

# UML DIJAGRAMI SEKVENCI ZA ODRŽAVANJE ODNOSA ZAVISNOSTI IZMEĐU KLASA

## Uvod

Zavisnost (tradicionalni model podataka) se u UML metodologiji modelira agregacijom (dijagram klasa). Radi jednostavnosti, za identifikaciju objekata koriste se ID-ovi iz baze podataka, mada se u stvari koriste reference na objekte (Java).

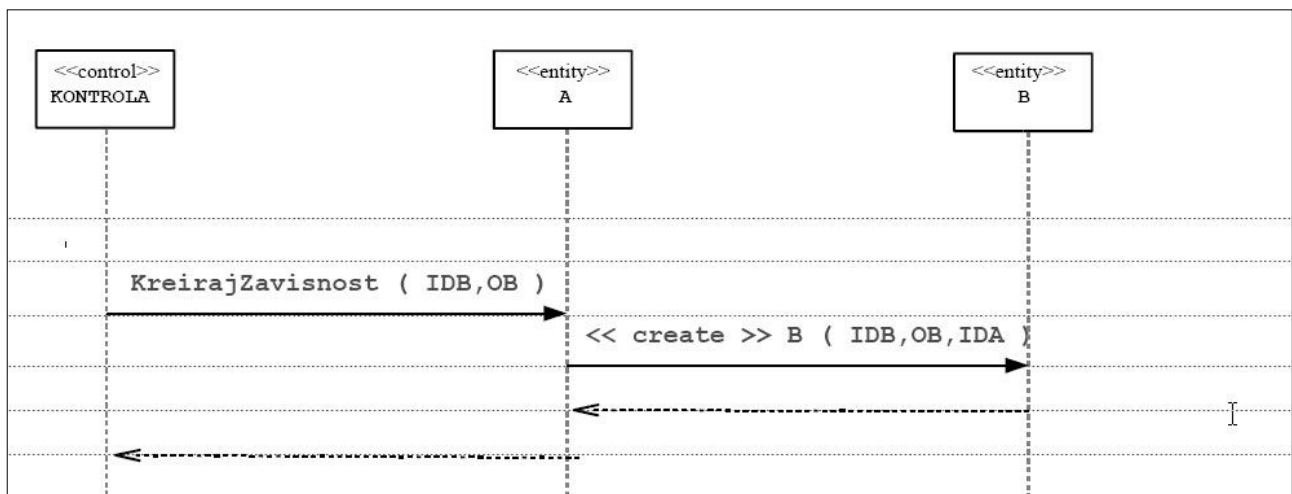
Ako je neka klasa X u odnosu na nekom drugom klasom, podaci o tome se čuvaju u vidu dodatnog dinamičkog atributa klase u vidu:

- skalar koji sadrži referencu na objekat koji je drugi učesnik u odnosu, ako je kardinalnost učešća klase X u odnosu 0..1 ili 1..1; ako odnosa nema, taj skalar ima vrednost null;
- skupa koji sadrži reference na objekte koji su drugi učesnici u odnosu, ako je kardinalnost učešća klase X u odnosu 0..N ili 1..N; ako odnosa nema, taj skup je prazan.

## Egzistencijalna zavisnost stepena 1

### Kreiranje zavisnosti

Na slici je prikazan dijagram sekvenci za kreiranje egzistencijalne zavisnosti između



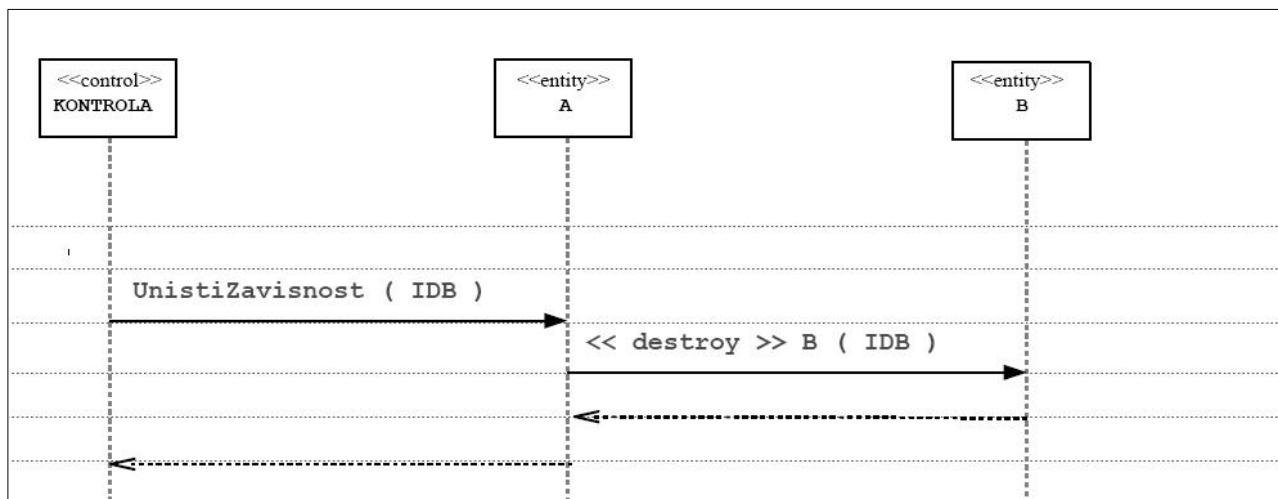
objekta klase A kao uslovitelja i objekta klase B kao uslovljenog, pri čemu objekat A inicira kreiranje objekta B. Klasa kontrolor upućuje poruku objektu klase A **KreirajZavisnost** (samim tim je obezbeđena njegova identifikacija) sa identifikacijom objekta zavisne klase i njegovim svojstvima.

Na taj način, klasa kontrolor je odgovorna samo za opštu logiku slučaja upotrebe, a detalji uspostavljanja odnosa su odgovornost domenskih klasa. Drugim rečima, kontrola je distribuirana umesto centralizovana, što je bolje rešenje.

### Uništavanje zavisnosti

Na slici je prikazan dijagram sekvenci za uništavanje egzistencijalne zavisnosti između objekata klase A i B, odnosno uništavanje zavisnog objekta B (u većini slučajeva, pošto je u bazama podataka ponekad dozvoljeno i raskidanje zavisnosti a da pri tome

zavisni objekat nastavi da postoji). Klasa kontrolor upućuje poruku **UnistiZavisnost** objektu klase A.

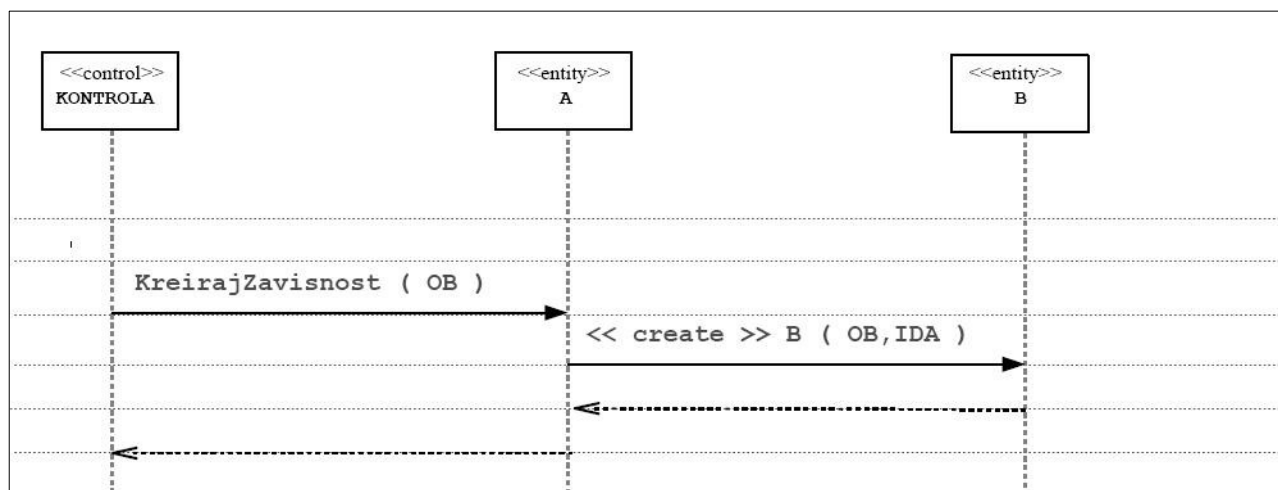


Treba naglasiti da je poruka **<<destroy>>** za određene implementacije (Java pre svega) konceptualna i da u tim implementacijama metod **UnistiZavisnost** samo u objektu klase A uklanja referencu na zavisni objekat klase B, a da stvarno uništavanje tog objekta obavlja "garbage collector".

## Identifikaciona zavisnost

### Kreiranje zavisnosti

Na slici je prikazan dijagram sekvenci za kreiranje identifikacione zavisnosti između

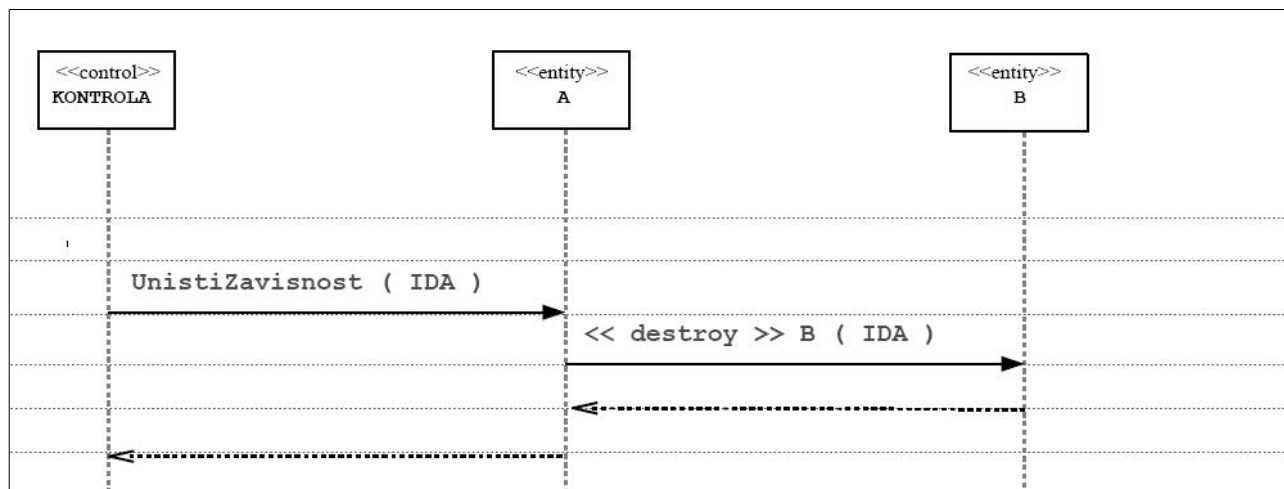


objekata klase A i B, pri čemu objekat A inicira kreiranje objekta B. Klasa kontrolor upućuje poruku objektu klase A **KreirajZavisnost** (samim tim je obezbeđena njegova identifikacija) sa identifikacijom objekta zavisne klase i njegovim svojstvima.

Prethodno se odnosi za slučaj totalne identifikacione zavisnosti, kada jedan obekat klase A može da uslovi nastanak najviše jednog objekta klase B. U slučaju parcijalne identifikacione zavisnosti metod **KreirajZavisnost** i konstruktor **B** treba da imaju kao dodatni argument vrednost parcijalnog identifikatora.

## Uništavanje zavisnosti

Na slici je prikazan dijagram sekvenci za uništavanje identifikacione zavisnosti između objekata klasa A i B, odnosno uništavanje zavisnog objekta B. Za poruku

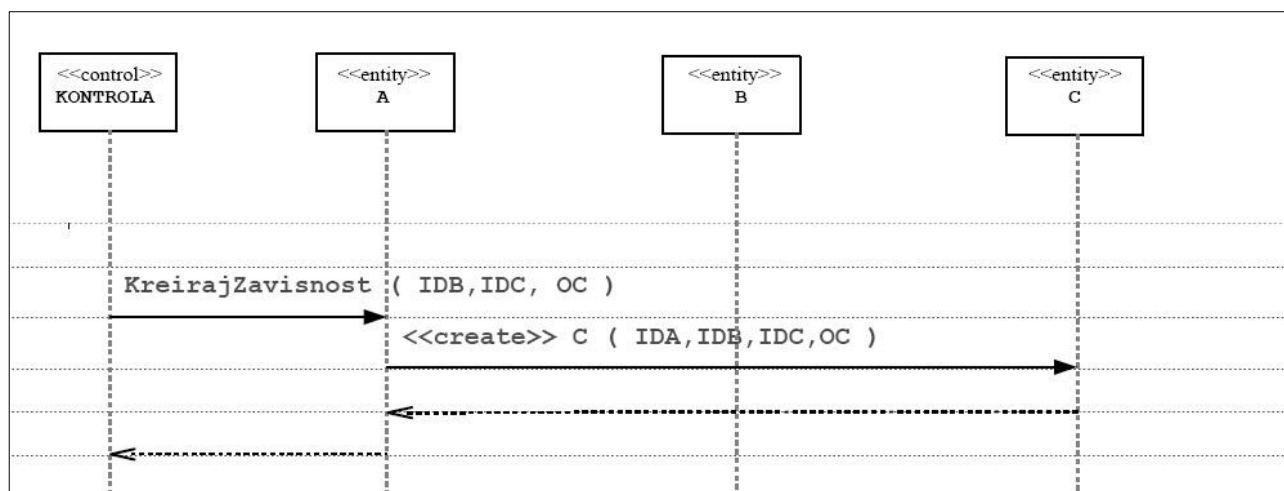


**<<destroy>>** važi ranija napomena.

## **Egzistencijalna zavisnost stepena većeg od 1**

### Kreiranje zavisnosti

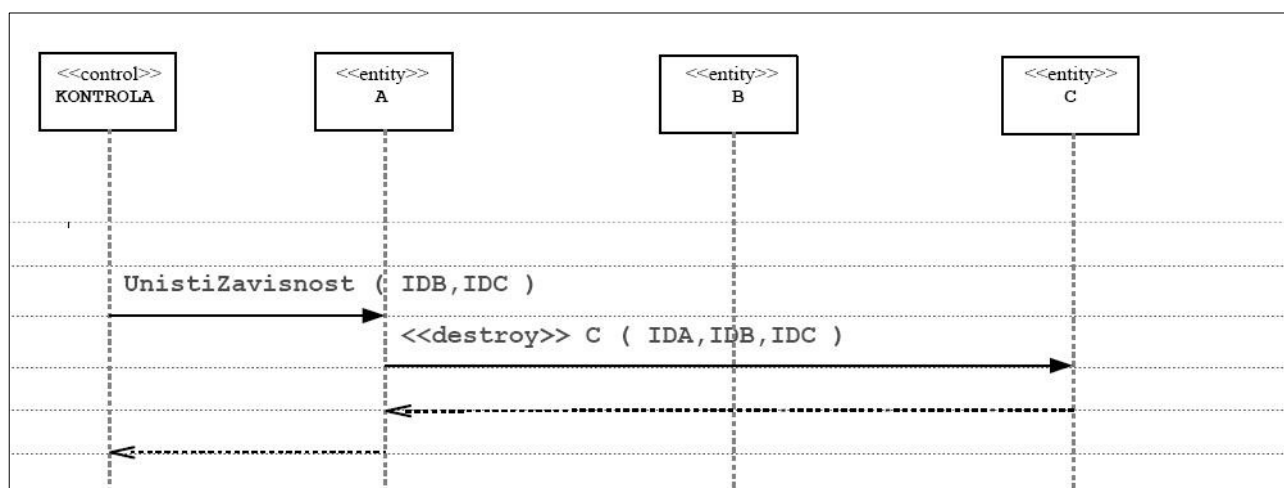
Na slici je prikazan dijagram sekvenci za kreiranje egzistencijalne zavisnosti između



objekata klasa A i B kao uslovitelja i objekta klase C koji se kreira kao uslovljeni. Klasa kontrolor upućuje poruku objektu klase A (a može i objektu klase B) **KreirajZavisnost** (samim tim je obezbeđena njegova identifikacija) sa identifikacijom objekta klase B i zavisnog objekta klase C i njegovim svojstvima. Pri tome objekat klase A ažurira i svoje reference na objekte klase C i te reference kod objekta klase B (kao što je ranije rečeno, reference na objekte učesnike u odnosu su **public**).

## Uništavanje zavisnosti

Na slici je prikazan dijagram sekvenci za uništavanje egzistencijalne zavisnosti između objekata klasa A i B , odnosno uništavanje zavisnog objekta B. Bilo kom od



objekata uslovitelja (prikazan je slučaj za objekat klase A) kontrolor klasa upućuje poruku **UnistiZavisnost**. Za **<<destroy>>** važi ranija napomena.