrami komponenata



Uvod

- Komponenta je zamenjivi deo sistema koji realizuje skup interfejsa
- Dijagram komponenata prikazuje organizaciju i zavisnosti između komponenata
- Namenjen je prikazu:
 - organizacije softverskog sistema
 - statičkog pogleda na implementaciju sistema
- Dijagrami sadrže:
 - stvari:
 - komponente, artefakte, portove, interfejse, klase, pakete
 - relacije:
 - zavisnosti, generalizacije, asocijacije, realizacije

Najčešće primene

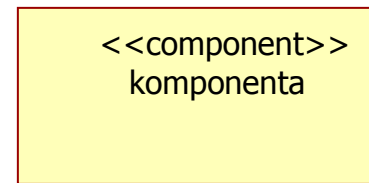
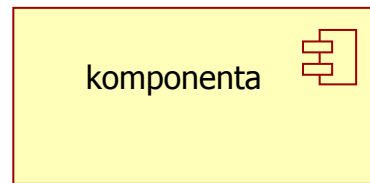
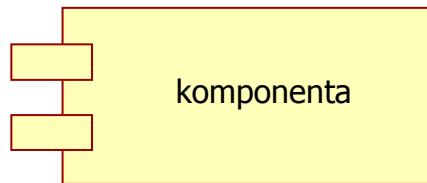
- Modeliranje
 - izvornog koda
 - izdanja za isporuku
 - izvršnih izdanja i okruženja
 - fizičkih baza podataka

Komponenta

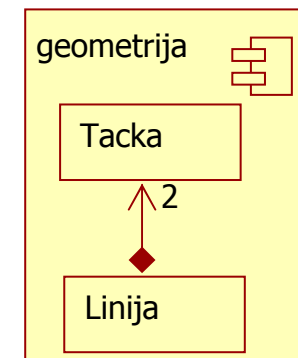
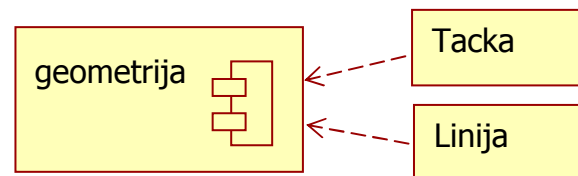
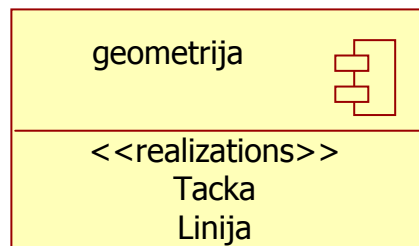
- Komponenta je modularni deo sistema
 - manifestacija je zamenjiva u okruženju
 - kapsulira neki sadržaj (koji nije vidljiv osim kroz interfejs)
 - definiše svoje ponašanje kroz ponuđene i zahtevane interfejse
- Komponenta predstavlja tip (apstrakciju stvari)
 - obuhvata statičku i dinamičku semantiku
- Primeri komponentata
 - Java Bean, EJB, CORBA, COM+, .NET assembly
- Razvoj zasnovan na komponentama
 - razvijani sistem se gradi i strukturira od postojećih komponentata
 - bitna osobina – reupotreba ranije razvijenih komponentata

Grafička notacija

- (UML 1):
- (UML 2):



- Komponenta može da sadrži i odeljak klasa koje realizuje



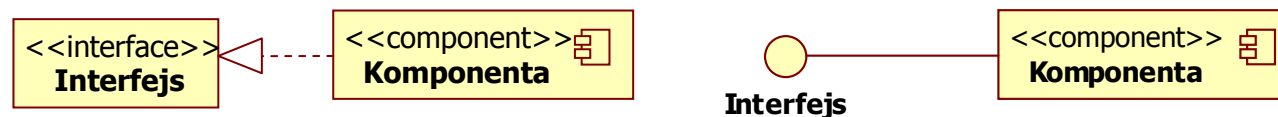
Artefakt

- Artefakt je fizička informacija koju koristi ili proizvodi
 - razvojni proces ili
 - izvršenje
- Primeri:
 - modeli, izvorni fajlovi, skriptovi, binarni izvršni fajlovi, arhive, tabele baze podataka, dokumenti
- Može da predstavlja manifestaciju komponente
- Notacija:



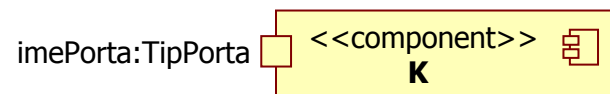
Komponente i klase/interfejsi

- Komponente i klase
 - komponenta predstavlja "pakovanje" logičkih apstrakcija (klasa) u implementaciji
 - klase mogu imati attribute i operacije, a komponente samo operacije
 - servisi klase mogu biti pristupačni direktno, a servisi komponente samo kroz interfejse
- Komponente i interfejsi
 - interfejs je skup operacija koji specificira servis klase ili komponente
 - komponenta realizuje jedan ili više interfejsa
 - standardi COM, CORBA, EJB (Enterprise Java Beans) koriste interfejse
 - relacija realizacije interfejsa (kanonička i skraćena forma):



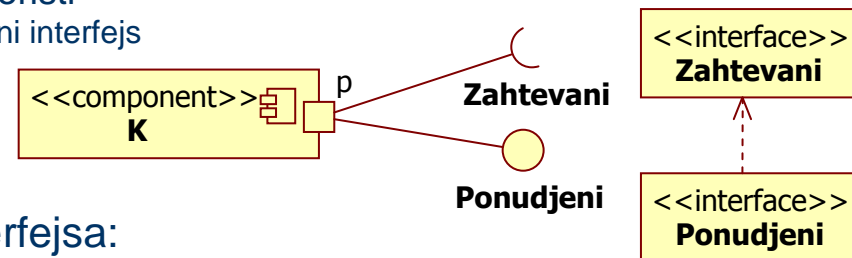
Port

- Port reprezentuje tačku interakcije između klasifikatora i njegovog okruženja
- Interfejsi pridruženi portu specificiraju prirodu interakcija koje se dešavaju preko porta:
 - zahtevani interfejsi karakterizuju zahteve koje može da pravi klasifikator svom okruženju
 - ponuđeni interfejsi karakterizuju zahteve koje okruženje može da pravi klasifikatoru
- Komponenta potencijalno ispoljava interfejse preko portova
- Notacija:

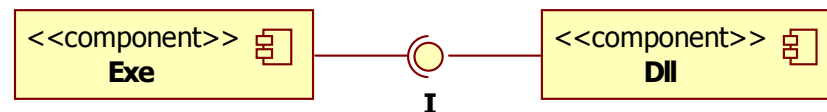


Port i interfejsi

- Port sa interfejsima (postolje/socket - zahtevani, loptica/ball - ponudjeni):
 - Interfejs koji realizuje komponenta
 - izvozni (*export*) ili ponudjeni interfejs
 - Interfejs koji komponenta koristi
 - uvozni (*import*) ili zahtevani interfejs



- Primer interakcije preko interfejsa:



- Konstrukcija postolja i loptice se naziva veznik sklopa (*assembly connector*)
- Ako je port na ivici klasifikatora
 - vidljiv je spolja (*public*), inače je zaštićen (*protected*)
- Port može imati i multiplikativnost – piše se iza imena u zagradama []

Vrste artefakata i stereotipovi

- Mogu se uočiti 3 kategorije artefakata:
 - iz razvojnog procesa
 - modeli, izvorni kod, projektni fajlovi, skriptovi, resursi
 - za isporuku
 - exe, dll, jar, dokumenti, tabele
 - izvršni
 - kreirani kao posledica izvršenja, npr. COM objekat kreiran iz DLL-a
- UML definiše sledeće standardne stereotipove za komponente:
 - **executable** - komponenta koja se može izvršavati na čvoru
 - **library** - statička ili dinamička objektna biblioteka
 - **file** - datoteka (proizvoljan sadržaj)
 - **document** - dokument
 - **script** - skript
 - **source** - datoteka sa izvornim kodom
 - **table** - tabela baze podataka (UML 1)

Paketi, podsistemi i relacije zavisnosti

- Paketi na dijagramima komponenata:
 - sadrže druge pakete i komponente
 - koriste se da predstave fizičko grupisanje komponenata
 - tipično reprezentuju kataloge (foldere) u sistemu datoteka
- Logički paketi iz dijagrama klasa se često preslikavaju u pakete u dijagramima komponenata
- U UML1 podsistemi su bili stereotip <<subsystem>> paketa
- U UML 2 podsistemi su stereotip komponente
- Paketi ili komponente se povezuju relacijom zavisnosti koja reprezentuje:
 - zavisnost u vreme prevođenja kada se radi sa fajlovima izvornog koda
 - zavisnost u vreme povezivanja kada se radi sa bibliotečkim i objektnim fajlovima
 - zavisnost u vreme izvršenja kada se radi sa izvršnim fajlovima

Primer dijagrama komponenata

- Softver za administraciju nekog sistema
 - izvršna komponenta *Administracija*
 - za sifrovanje podataka koji se razmenjuju sa administriranim sistemom
 - koristi se dinamička biblioteka (dll) *Sifrovanje*
 - za najavu administratora na sistem
 - koristi se dinamička biblioteka (dll) *Najava*

