

Projektovanje softvera

Apstraktna fabrika



Apstraktna fabrika (1)

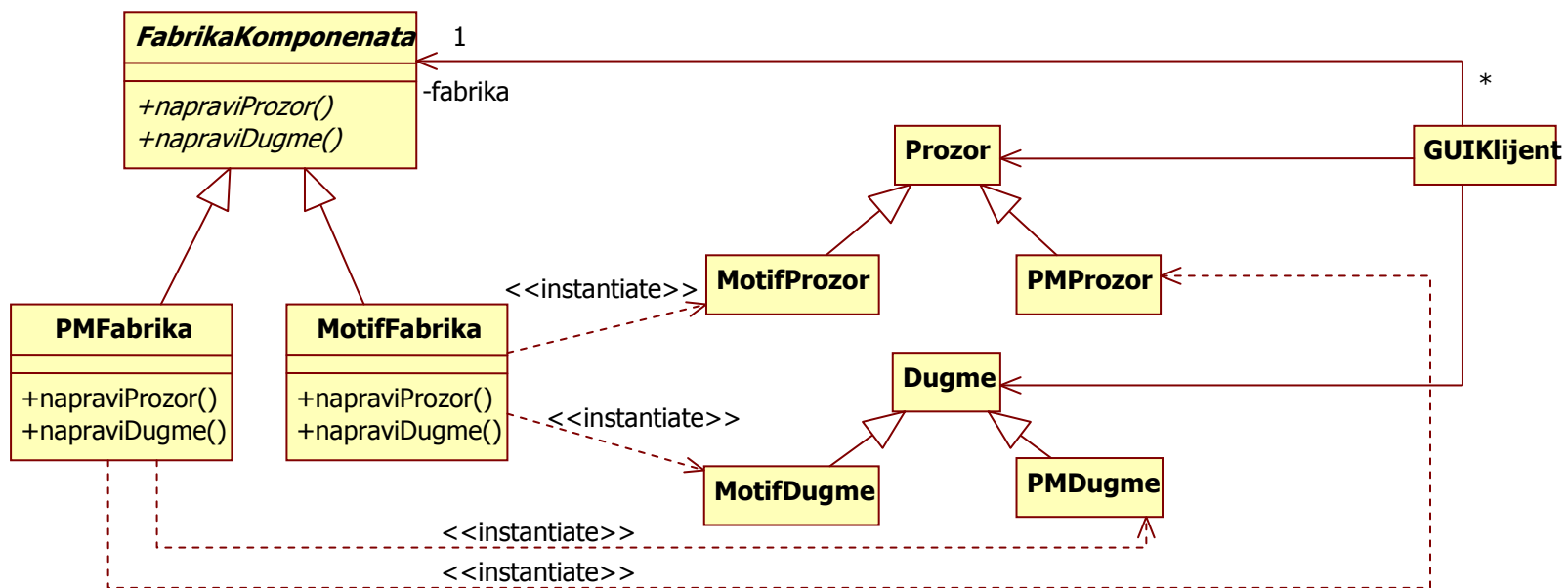
- Ime i klasifikacija:
 - *Apstraktna fabrika (eng. Abstract Factory)*
 - objektni uzorak kreiranja
- Namena:
 - obezbeđuje interfejs za kreiranje različitih familija povezanih objekata (proizvoda)
 - klijent nije svesan konkretnih klasa te familije proizvoda
- Drugo ime:
 - *Kit*

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
 - ne smeju se fiksno kodirati stvari izgleda i osećaja
 - ne treba kreirati objekte specifičnog izgleda svuda po aplikaciji
 - rešenje problema - definiše se apstraktna klasa `FabrikaKomponentata`
 - klasa predstavlja apstrakciju fabrike familije komponenata
 - klasa definiše interfejs za kreiranje svih vrsta komponenata od interesa
 - za svaku vrstu komponente postoji apstraktan metod za kreiranje

Apstraktna fabrika (3)

- Motivacija (nastavak):



Apstraktna fabrika (4)

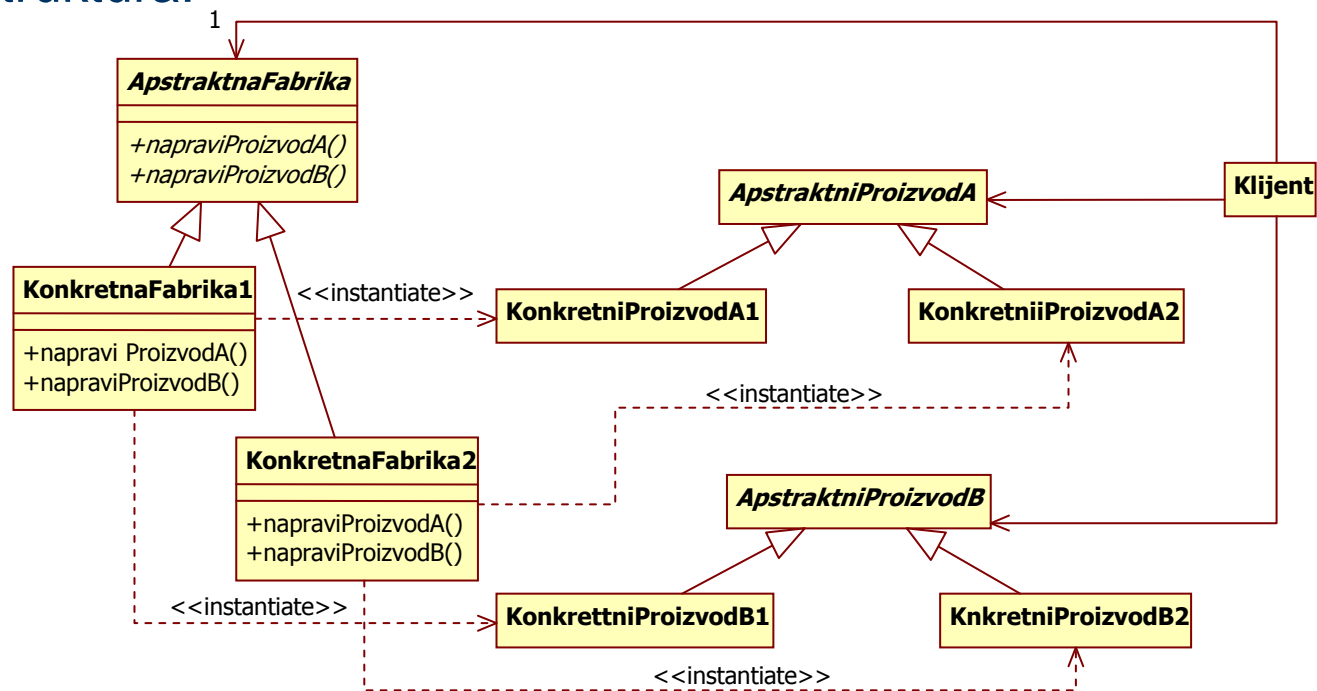
- Motivacija (nastavak):
 - potrebna je po jedna izvedena klasa stvarne fabrike za svaki standard izgleda i osećaja
 - one implementiraju operacije kreiranja koje je propisala apstraktna fabrika
 - operacije kreiranja poštuju specifičan standard izgleda i osećaja
 - potrebna je po jedna apstraktna klasa 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 proizvodnju komponenata
 - ostaju nezavisni od izgleda i osećaja
 - klijenti moraju da poštuju interfejs koji definiše apstraktna fabrika
 - samo pri kreiranju 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 koristiti kada
 - sistem treba da bude konfigurisan jednom od više familija proizvoda
 - sistem treba da bude nezavisan od načina
 - kreiranja proizvoda
 - predstavljanja proizvoda
 - potrebno je forsirati da proizvodi iz familije budu korišćeni zajedno
 - potrebno je ponuditi klasnu biblioteku proizvoda otkrivajući samo interfejse, ne implementacije

Apstraktna fabrika (6)

- Struktura:

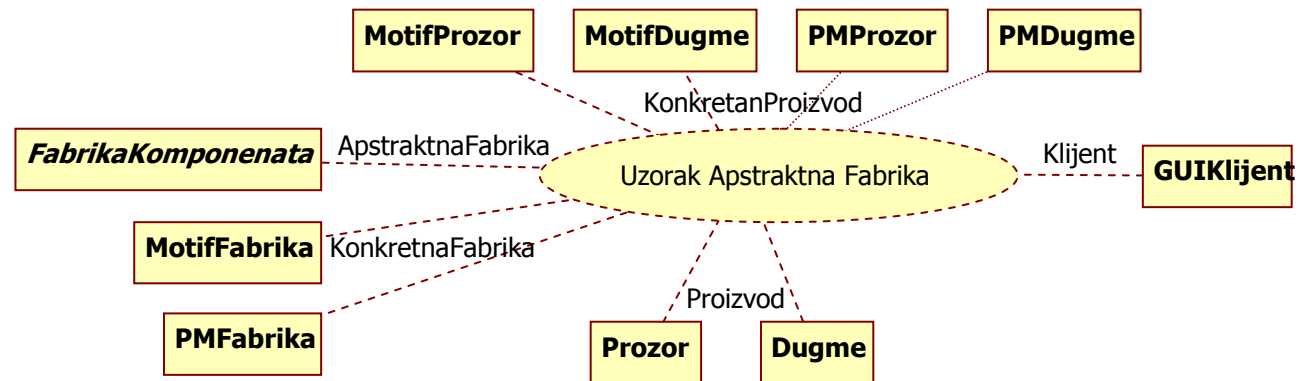


Apstraktna fabrika (7)

- Učesnici:
 - ApstraktnaFabrika (klasa FabrikaKomponenata)
 - deklarise interfejs za operacije koje kreiraju objekte apstraktnih proizvoda
 - KonkretnaFabrika (klase PMFabrika i MotifFabrika)
 - implementira operacije koje kreiraju objekte konkretnih proizvoda
 - ApstraktniProizvodX (klase Prozor i Dugme)
 - deklarise interfejs za određeni tip proizvoda
 - KonkretanProizvod (klase PMProzor, PMDugme, MotifProzor, MotifDugme)
 - implementira interfejs apstraktnog proizvoda
 - definiše proizvod koji će biti kreiran pomoću odgovarajuće konkretne fabrike
 - Klijent (klasa GUIKlijent)
 - koristi samo interfejse deklarisanе pomoću apstraktne fabrike i apstraktnog proizvoda
- Saradnja:
 - apstraktna fabrika odlaže kreiranje proizvoda do njenih potklasa – konkretnih fabrika
 - uobičajeno treba kreirati samo po jednu instancu konkretne fabrike

Apstraktna fabrika (8)

- UML notacija:



Apstraktna fabrika (9)

- Posledice:
 - Izolacija konkretnih klasa proizvoda
 - klijent manipulira proizvodima kroz njihove apstraktne interfejsne
 - 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 pravi njen objekat
 - aplikacija može da koristi različite konfiguracije proizvoda menjanjem konkretne fabrike
 - pošto apstraktna fabrika kreira 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 se mogu kreirati
 - podrška novog proizvoda zahteva proširenje apstraktne fabrike i svih potklasa
- Povezani uzorci:
 - *Apstraktna fabrika* se često implementira sa *Proizvodnim metodom* i/ili koristeći *Prototip*
 - konkretne fabrike mogu realizovati proizvodne metode koje konstruišu proizvode (uobičajeno)
 - konkretne fabrike se mogu i paramerizovati prototipskim proizvodima, te praviti njihove kolonove
 - konkretna fabrika je često *Unikat*
 - konkretna fabrika predstavlja konkretnu *Strategiju* kreiranja familije proizvoda