

Projektovanje softvera

Fabrički metod

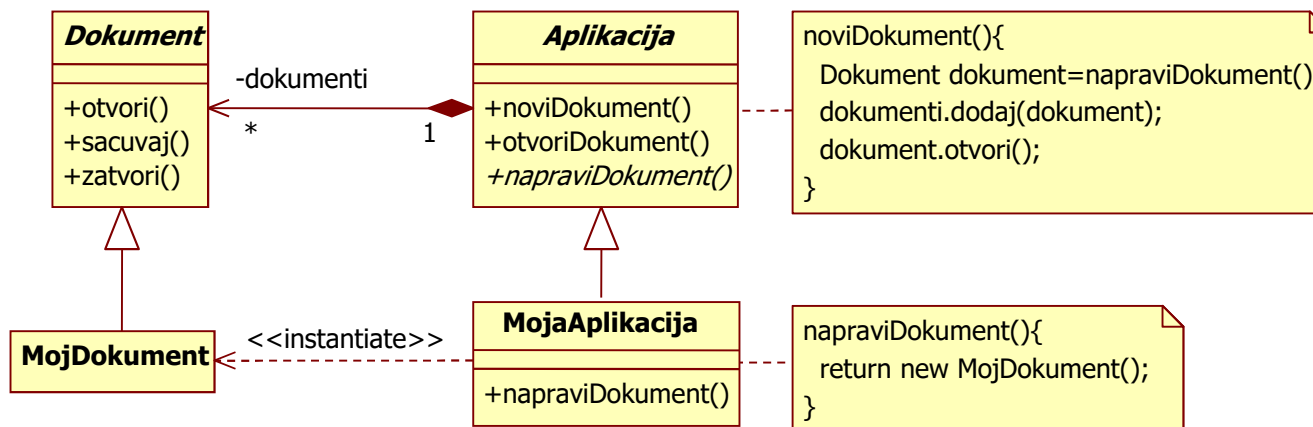


Fabrički metod (1)

- Ime i klasifikacija:
 - Fabrički (proizvodni) metod (*Factory Method*)
 - klasni uzorak kreiranja
- Namena:
 - uzorak dopušta klasi da delegira stvaranje objekata potklasi
 - definiše interfejs za stvaranje objekata, ali ostavlja potklasama da odluče čije objekte stvaraju
- Drugo ime:
 - Virtuelni konstruktor (*Virtual Constructor*)

Fabrički metod (2)

- Motivacija:
 - razmatra se radni okvir (*framework*) za aplikacije koje mogu da rade sa više dokumenata
 - ključne apstrakcije su apstraktne klase `Aplikacija` i `Dokument`
 - programeri konkretnih aplikacija treba da naprave potklase u kojima će da ostvare specifične realizacije

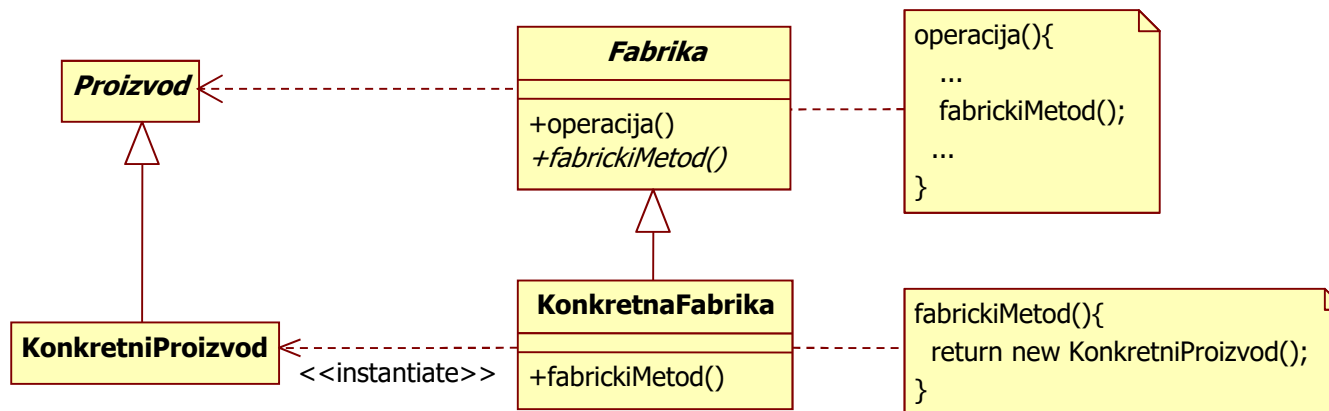


Fabrički metod (3)

- Motivacija (nastavak):
 - klasa aplikacije je odgovorna za upravljanje dokumentima
 - ona stvara dokument kada korisnik zahteva `Otvori` ili `Novi` iz menija
 - ona zna samo kada se stvara dokument
 - ona ne zna konkretan tip dokumenta čiji objekat treba stvoriti
 - radni okvir (*framework*)
 - skup opštih klijentskih klasa namenjenih izvođenju iz njih
 - korisnik projektuje serverske (klijentske pozivaju serverske)
 - radni okvir mora da predvidi stvaranje objekta klase dokumenta
 - on poznaje samo apstraktnu klasu dokumenta
 - konkretna potklasa aplikacije
 - nadjačava apstraktni metod klase aplikacije iz radnog okvira za stvaranje specifičnih dokumenata za datu aplikaciju
 - metod za stvaranje dokumenata je proizvodni (fabrički) metod (odgovoran je za proizvodnju objekata konkretnog tipa dokumenta)

Fabrički metod (4)

- Primenljivost: uzorak treba da se koristi kada:
 - klasa ne može da predvidi klasu objekata koje mora da stvara
 - klasa želi da njene potklase odrede tipove objekata čije će stvaranje ona zahtevati (ili će njoj biti zahtevano)
- Struktura:

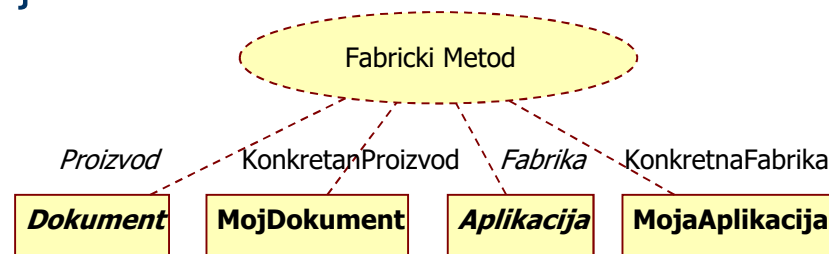


Fabrički metod (5)

- Učesnici:
 - *Proizvod* (klasa *Dokument*)
 - definiše interfejs objekata koje `fabrickiMetod()` stvara
 - *KonkretanProizvod* (klasa *MojDokument*)
 - implemetira interfejs *Proizvod*
 - *Fabrika* (klasa *Aplikacija*)
 - deklariše `fabrickiMetod()` koji vraća objekat tipa *Proizvod*
 - može da definiše podrazumevanu implementaciju za `fabrickiMetod()`, koji vraća podrazumevani objekat tipa *KonkretniProizvod*
 - iz neke operacije poziva `fabrickiMetod()` da stvori objekat *Proizvod*
 - *KonkretnaFabrika* klasa (klasa *MojaAplikacija*)
 - redefiniše `fabrickiMetod()` tako da vrati objekat *KonkretanProizvod*
- Saradnja:
 - *Fabrika* delegira svojim potklasama da definišu `fabrickiMetod()` tako da on vraća odgovarajući objekat *KonkretanProizvod*

Fabrički metod (6)

- UML notacija:



- Posledice:

- uzorak eliminiše potrebu da se klijentski kod vezuje za aplikativno-specifične klase
 - klijent radi preko ugovora *Proizvod*, pa može da radi sa bilo kakvim klasama `KonkretanProizvod`
 - za klijenta je i konkretan tip proizvoda pri stvaranju transparentan
- *Fabrički metod* daje potklasama mogućnost da obezbede proširenu verziju nekog objekta

Fabrički metod (7)

- Povezani uzorci:
 - *Fabrički metod* se obično poziva iz *Šablonskog metoda*
 - *Prototip* ne zahteva izvođenje potklase iz klase `Fabrika`
 - on zahteva `klon()` operaciju u klasi `Proizvod (Prototip)`, koju poziva `Fabrika`
 - *Fabrički metod* ne zahteva takvu operaciju
 - *Apstraktna fabrika* se često implementira pomoću *Fabričkog metoda*