

# Projektovanje softvera

Lanac odgovornosti



# Lanac odgovornosti (1)

- Ime i klasifikacija:
  - Lanac odgovornosti (engl. *Chain of Responsibility*)
    - objektni uzorak ponašanja
- Namena:
  - povezuje objekte primaoca zahteva u lanac i prosleđuje zahtev niz lanac, dok ga neki objekat ne obradi
  - izbegava neposredno vezivanje pošiljaoca zahteva sa primaocem, dajući šansu većem broju objekata da obrade zahtev

# Lanac odgovornosti (2)

- Motivacija:
  - sistem pomoći zavisne od konteksta u grafičkom korisničkom interfejsu
    - treba da se omogući da od svake komponente može da se dobije pomoć
    - ako za konkretnu komponentu ne postoji pomoć, treba da se prikaže pomoć za dijalog koji sadrži komponentu
    - ako ni za dijalog ne postoji pomoć, treba da se prikaže pomoć za aplikaciju
  - informacije pomoći se organizuju prema opštosti
    - od specifičnijih (u komponentama) prema opštijim (dijalog, aplikacija)
  - zahtev za pomoć treba da obrađuje nekoliko objekata
    - od specifičnijih ka opštijim, dok neko ne obradi u celini

# Lanac odgovornosti (3)

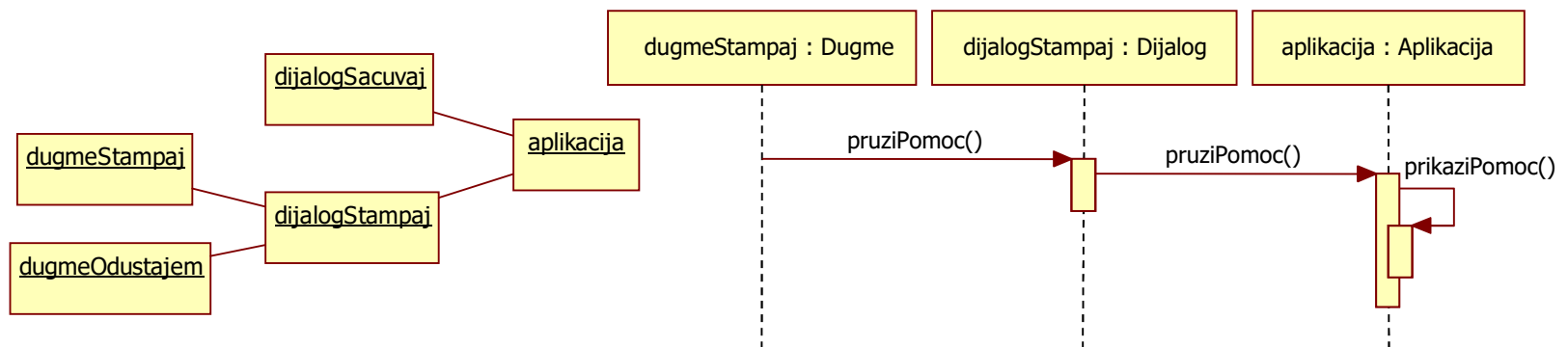
- Motivacija (nastavak):

- problem:

- objekt koji pruža pomoć nije unapred poznat onome ko zahteva pomoć
- potrebno je razdvojiti onoga ko inicira zahtev za pomoć (dugme) od objekta koji će na kraju da prikaže pomoć

- rešenje:

- lanac odgovornosti: zahtev se predaje duž lanca dok ga neko ne obradi

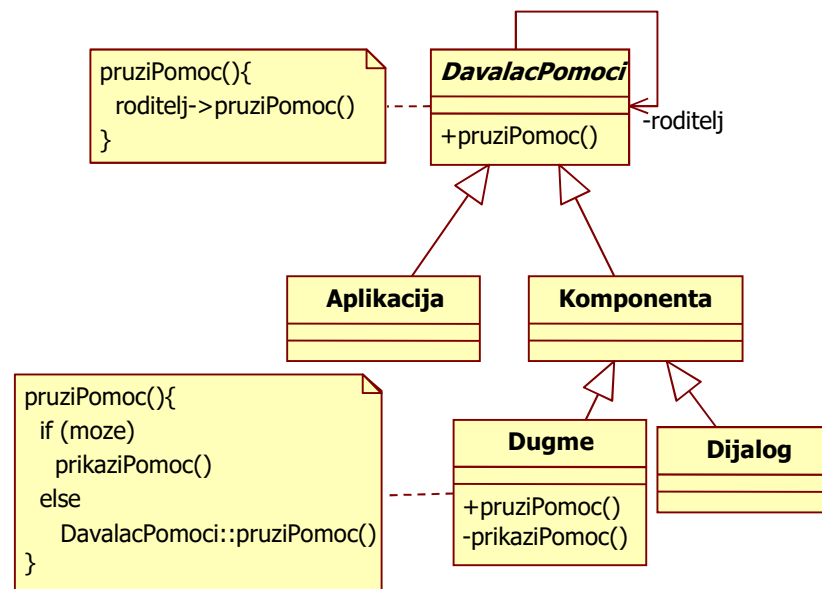


Lanac odgovornosti

27.12.2023.

# Lanac odgovornosti (4)

- Motivacija (nastavak):
  - da bi se obezbedilo prosleđivanje kroz lanac a da primaoci ostanu implicitni, svi objekti u lancu treba da imaju isti ugovor primanja zahteva

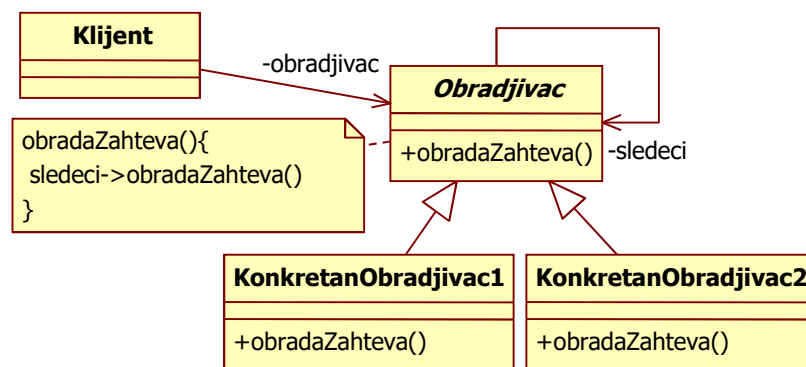


# Lanac odgovornosti (5)

- **Primenljivost: uzorak treba da se primeni**
  - kada više objekata može da obradi zahtev, ali se ne zna unapred koji od njih će da ga obradi
  - kad treba da se izda zahtev jednom od nekoliko objekata, a da se eksplicitno ne odredi primalac
  - kada skup objekata koji obrađuju zahtev treba da se odredi dinamički (vrši se konfigurisanje lanca u vreme izvršenja)

# Lanac odgovornosti (6)

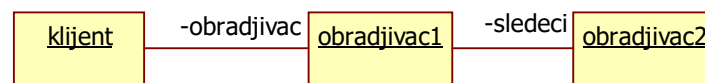
- Struktura:



- Učesnici:

- *Obradivac* (klasa *DavalacPomoci*)
  - definiše ugovor obrade zahteva
  - implementira vezu prema sledećem u lancu
    - opciono, svi osim poslednjeg
- *KonkretanObradivacX* (klase *Dugme*, *Dijalog*)
  - obrađuje zahteve koje ume
  - može da pristupi sledbenicima u lancu
  - ako može da obradi zahtev – to i čini, u suprotnom ga prosleđuje sledbeniku

- Saradnja:



- kada klijent izda zahtev on putuje po lancu odgovornosti dok konkretni obrađivač ne preuzme odgovornost za obradu

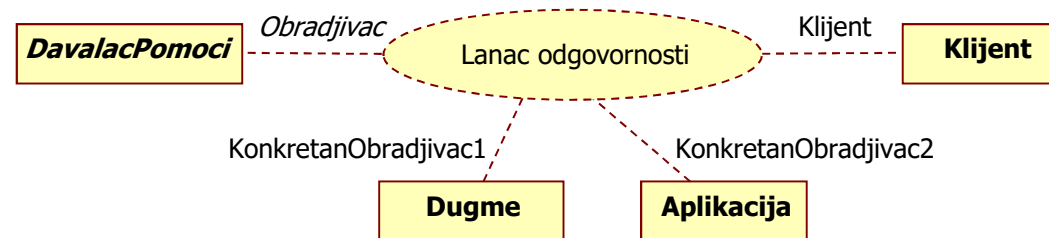
# Lanac odgovornosti (7)

- Posledice:
  - raspredanje pošiljaoca i primaoca
    - pošiljalac i primalac ne treba da znaju ništa jedan o drugom
    - objekat u lancu ne treba da poznaje strukturu lanca
    - smanjuje se broj veza među objektima
      - objekat ne mora da pristupa svima, dovoljno je sledbeniku
  - dodatna fleksibilnost u pridruživanju odgovornosti za obradu zahteva
    - odgovornosti za obradu zahteva mogu da se dodaju i menjaju u vreme izvršenja
  - prijem nije garantovan
    - moguća je situacija da zahtev stigne do kraja lanca, a da nije obrađen



# Lanac odgovornosti (8)

- UML notacija:



- Povezani uzorci:

- često se primenjuje sa uzorkom *Sastav*;  
roditelj komponente može da igra ulogu “sledećeg” u lancu
- *Dekorater* ima sličnu strukturu, ali
  - kod njega svi objekti u lancu nužno učestvuju u obradi zahteva
  - zahtev se svakako na kraju obrađuje od strane dekorisanog subjekta