

# Projektovanje softvera

Posetilac



# Posetilac (1)

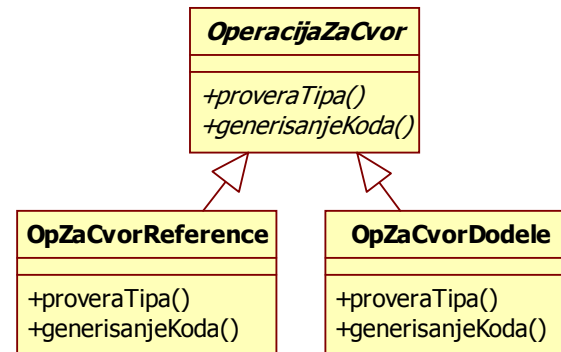
- Ime i klasifikacija:
  - Posetilac (engl. *Visitor*) – objektni uzorak ponašanja
- Namena:
  - skup različitih operacija koje treba primenjivati na elemente jedne objektno strukture
  - omogućava definisanje nove operacije bez izmene klasa elemenata nad kojima se izvršava

# Posetilac (2)

- Motivacija:
  - razmatra se prevodilac koji reprezentuje programe kao stabla apstraktne sintakse (SAS)
  - potrebno je obezbediti operacije nad SAS za:
    - statičku proveru tipova
    - generisanje koda
    - ...
  - većina operacija će na različit način tretirati čvorove SAS za:
    - naredbu dodele vrednosti
    - referisanje promenljive
    - ...
  - hijerarhija čvorova SAS zavisi od jezika, ali je za dati jezik stabilna

# Posetilac (3)

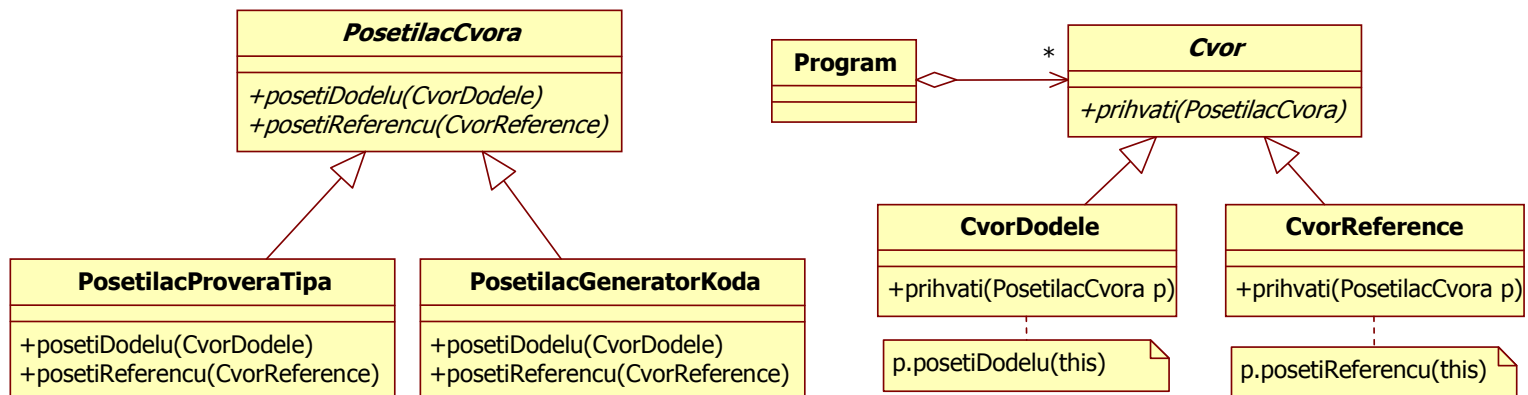
- Motivacija (nastavak):
  - nefleksibilan dizajn:



- problem: prisustvo svih operacija u različitim klasama čvorova vodi do sistema koji je teško razumeti, održavati i menjati
  - mešanje koda za proveru tipova i koda za generisanje koda kviri razumljivost
  - dodavanje nove operacije zahteva ponovno prevođenje svih klasa čvorova
- cilj: razdvajanje hijerarhije čvorova od operacija koje se vrše nad njima
- rešenje: pakovanje operacija u objekte posetilaca koji se prosleđuju čvorovima SAS kada se struktura obilazi

# Posetilac (4)

- Motivacija (nastavak):
  - posetilac kapsulira jednu operaciju za različite čvorove SAS
  - kada čvor primi posetioca – on poziva operaciju posetioca
    - operacija odgovara klasi čvora
    - čvor dostavlja referencu na sebe kao parametar operacije
  - posetilac tada izvršava operaciju za dotični čvor



# Posetilac (5)

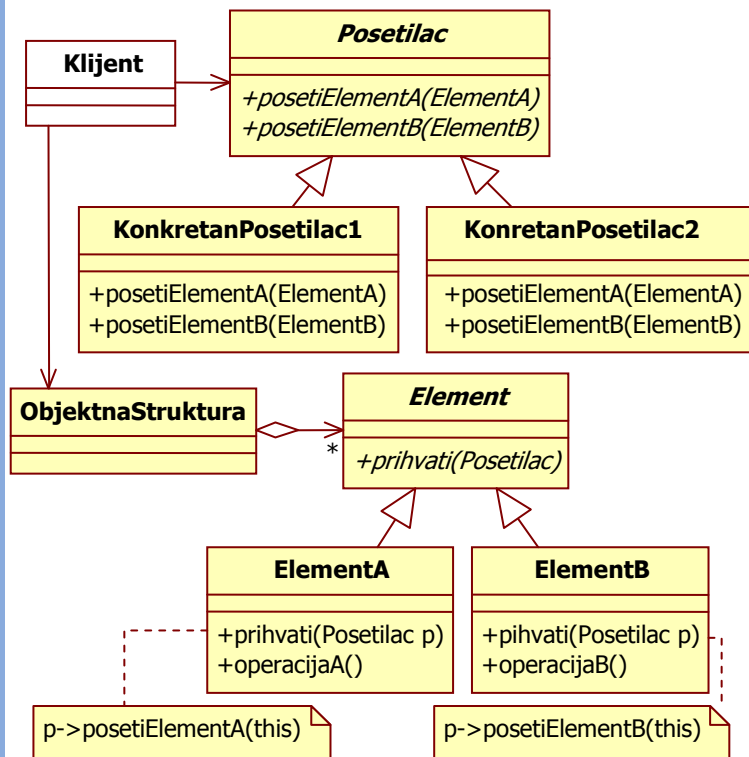
- Motivacija (nastavak):
  - u uzorku *Posetilac* se definišu dve hijerarhije klasa
    - za elemente nad kojima se vrše operacije (h. čvorova)
    - za posetioce koji definišu operacije nad elementima (h. posetilaca)
  - nova operacija se dodaje kao nova potklasa u h. posetilaca
    - ako bi aplikacija trebalo da računa metriku programa samo bi se definisala nova potklasa klase `PosetilacCvora` a ne bi se dodavao aplikativno zavisani kod u klase čvorova SAS

# Posetilac (6)

- **Primenljivost: uzorak treba koristiti kada**
  - jedna objektna struktura sadrži objekte raznih klasa, a potrebno je nad tim objektima izvršiti operacije koje zavise od njihovih konkretnih klasa
  - više nepovezanih operacija se izvršava nad objektima strukture, a ne želimo da “zagadimo” njihove klase tim operacijama
    - posetilac omogućava grupisanje povezanih operacija (koje odgovaraju jednoj primeni) u jednoj klasi
  - klase koje definišu strukturu objekata se retko menjaju, a često se definišu nove operacije nad elementima strukture
    - promena klasa objektna strukture zahteva promenu svih posetilaca
    - može mnogo da košta, pa ako je ovo često, bolje je definisati operacije u tim klasama

# Posetilac (7)

- Struktura:



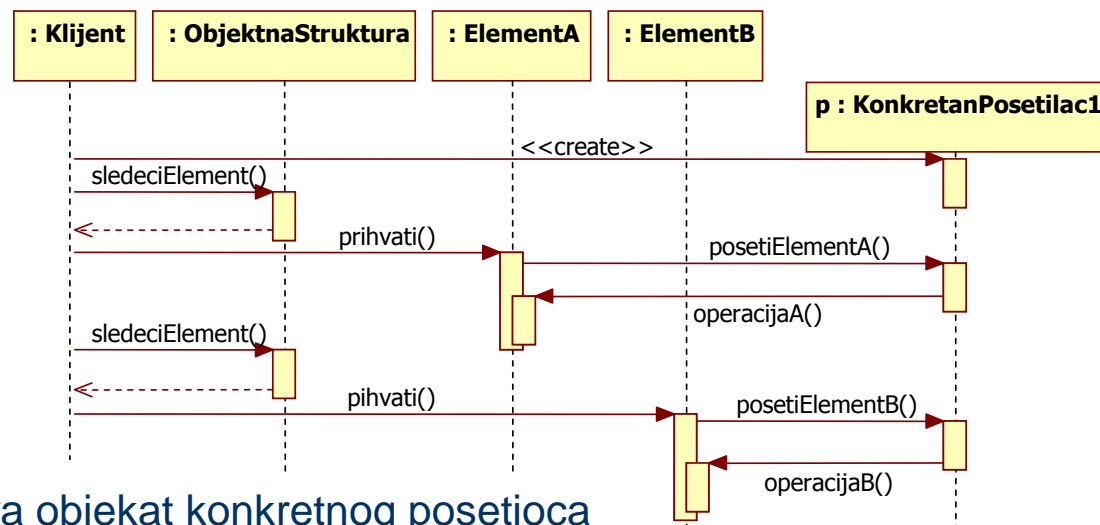
- Učesnici:

- Posetilac (klasa PosetilacCvora)
  - po jedna operacija `poseti()` za svaku potklasu `Element`
  - potpis operacije identifikuje odgovarajući potklasu `Element`
- KoncretanPosetilacX (klase `PosetilacProveraTipa,...`)
  - implementira sve operacije koje deklariraše `Posetilac`
  - predstavlja kontekst algoritma i čuva njegovo stanje koje često akumulira rezultate obilaska strukture objekata
  - svaka operacija implementira deo algoritma za odgovarajuću klasu objekata u strukturi
- Element (klasa `Cvor`)
  - deklariraše operaciju `prihvati()` - posetilac je argument
- ElementX (klase `CvorDodele, CvorReference`)
  - implementira operaciju `prihvati()` tako što poziva `posetilac.posetiElementX(this)`
- ObjektnaStruktura (klasa `Program`)
  - pruža interfejs visokog nivoa koji omogućava klijentu da obiđe elemente dostavljajući im posetioca
  - može da bude objekat `Sastav` (stablo) ili kolekcija (lista)



# Posetilac (8)

- Saradnja:



- Klijent:

- stvara objekat konkretnog posetioca
- obilazi objektnu strukturu posećujući svaki njen element
- svakom elementu prosledi posetioca

- Element

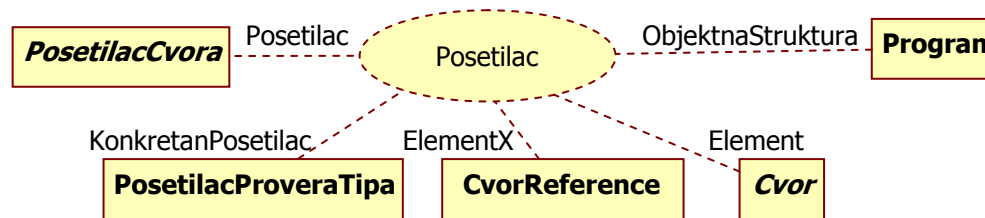
- poziva operaciju posetioca kojom mu traži da ga poseti

# Posetilac (9)

- Posledice:
  - olakšava dodavanje novih operacija (klasa posetilaca)
  - dodavanje novih klasa konkretnih elemenata nije lako
  - grupiše srodne i razdvaja različite operacije
  - mogućnost akumuliranja stanja za vreme posećivanja elemenata
    - bez posetioca, stanje bi se predavalo operacijama kao dodatni argument
  - probijanje kapsuliranja elemenata
    - često su potrebne javne operacije za pristup unutrašnjem stanju elemenata strukture
    - proglašavanje posetioca prijateljem u elementa nije dobro rešenje
      - dodavanje nove operacije (posetioca) bi zahtevalo izmenu svih elemenata

# Posetilac (10)

- UML notacija:



- Povezani uzorci:

- *Kompozicija* – posetilac može da se koristi da bi se neka operacija primenila na strukturu objekata definisanu kao kompozicija
- *Iterator* se često koristi zajedno sa *Posetiocem*,
  - služi za sistematičan obilazak objekata strukture kojima se šalju posetioci
- *Interpreter* – posetilac se može koristiti da obavi interpretaciju