Projektovanje softvera

Strategija

Strategija (1)

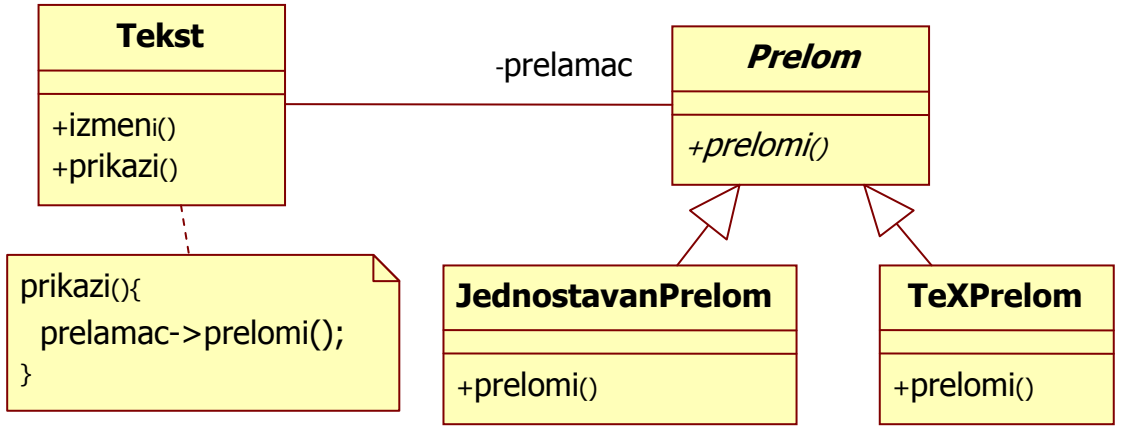
- Ime i klasifikacija:
  - Strategija (engl. *Strategy*) – objektni uzorak ponašanja
- Namena:
  - definiše familiju algoritama, kapsulirajući svaki, i čini ih međusobno zamenjivim
  - omogućava jednostavnu promenu algoritma u vreme izvršenja
- Drugo ime:
  - Politika (*Policy*)

Strategija (2)

- Motivacija:
  - postoji više algoritama za prelom teksta u linije
  - ugrađivanje tih algoritama u klase koje ih zahtevaju nije dobro iz više razloga:
    - klase postaju kompleksnije i teže za održavanje
    - različiti algoritmi su odgovarajući u različitim trenucima, a teško ih je menjati
    - teško je dodati novi algoritam i varirati postojeći
  - ni izvođenje iz klase teksta nije dobro rešenje jer je statičko
  - alternativa - kapsulirani algoritam u posebnu klasu koja se naziva strategija

Strategija (3)

- Motivacija



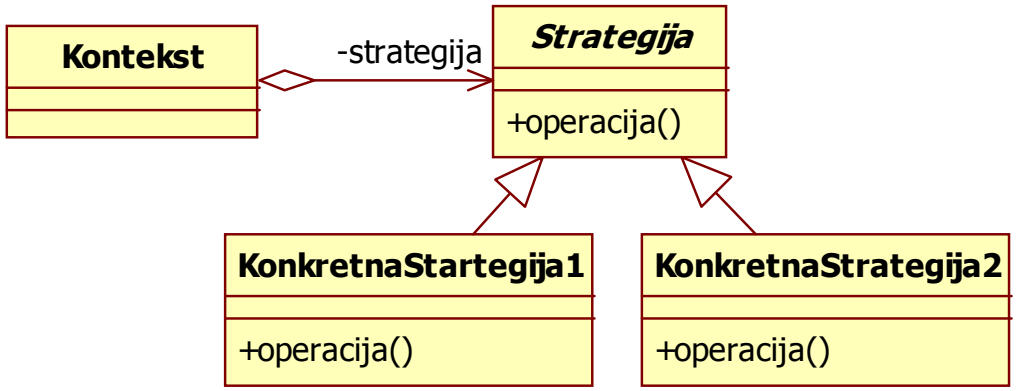
  - klasa `Tekst` je odgovorna za prikaz i izmenu teksta
  - prilikom prikaza treba ažurirati prelom izmenjenog teksta
  - strategije preloma nisu implementirane u klasi `Tekst`
    - Implementirane su u potklasama `Prelom`
  - klasa `Tekst` sadrži referencu na objekat tipa `Prelom`
  - kada se `Tekst` izmeni, da bi se prikazao, treba da se reformatira
    - objekat klase `Tekst` prosleđuje tu odgovornost objektu klase `Prelom`
  - klijent klase `Tekst` specificira objekat `Prelom` instalirajući ga u `Tekst`

Strategija (4)

- Primenljivost:
  - kada bi se više srodnih klasa razlikovalo samo po nekom ponašanju
    - uzorak omogućava konfigurisanje jedne klase jednim od više ponašanja
  - kada su potrebne različite varijante nekog algoritma
  - kada algoritam koristi podatke o kojima klijenti ne treba ništa da znaju
    - izbegava se eksponiranje kompleksnih struktura podataka
      - te strukture su specifične za algoritam, klijent ne treba da ih bude svesan
    - potrebno je kapsuliranje algoritma i strukture podataka
  - kada klasa konteksta definiše više ponašanja koja se pojavljuju kao grane uslovne naredbe u raznim operacijama
    - grane uslovne naredbe treba kapsulirati u odgovarajuće strategije
    - jedna strategija u svojim operacijama izvršava samo odgovarajuću granu

Strategija (5)

- Struktura:



- Učesnici:
  - `Strategija` (klasa `Prelom`)
    - deklariše zajednički interfejs za sve podržane algoritme
  - `KonkretnaStrategijaX` (klase `JednostavanPrelom`, `TeXPrelom`)
    - implementira konkretan algoritam tako da odgovara interfejsu klase *`Strategija`*
  - `Konktekst` (klasa `Tekst`)
    - konfigurisan je objektom `KonkretnaStrategijaX`
    - poseduje referencu na objekat tipa `Strategija`
    - može da pruži interfejs koji omogućava objektu strategije da pristupi njegovim podacima

Strategija (6)

- Saradnja:
  - interaguju `Kontekst` i `KonkretnaStrategijaX`
  - `Kontekst` može da pošalje
    - sve podatke koje zahteva algoritam pri pozivu objekta `Strategija`
    - referencu na sebe i tako omogući povratni poziv (*callback*)
  - `Kontekst` prosleđuje zahteve svojih klijenata svom objektu `Strategija`
    - klijenti obično kreiraju i prosleđuju objekte `KonkretnaStrategijaX` objektu `Kontekst`
    - kasnije klijenti interaguju samo sa objektom `Kontekst`
- UML notacija:

Strategija (7)

- Posledice:
  - familije srodnih algoritama definisanih kao hijerarhija klasa `Strategija`
  - fleksibilna alternativa izvođenju iz klase `Kontekst`
    - kad bi `Kontekst` implementirao algoritam
  - strategije eliminišu potrebu za uslovnim naredbama u klijentskom kodu
  - izbor implementacija
    - klijent bira implementaciju da postigne performanse u vremenu/prostoru
  - nedostatak - klijenti moraju biti svesni strategije kojom parametrizuju kontekst
  - `Kontekst` kreira nepotrebne parametre za neke `KonkretnaStrategijaX`
  - povećava se broj objekata u aplikaciji zbog objekata `KonkretnaStrategijaX`
- Povezani uzorci:
  - Objekti *Strategije* često predstavljaju dobre objekte *Muve*