

## Strukturalni design paterni

### Fasadni patern

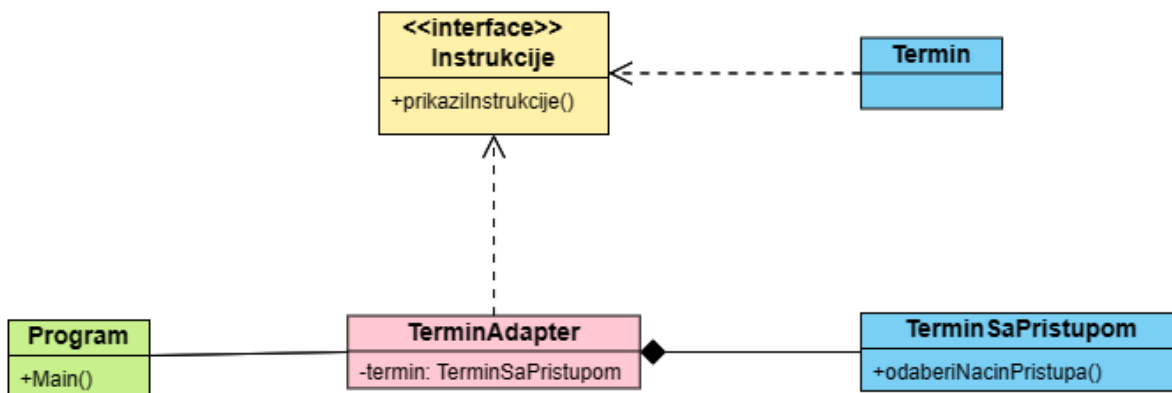
Fasadni patern koristimo kada imamo ograničen ali jednostavan interfejs ka kompleksnom podsistemu. Koristimo ga kada želimo struktuirati podsistem u slojeve.

Ovaj patern možemo iskoristiti jer naši tutori i studenti imaju drugačije privilegije pa tako i view-e.

### Adapter patern

Adapter patern se koristi kada je potreban drugačiji interfejs već postojeće klase, a ne želimo je mijenjati.

Ovaj patern možemo iskoristiti za zakazivanje termina instrukcija. Naš sistem možemo dodatno unaprijediti, tako što bi student osim mogućnosti da odabere termin instrukcija mogli da unese i način pristupa instrukcija (uživo ili online). Time bi iskoristili adapter patern proširujući funkcionalnosti već postojeće klase.



### Dekorater patern

Ovaj patern možemo iskoristiti da unaprijedimo funkcionalnost pretrage tutora po kategorijama.

Možemo dodati nove kriterije sortiranja ili prilagoditi postojeće bez mijenjanja osnovne funkcionalnosti pretrage lijekova. Također, dekoratori se mogu kombinovati kako bismo pružili još složenije mogućnosti sortiranja, prilagođene potrebama studenata.

### Bridge patern

Ovaj pattern smo upotrijebili prilikom kreiranja naloga na našem sistemu. Na naš sistem se možete prijaviti kao student ili kao tutor. Oba ova načina su ustvari iste operacije ali se primjenjuju na 2 različite klase, jedna je Student a druga Tutor.

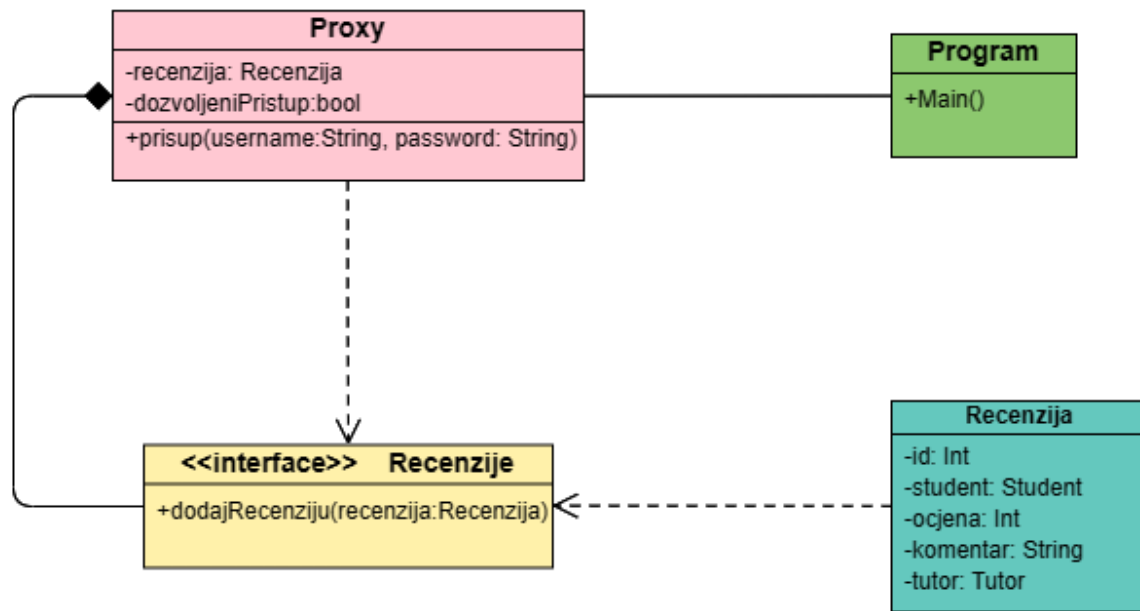
## Kompozitni patern

Ovaj patern možemo koristiti u klasi Tutori kako bi omogućili grupisanje tutora . Na primjer možemo ih grupisati po predmetima ili gradovima u kojim predaju.

## Proxy patern

Ovaj patern koristimo u situacijama kada je potrebno kontrolirati pristup objektu i obaviti dodatne provjere ili funkcionalnosti.

U ovom sistemu se može primijeniti na klasu "Recenzije" kako bi se osiguralo da samo student koji su imali čas kod tog tutora mogu dodavati recenzije.



## Flyweight patern

Ovaj patern koristimo kada imamo više objekata sličnih atributa i želimo optimizirati upotrebu memorije.

Umjesto stvaranja novih instanci atributa za svaki pojedinačni predmet, koristimo već postojeće objekte koji dijele iste vrijednosti za ime predmeta i cijenu.