

Projektovanje softvera

Šablonski metod

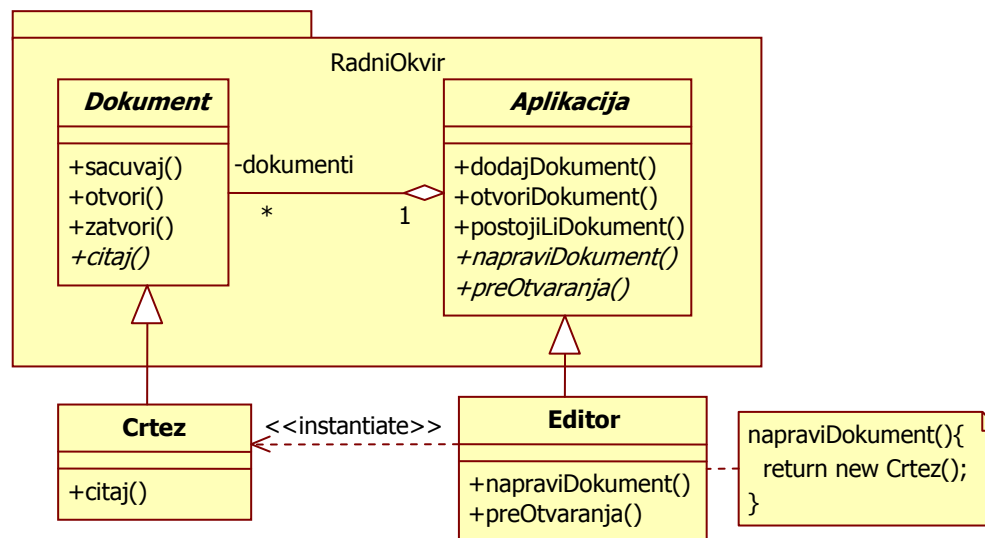


Šablonski metod (1)

- Ime i klasifikacija:
 - Šablonski metod (engl. *Template Method*)
- klasni uzorak ponašanja
- Namena:
 - definiše skelet nekog algoritma operacije, delegirajući pojedine korake potklasama
 - omogućava potklasama da redefinišu neke korake algoritma bez izmene njegove strukture

Šablonski metod (2)

- Motivacija:
 - razmatra se radni okvir za razvoj aplikacije koji obezbeđuje klase `Aplikacija` i `Dokument`



Šablonski metod (3)

- Motivacija (nastavak):
 - klasa `Aplikacija` je odgovorna za otvaranje postojećih dokumenata
 - dokumenti se čuvaju u nekom eksternom formatu (fajlu)
 - objekt klase `Dokument` reprezentuje informaciju u dokumentu nakon što je učitana iz fajla
 - aplikacije koje se grade iz radnog okvira mogu da naslede klase `Aplikacija` i `Dokument`
 - npr. aplikacija za crtanje definiše potklase `Editor` i `Crtež`

Šablonski metod (4)

- Motivacija (nastavak):

- klasa Aplikacija definiše algoritam za otvaranje dokumenta

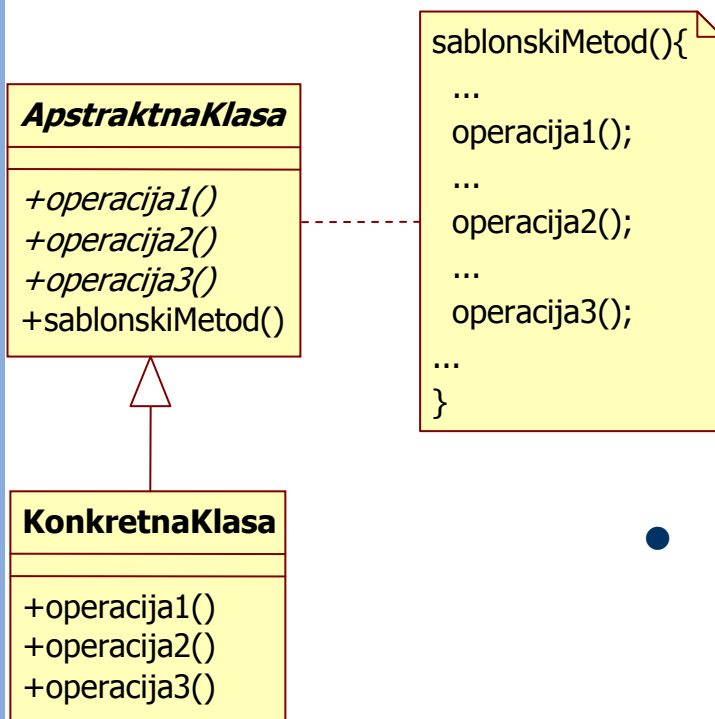
```
void Aplikacija::otvoriDokument(const char* ime){
    if (!postojiLiDokument(ime))return;
    Dokument * dokument=napraviDokument();
    if (dokument!=nullptr){
        dodajDokument(dokument);
        preOtvaranja(dokument);
        dokument->otvori();
        dokument->citaj();
    }
}
```

Šablonski metod (5)

- Motivacija (nastavak):
 - `otvoriDokument()` definiše korake za otvaranje dokumenta
 - metod `otvoriDokument()` se naziva *Šablonski metod*
 - šablonski metod definiše algoritam sastavljen od apstraktnih operacija
 - potklase definišu operacije da obezbede konkretno ponašanje
 - definisanjem koraka algoritma šablonski metod fiksira njihov redosled
 - potklase prilagođavaju korake algoritma svojim potrebama
- Primenljivost: uzorak treba koristiti
 - da se implementiraju invarijantni delovi algoritma jednom, a da se ostave potklasama za implementaciju delovi koji mogu varirati
 - kada zajedničko ponašanje među potklasama treba lokalizovati da se izbegne dupliranje

Šablonski metod (6)

- Struktura:



- Učesnici:

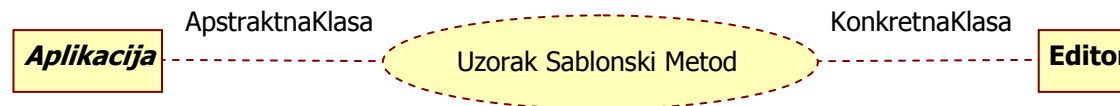
- **ApstraktnaKlasa (klasa Aplikacija)**
 - implementira šablon (skelet) algoritma
 - poziva primitivne operacije
 - deklarira apstraktne primitivne operacije
 - za korake algoritma koje će potklase definisati
- **KonkretnaKlasa (klasa Editor)**
 - implementira primitivne operacije
 - obavljaju delove algoritma
 - delovi su specifični za potklase

- Saradnja:

- **KonkretnaKlasa implementira varijantne korake algoritma za klasu ApstraktnaKlasa**

Šablonski metod (7)

- UML notacija:



- Posledice:
 - fundamentalna tehnika za reupotrebu koda (zajedničko ponašanje u klasama radnog okvira)
 - vodi invertovanoj kontrolnoj strukturi (holivudski princip: "Ne zovite nas, mi ćemo zvati Vas" – roditeljska klasa zove operacije dece)
 - izbegava se rizik da izvedena klasa iz redefinisane operacije ne pozove potrebnu operaciju roditeljske klase
 - mogu se neke primitivne operacije realizovati u apstraktnoj klasi kao rudimentarne (*hook*) operacije koje obezbeđuju podrazumevano ponašanje
- Povezani uzorci:
 - *Fabrički Metod* (npr. `napraviDokument`) se često poziva iz *Šablonskog Metoda*
 - *Šablonski Metod* statički varira deo algoritma, a *Strategija* dinamički varira ceo algoritam