

Java – nasledjivanje

Vrlo često novi programi nastaju proširivanjem prethodnih. Najbolji način za stvaranje novog softvera je imitacija, doterivanje i proširivanje postojećeg. Tradicionalni metodi razvoja softvera su zanemarivali ovakvo stanovište. U objektno orijentisanom programiranju to je osnovni način rada.

Zamislite slučaj kada imate izvorni kod neke klase. Postavite sebi pitanje, „Kako bi vi mogli da taj kod ponovo iskoristite?“ Mogli biste da ga iskopirate i u kopiji menjate ono što je potrebno. Kod tradicionalnih jezika to je bila jedina mogućnost. Problem je kako da sve ostane dobro organizovano i kako da se do kraja razume originalni kod.

Ako biste imali nekoliko desetina klasa i ako je vašem programu potrebno još nekoliko desetina klasa, na kraju biste prethodno opisanom tehnikom kopiranja stigli do više desetina datoteka sa izvornim kodom. Postavite sebi pitanje "Koliki bi problem nastao ukoliko kod originalne klase sadrži neku grešku? Da li moramo tu grešku ispraviti u svim novim klasama?" Bez pažljivog planiranja završili biste sa reorganizovanom gomilom koda, prepunom bagova.

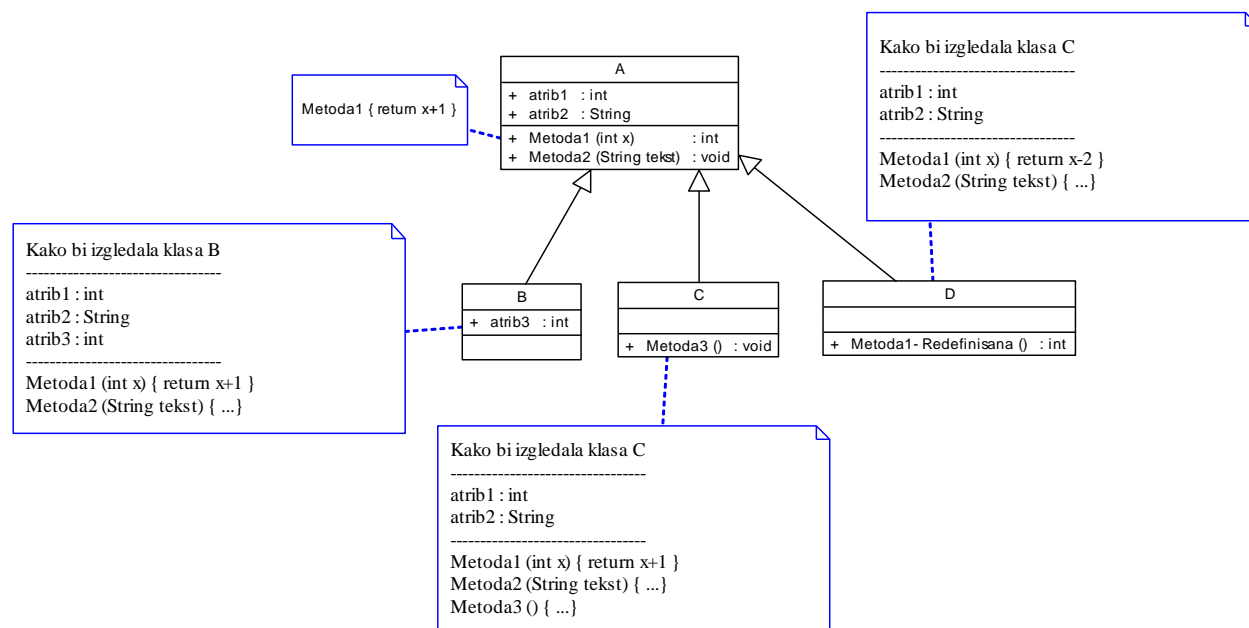
Relacija nasleđivanja omogućuje proširenje ponašanja postojeće klase.

Klasa koja nasleđuje drugu klasu (izvedena klasa) preuzima sve atribute i metode klase koju nasleđuje (osnovna klasa), efekat je sličan kao kad bi mi ručno prekopirali kod osnovne klase u izvedenoj klasi - ali nismo.

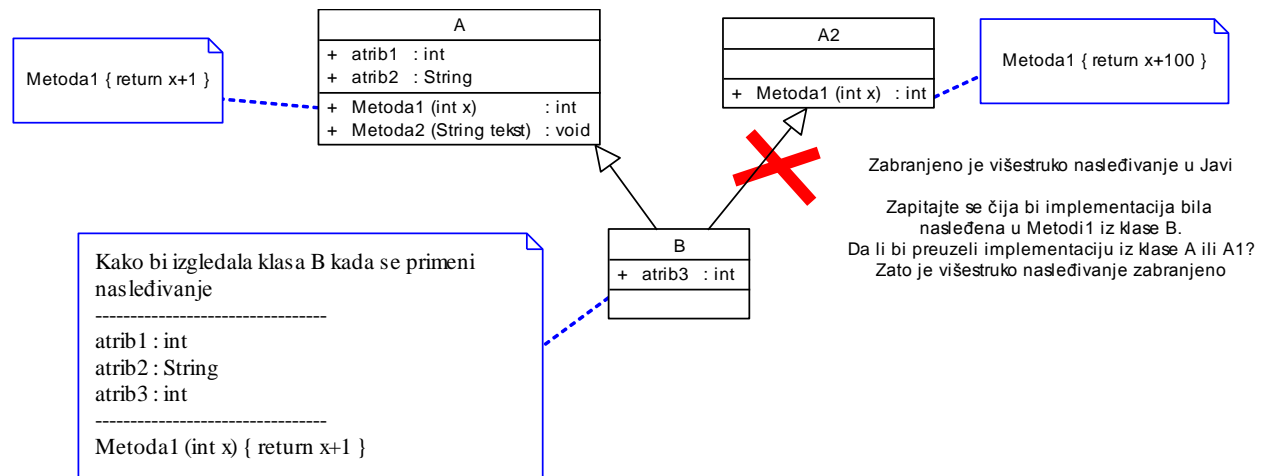
Npr. klase B,C,D nasleđuju klasu A. Izvedene klase B,C,D (podklasa – subclass, dete klasa - child class) predstavljaju jednu specijalnu vrstu osnovne klase A (nadklasa - superclass, roditeljska klasa - parent class), gde klase B,C,D nasleđuje sve atribute i sve metode od klase A.

Nova izvedene klase mogu da:

- proširi strukturu podataka osnovne klase A dodavanjem novih atributa (klasa B)
- proširi funkcionalnost osnovne klase A dodavanjem novih metoda (klasa C)
- izmeni funkcionalnost osnovne klase A redefinisanjem postojećih metoda (klasa D)



Java podržava samo jednostruko nasleđivanje.



Nasleđivanje

- **Postoji samo jednostruko nasleđivanje**
 - **Jedna klasa može samo jednu naslediti, ali više klasa može nasleđivati istu klasu**
- Ako ništa ne napišemo klasa nasleđuje Object klasu
- Ključna reč **extends**
- Primer nasleđivanja: student je osoba koja studira, profesor je osoba koja predaje na fakultetu. Imamo osnovnu klasu osoba (JMBG, ime i prezime, grad) i specijalizacije bi bile student (osoba koja ima indeks i ocene) i profesor (osoba koja radi na fakultetu i ima zvanje, platu, radno mesto, predmete).
- Klasa koja nasleđuje drugu klasu ima sve metode i attribute klase koju nasleđuje i može dodavati nove attribute i metode, ali može i redefinisati postojeće metode
- Zavisno od modifikatora pristupa metode i atributi klase pretka su:
 - vidljivi unutar metoda klasa naslednica i mogu se pozivati nad objektima klasa naslednica – public, protected
 - nisu vidljivi unutar metoda klasa naslednica i ne mogu se pozivati nad objektima klasa naslednica – private, unspecified