

# Projektovanje softvera

Apstraktna fabrika



# Apstraktna fabrika (1)

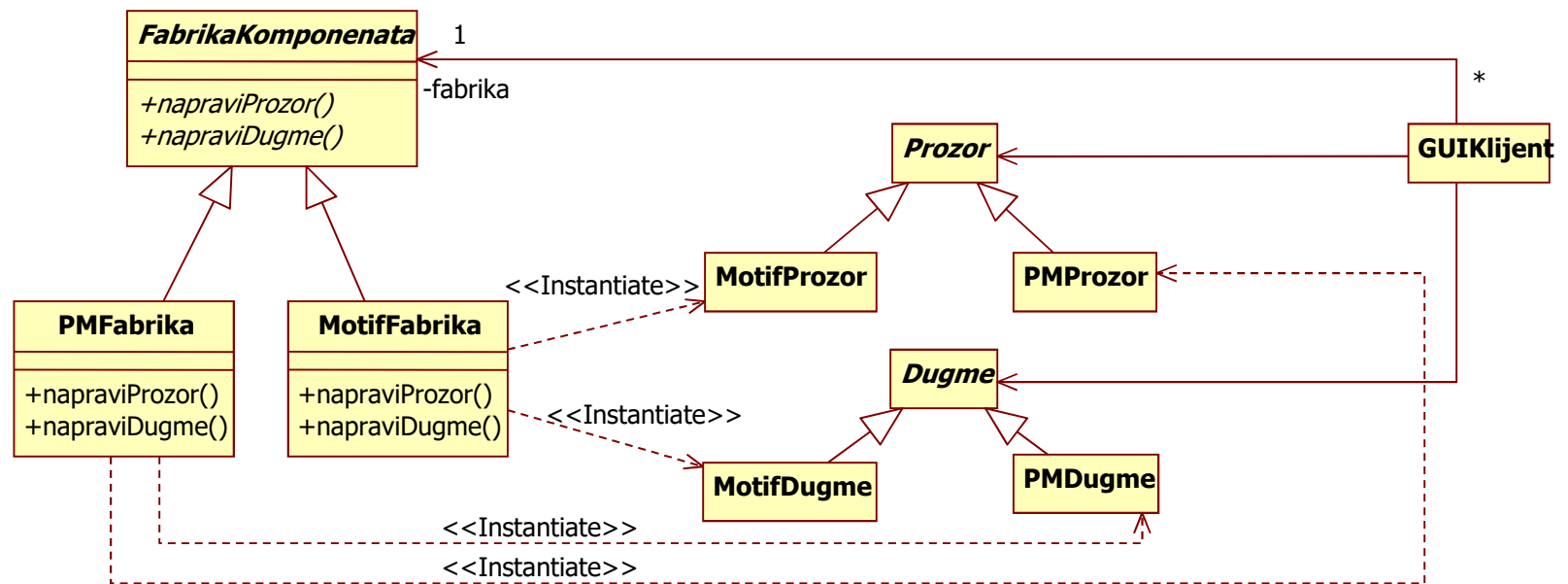
- Ime i klasifikacija:
  - *Apstraktna fabrika (eng. Abstract Factory)*
  - objektni uzorak stvaranja
- Namena:
  - obezbeđuje ugovor za konzistentnu proizvodnju različitih varijanti familije povezanih proizvoda
  - klijent nije svesan konkretnih klasa familije proizvoda
- Drugo ime:
  - *Kit*

# Apstraktna fabrika (2)

- Motivacija:
  - razmatra se klasna biblioteka za realizaciju korisničkog interfejsa
    - biblioteka podržava više standarda „izgleda-i-osećaja“ (*look-and-feel*)
    - primeri su *Motif* i *Presentation Manager* (PM)
  - različit izgled i osećaj definiše različite pojave komponenata (*widget*)
    - primeri su prozori (*windows*), klizači (*scroll bars*) i dugmad (*buttons*)
  - aplikacija treba da bude portabilna između standarda izgleda i osećaja
    - stvari izgleda i osećaja ne treba da se fiksno kodiraju
    - ne treba da se stvaraju objekti specifičnog izgleda svuda po aplikaciji
  - rešenje problema - definiše se apstraktna klasa `FabrikaKomponentata`
    - klasa predstavlja apstrakciju fabrike familije komponenata
    - klasa definiše ugovor za stvaranje svih vrsta komponenata od interesa
    - za svaku vrstu komponente postoji apstraktan metod za stvaranje

# Apstraktna fabrika (3)

- Motivacija (nastavak):



# Apstraktna fabrika (4)

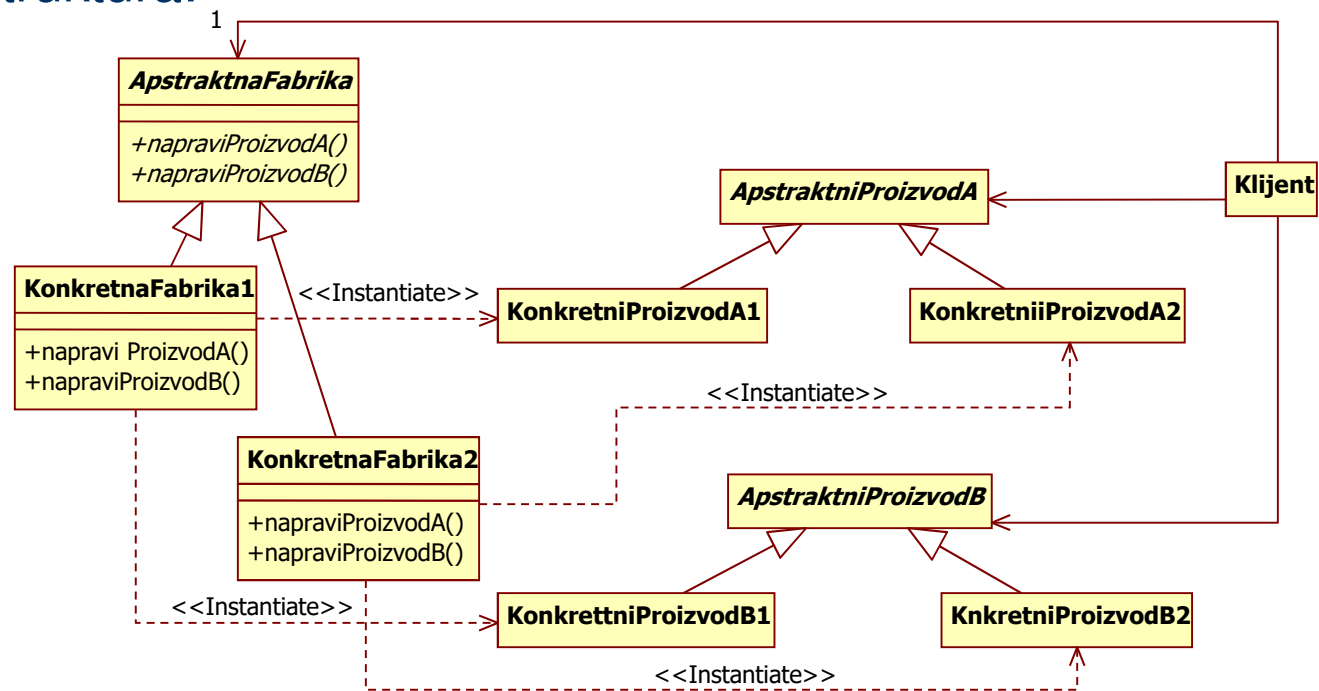
- Motivacija (nastavak):
  - potrebna je po jedna izvedena klasa stvarne fabrike za svaki standard izgleda i osećaja
    - one implementiraju operacije stvaranja koje je propisala apstraktna fabrika
    - operacije stvaranja poštuju specifičan standard izgleda i osećaja
  - potrebna je po jedna apstraktna klasa za svaku vrstu komponenata
    - konkretne potklase implementiraju komponentu za jedan standard izgleda
  - klijenti metodima apstraktne fabrike proizvode objekte komponenata
    - klijenti nisu svesni konkretne fabrike koju koriste za stvaranje komponenata
      - ostaju nezavisni od izgleda i osećaja
    - klijenti moraju da poštuju ugovor koji definiše apstraktna fabrika
    - samo pri stvaranju objekta konkretne fabrike klijent je svesan njegovog tipa
  - osim apstraktne fabrike klijenti „vide“ i apstraktne komponente (neopterećene specifičnostima izgleda i osećaja)

# Apstraktna fabrika (5)

- Primenljivost: uzorak treba da se koristi kada
  - sistem treba da bude konfigurisan jednom od više varijanti familije proizvoda
  - klijent treba da bude nezavisan od načina
    - stvaranja proizvoda
    - predstavljanja konkretnih proizvoda
  - potrebno je forsirati da proizvodi iz jedne varijante familije budu korišćeni zajedno
  - potrebno je ponuditi klasnu biblioteku proizvoda otkrivajući samo ugovore, ne implementacije

# Apstraktna fabrika (6)

- Struktura:



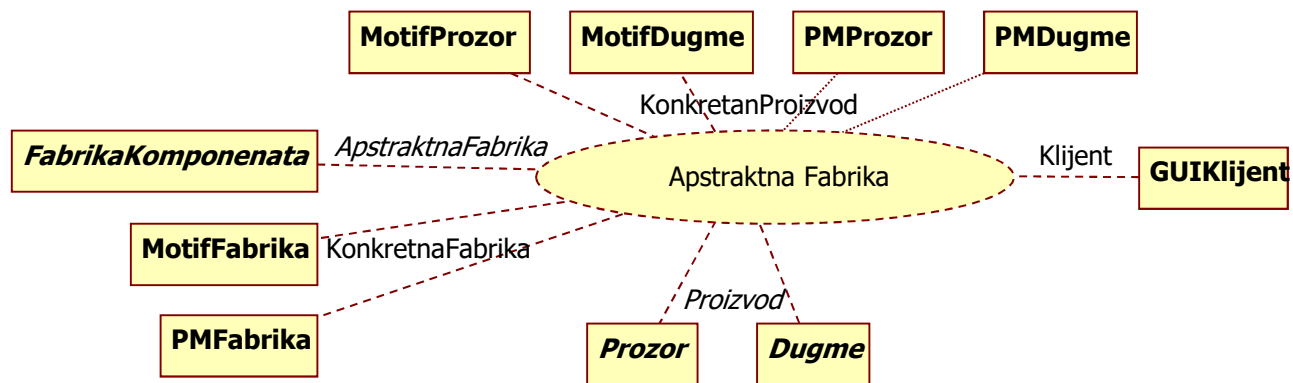
# Apstraktna fabrika (7)

- Učesnici:
  - *ApstraktnaFabrika* (klasa *FabrikaKomponenata*)
    - deklarise ugovor za operacije koje stvaraju objekte apstraktnih proizvoda
  - *KonkretnaFabrika* (klase *PMFabrika* i *MotifFabrika*)
    - implementira operacije koje stvaraju objekte konkretnih proizvoda
  - *ApstraktniProizvodX* (klase *Prozor* i *Dugme*)
    - deklarise ugovor za određenu vrstu proizvoda
  - *KonkretanProizvod* (klase *PMProzor*, *PMDugme*, *MotifProzor*, *MotifDugme*)
    - implementira ugovor apstraktnog proizvoda
    - definiše proizvod koji će biti stvaran pomoću odgovarajuće konkretne fabrike
  - *Klijent* (klasa *GUIKlijent*)
    - koristi samo ugovore deklarisanane pomoću apstraktne fabrike i apstraktnog proizvoda
- Saradnja:
  - klijent uobičajeno stvara po jedan primerak konkretne fabrike
  - klijent zahteva od apstraktne fabrike stvaranje proizvoda iz familije proizvoda
  - apstraktna fabrika odlaže stvaranje proizvoda do njenih potklasa – konkretnih fabrika
  - klijent koristi konkretne proizvode preko njihovih apstraktnih ugovora



# Apstraktna fabrika (8)

- UML notacija:



# Apstraktna fabrika (9)

- Posledice:
  - Apstrahovanje konkretnih klasa proizvoda
    - klijent manipulše proizvodima kroz njihove apstraktne ugovore
    - klijent ne koristi imena konkretnih klasa proizvoda (čak ni za njihovo stvaranje)
  - Olakšava izmenu familije proizvoda
    - klasa konkretne fabrike se pojavljuje samo jednom u aplikaciji – tamo gde se stvara njen objekat
    - aplikacija može da koristi različite konfiguracije proizvoda menjanjem konkretne fabrike
    - pošto apstraktna fabrika stvara kompletnu familiju, cela familija se menja odjednom
  - Unapređuje konzistenciju među proizvodima
    - aplikacija koristi objekte proizvoda samo iz jedne familije u jednom trenutku
  - Problem: podrška novoj vrsti proizvoda nije jednostavna
    - razlog je taj što apstraktna fabrika fiksira skup proizvoda koji mogu da se stvaraju
    - podrška novog proizvoda zahteva proširenje apstraktne fabrike i svih potklasa
- Povezani uzorci:
  - *Apstraktna fabrika* se često implementira sa *Fabričkim metodom* ili koristeći *Prototip*
    - konkretne fabrike mogu da realizuju fabričke metode koji konstruišu proizvode (uobičajeno)
    - „apstraktna“ fabrika može da bude i konkretna klasa parametrizovana prototipskim proizvodima čije klonove stvara
  - konkretna fabrika je često *Unikat*
  - konkretna fabrika predstavlja konkretnu *Strategiju* stvaranja familije proizvoda