

Projektovanje softvera

Dekorater



Dekorater (1)

- Ime i klasifikacija:
 - Dekorater (engl. *Decorator*) – objektni uzorak strukture
- Namena:
 - dinamički dodaje odgovornosti (mogućnosti) nekom objektu
 - predstavlja fleksibilnu alternativu izvođenju za proširivanje funkcionalnosti
- Druga imena:
 - Dopuna, Omotač (engl. *Wrapper*)

Dekorater (2)

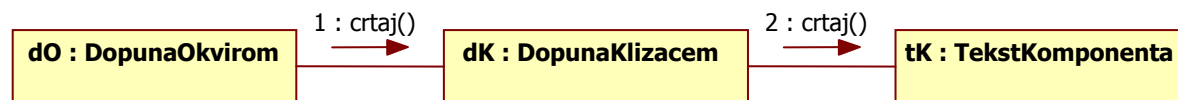
- Motivacija:
 - alati za GUI podržavaju dodavanje “ukrasa” na razne komponente
 - klizači, okviri sa raznim efektima,...
 - jedan način za dodavanje mogućnosti – nasleđivanje sa proširivanjem
 - nasleđivanje je nefleksibilno, jer bi se ukrasi definisali statički
 - opasnost od eksplozije broja izvedenih klasa kada se kombinuju ukrasi
 - ukrasi se ne mogu dinamički kombinovati na osnovnoj komponenti
 - fleksibilniji pristup je da se komponenta obuhvati drugom komponentom
 - komponenta koja obuhvata osnovnu dodaje ukras (okvir, klizače)
 - proširenja mogu biti specifična stanja ili ponašanja
 - odgovornosti se dodaju pojedinom objektu, a ne celoj klasi
 - odgovornosti se mogu menjati dinamički

Dekorater (3)

- Motivacija (nastavak):
 - komponenta koja obuhvata osnovnu se naziva
 - dekoraterom, omotačem ili dopunom
 - dekorater prosleđuje zahteve klijenta obuhvaćenoj komponenti i može da obavi dodatne akcije
 - kao što je crtanje okvira ili dodavanje klizača za pomeranje
 - dekorater nasleđuje interfejs komponente koju ukrašava
 - prisustvo dekoratera je transparentno za klijente komponente
 - transparentnost omogućava neograničen broj dodataka uvedenih posebnim dekoraterima

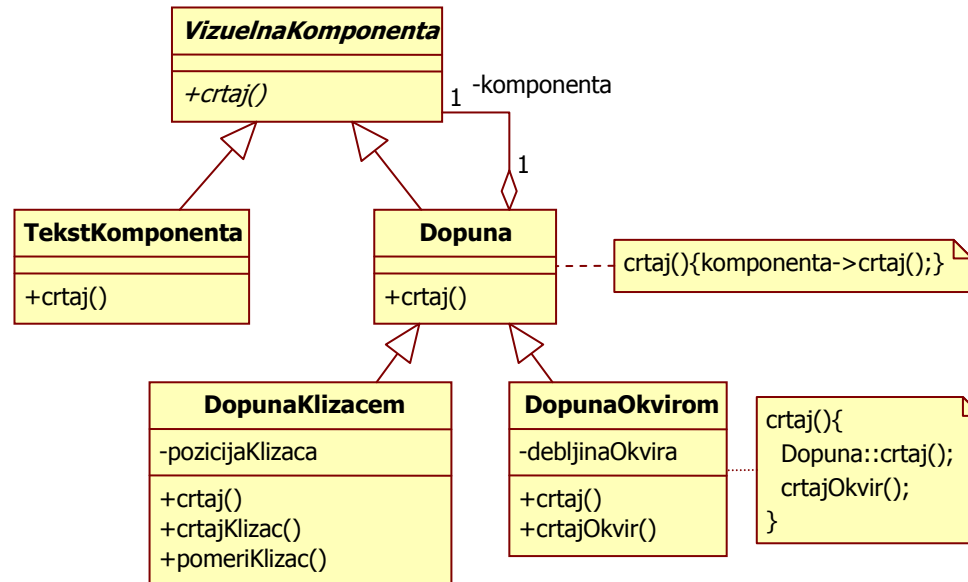
Dekorater (4)

- Motivacija (nastavak):
 - primer:
 - `TekstKomponenta` prikazuje tekst u prozoru i nema klizace za H/V pomeranje sadržaja, ni okvir
 - ako su potrebni klizači
 - dodajemo objekat klase `DopunaKlizacem` koji ih dodaje
 - ako se želi i okvir
 - dodajemo i objekat klase `DopunaOkvirom` koji crta okvir
 - sledeći dijagram prikazuje objektu kompoziciju i interakciju:



Dekorater (5)

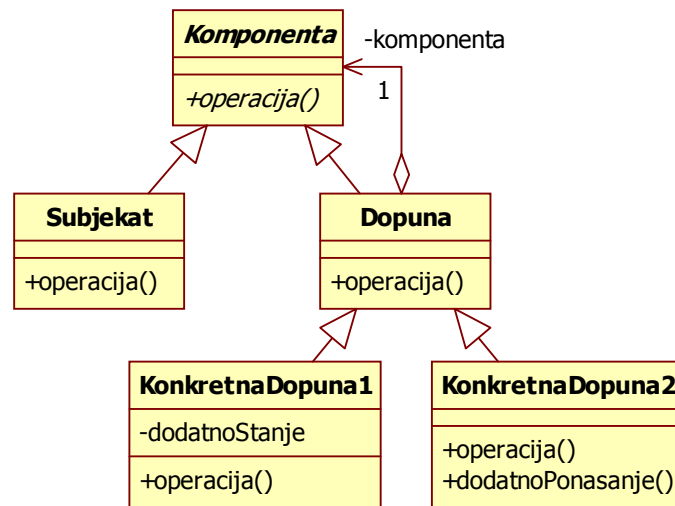
- Motivacija (nastavak):
 - sledeći dijagram prikazuje opisanu klasnu strukturu:



- Primenljivost:
 - kada je potrebno dinamički dodavati odgovornosti objektima na transparentan način
 - kada proširenje izvođenjem nije praktično (kada je moguć veliki broj nezavisnih proširenja, pri čemu se ona mogu kombinovati, što vodi eksploziji broja klasa)

Dekorater (6)

- **Struktura:**



- **Učesnici:**

- Komponenta (klasa VizuelnaKomponenta)
 - definiše interfejs za objekte kojima se mogu dinamički dodavati odgovornosti
- Subjekat (klasa TekstKomponenta)
 - definiše objekat kojem se dodaju odgovornosti
- Dopuna (klasa Dopuna)
 - održava referencu na objekat klase Komponenta i nasleđuje interfejs klase Komponenta
- KonkretnaDopunaX (klase DopunaKlizacem, DopunaOkvirom)
 - dodaje odgovornost objektu tipa Komponenta

- **Saradnja:**

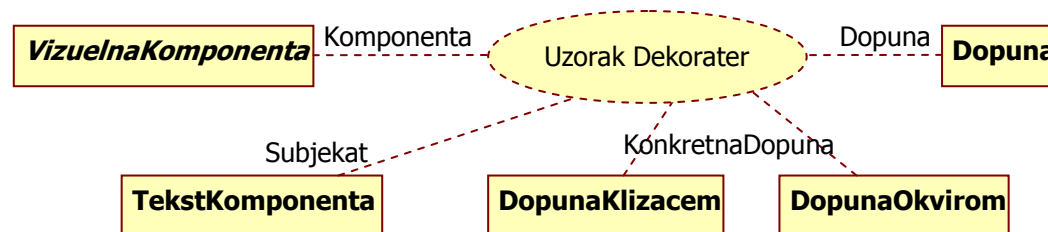
- objekat KonkretnaDopunaX
 - prosleđuje zahteve obuhvaćenom objektu (tipa Komponenta)
 - izvršava dodatne operacije pre i/ili posle prosleđivanja zahteva

Dekorater (7)

- Posledice
 - prednosti
 - veća fleksibilnost od statičkog nasleđivanja
 - dinamičko dodavanje odgovornosti
 - izbegavanje bogato parametrizovanih klasa visoko u hijerarhiji
 - izbegavanje eksplozije potklasa koje kombinuju ukrase
 - mane
 - identitet dekoratera i dekorisanog objekta su različiti
 - objekti se ne razlikuju po klasi već po načinu povezivanja
 - potencijalan problem kod razumevanja i održavanja sistema
 - nasleđeni atributi komponente, ako postoje (duplikat)
 - dobro je da apstraktna komponenta nema attribute

Dekorater (8)

- UML notacija:



- Povezani uzorci:

- *Dekorater* zadržava interfejs, a *Adapter* menja interfejs
- *Kompozicija* ima sličnu strukturu
 - *Dekorater* je po strukturi degenerisana *Kompozicija* (*Sastav*, *Sklop*)
 - objekat *Dopuna* sadrži samo jednu komponentu, a objekat *Kompozicija* više
 - po nameni je *Dekorater* sasvim različit, jer dodaje odgovornosti i nije namenjen grupisanju
- *Dekorater* i *Strategija* su alternative za promenu ponašanja objekta
 - *Dekorater* menja spoljašnjost objekta, a *Strategija* unutrašnjost