

# Projektovanje softvera

Kompozicija

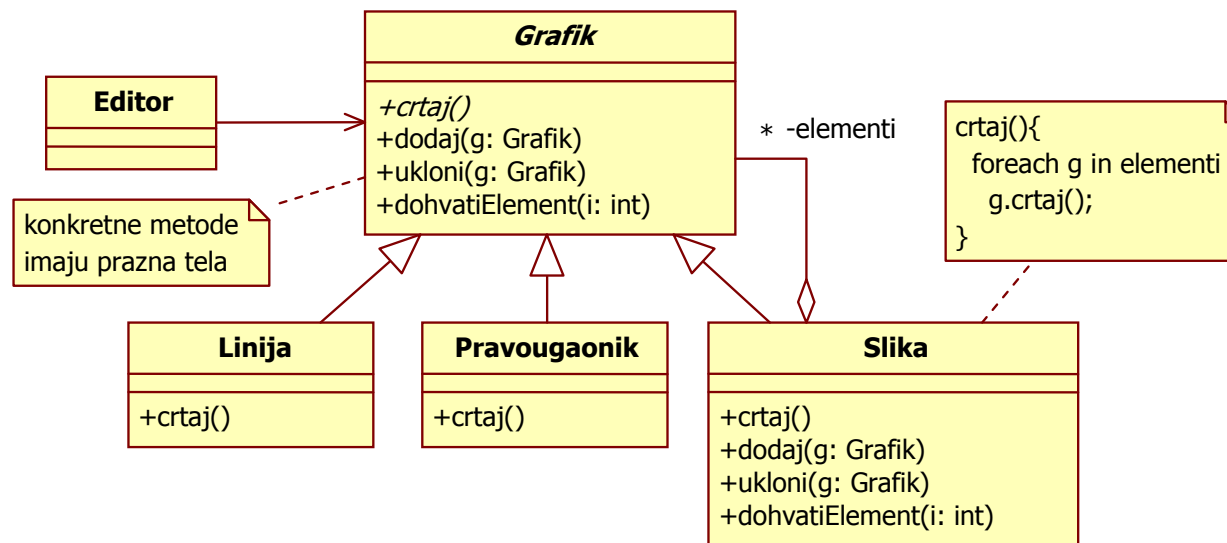


# Kompozicija (1)

- Ime i klasifikacija:
  - Kompozicija (Sklop, Sastav, engl. *Composite*) – objektni uzorak strukture
- Namena:
  - komponuje objekte u strukturu stabla (hijerarhija celina-deo)
  - omogućava klijentima da uniformno tretiraju
    - individualne objekte
    - njihove kompozicije
- Motivacija:
  - grafičke aplikacije kao što su editori crteža i sistemi za unos šema:
    - omogućavaju crtanje kompleksnih slika (dijagrama) sastavljenih od jednostavnih grafičkih elemenata (komponenta)
    - elementi se mogu grupisati da se formiraju složeni elementi (kompozicije)
    - složeni element je potrebno tretirati i kao vrstu grafičkog elementa
    - ključno: apstraktna klasa koja predstavlja i proste i složene elemente

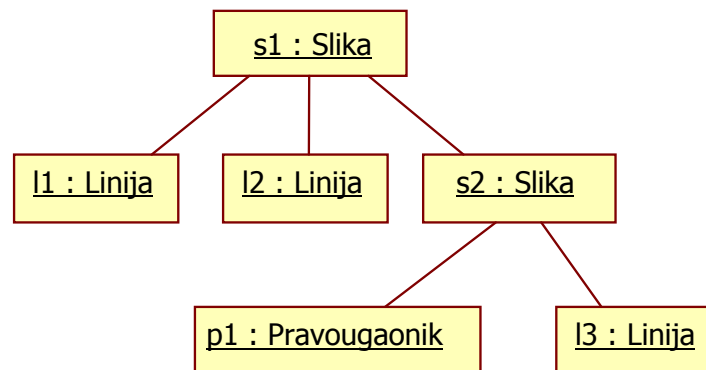
# Kompozicija (2)

- Motivacija (nastavak):
  - klasni dijagram



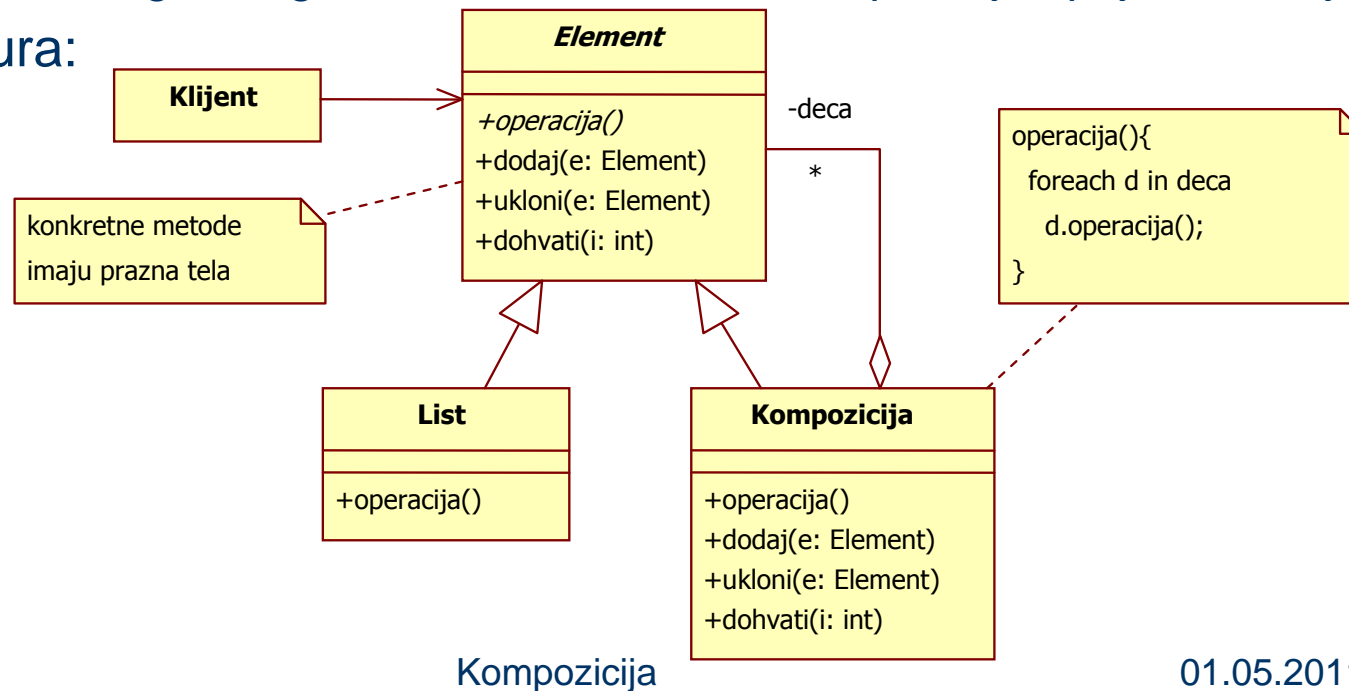
# Kompozicija (3)

- Motivacija (nastavak):
  - objektni dijagram – struktura rekurzivno komponovanih grafičkih objekata:



# Kompozicija (4)

- **Primenljivost:** uzorak treba koristiti kada se želi da
  - postoje hijerarhije objekata celina-deo takve da su celina i deo iste vrste
  - klijenti mogu da ignorišu razlike između kompozicija i pojedinih objekata
- **Struktura:**



# Kompozicija (5)

- Učesnici:
  - Element (klasa Grafik)
    - deklarirše zajednički interfejs za sve objekte u kompoziciji
    - implementira podrazumevano ponašanje zajedničko za sve klase
    - deklarirše interfejs za pristupanje i upravljanje decom
    - implementira prazne metode za pristup i upravljanje decom (zbog listova)
    - opciono deklarirše i implementira interfejs za pristup roditelju
  - List (klase Linija, Pravougaonik)
    - reprezentuje individualne objekte – listove u stablu
    - definiše ponašanje za primitivne objekte u kompoziciji
  - Kompozicija (klasa Slika)
    - definiše ponašanje za objekte koji imaju decu
    - sadrži komponente decu
    - implementira operacije za pristup i upravljanje decom
  - Klijent (klasa Editor)
    - manipuliše objektima u kompoziciji kroz interfejs klase Element



# Kompozicija (7)

- Posledice:
  - čini jedostavnim klijente
    - oni tretiraju na jedinstven način sve objekte u hijerarhiji
  - čini jednostavnim dodavanje nove vrste elemenata
  - nije jednostavno ograničiti vrste elemenata koje neke kompozicije sadrže
- Povezani uzorci:
  - Često se veza element-roditelj koristi za uzorak *Lanac odgovornosti*
    - ako element ne može da odgovori na zahtev – prosleđuje ga roditeljskom objektu
  - *Dekorater (Dopuna)* ima sličnu strukturu klasa i često se koristi sa Kompozicijom
    - kada se koriste zajedno obično imaju zajedničku natklasu
  - *Muva* dozvoljava deljenje elemenata, ali oni ne mogu da pristupaju svojim roditeljima
  - Iterator se koristi često za obilazak strukture *Kompozicije*
  - *Posetilac* lokalizuje operacije i ponašanje koje bi inače bilo distribuirano između klasa `Kompozicija` i `List`