

# Projektovanje softvera

Fabrički metod

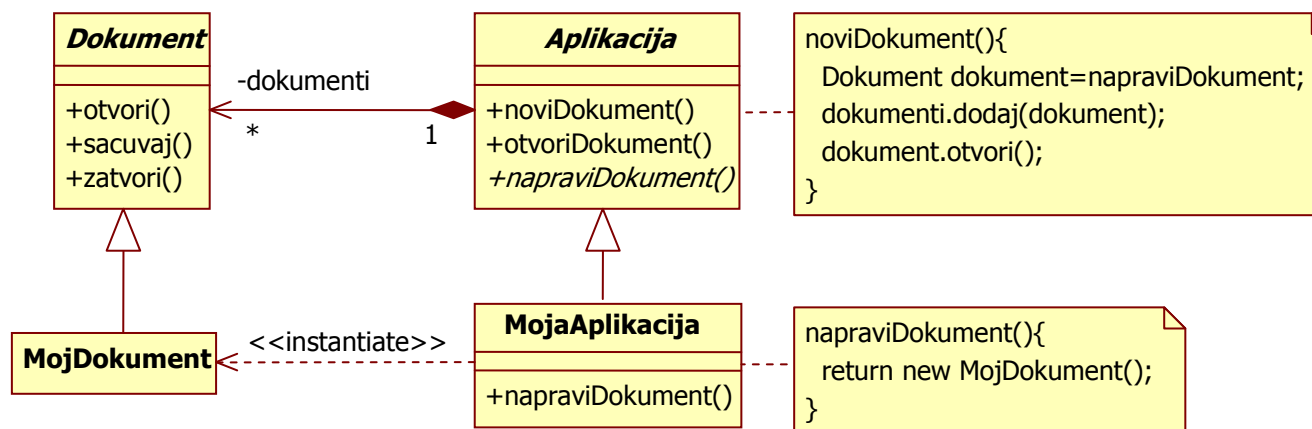


# Fabrički metod (1)

- Ime i klasifikacija:
  - Fabrički (proizvodni) metod (*Factory Method*)
    - klasni uzorak kreiranja
- Namena:
  - uzorak dopušta klasi da delegira stvaranje objekata potklasi
  - definiše interfejs za kreiranje objekata, ali ostavlja potklasama da odluče čije objekte kreiraju
- 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 `Aplikacija` i `Dokument`
  - obe klase su apstraktne i klijenti treba da naprave potklase da ostvare specifične realizacije

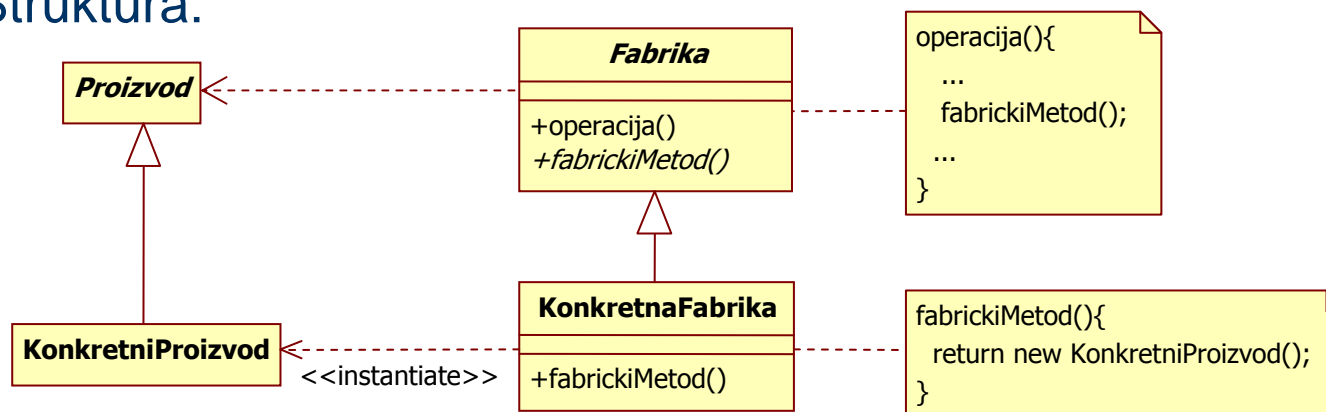


# Fabrički metod (3)

- Motivacija (nastavak):
  - klasa aplikacije je odgovorna za upravljanje dokumentima
    - ona kreira dokumente kada korisnik zahteva `Otvori` ili `Novi` iz menija
    - ona zna samo kada će kreirati dokument
    - ona ne zna konkretan tip dokumenta koji treba kreirati
  - radni okvir (*framework*)
    - skup opštih klijentskih klasa
    - korisnik projektuje serverske (klijentske pozivaju serverske)
    - radni okvir mora da stvara objekat klase dokumenta
    - on poznaje samo apstraktnu klasu dokumenta
  - konkretna potklasa aplikacije
    - redefiniše apstraktni metod klase aplikacije iz radnog okvira za kreiranje specifičnih dokumenata za datu aplikaciju
    - metod za kreiranje dokumenata je fabrički metod (odgovoran je za proizvodnju objekata konkretnog dokumenata)

# Fabrički metod (4)

- Primenljivost: uzorak treba koristiti kada:
  - klasa ne može da anticipira klasu objekata koje mora da kreira
  - klasa želi da njene potklase odrede objekte koje će ona kreirati
- Struktura:

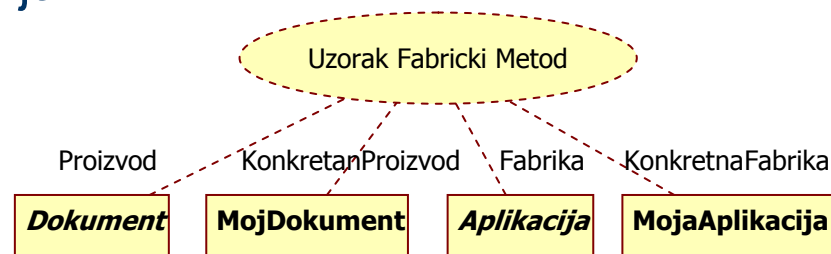


# Fabrički metod (5)

- Učesnici:
  - Proizvod (klasa Dokument)
    - definiše interfejs objekata koje `fabrickiMetod()` kreira
  - KonkretanProizvod (klasa MojDokument)
    - implemetira interfejs Proizvod
  - Fabrika (klasa Aplikacija)
    - deklarise `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 kreira 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 (npr. radni okvir) vezuje za aplikativno-specifične klase
    - klijent radi preko interfejsa `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*