

Strukturalni paterni

1. *Adapter* patern služi da se postojeći objekat prilagodi za korištenje na neki novi način u odnosu na postojeći rad, bez mijenjanja same definicije objekta. Obzirom na svrhu naše aplikacije, svi mogući budući dodaci ni na koji način neće moći koristiti trenutne funkcionalnosti bez narušavanja definicija klasa.

Zaključujemo da navedeni patern nije primjenjiv u našem projektu.

2. *Fasadni* patern služi kako bi se klijentima pojednostavilo korištenje kompleksnih sistema. Klijenti vide samo fasadu, odnosno krajnji izgled objekta, dok je njegova unutrašnja struktura skrivena.

Najkompleksnija struktura koju koristimo je Termin gdje hipotetički možemo iskoristiti fasadni patern, obzirom na sve provjere koje će se izvršavati u sistemu prilikom kreiranja termina predavanja ili vježbi o kojima korisnik ne treba voditi računa.

3. *Decorator* patern služi za omogućavanja različitih nadogradnji objektima koji svi u osnovi predstavljaju jednu vrstu objekta (odnosno, koji imaju istu osnovu).

Navedeni patern bismo hipotetički mogli primijeniti na klasu termin, ukoliko bismo iz nje izveli dvije vrste termina, termini za predavanja i termini za tutorijale. Međutim, u našem sistemu ne postoji razlika između te dvije vrste termina te navedeni patern neće biti iskorišten.

4. *Bridge* patern služi kako bi se apstrakcija nekog objekta odvojila od njegove implementacije. Navedeni patern bi također mogao biti iskorišten u slučaju izvedenih klasa iz klase Termin te da se implementacije termina predavanja i tutorijala razlikuju.

5. *Comoposite* patern spada u skupinu strukturalnih paterni i koristi se za kreiranje hijerarhije objekata, tačnije za objekte koji imaju različite implementacije metoda kojima se pristupa na isti način. U našem projektu nema vidljive primjene ovog paterni.

6. *Proxy* patern je patern koji osigurava objekte od pogrešne ili zlonamjerne upotrebe i omogućava kontrolu pristupa. Ukoliko bismo našu aplikaciju nadogradili tako da i asistenti imaju mogućnost dodavanja termina tutorijala, a onemogućili im dodavanje termina predavanja, tada bismo shodno tome dali li je osoba profesor ili asistent dali odgovarajuće pravo pristupa na dodavanje termina.

U našoj aplikaciji Proxy patern je iskorišten na način da je klasi Student, izvedenoj iz klase Korisnik, onemogućeno dodavanje tutorijala u bazu podataka, dok je ta metoda dozvoljena klasi Profesor, također izvedenoj iz klase Korisnik.

Pomenuti patern je iskorišten i pri optimizaciji studentskog rasporeda, tj. pri optimizaciji i uređivanju rasporeda bit će uzeti u obzir samo predmeti dodijeljeni studentu.

7. *Flyweight* patern koristi se kako bi se onemogućilo bespotrebno stvaranje velikog broja instanci objekata koji svi u suštini predstavljaju jedan objekat. Samo ukoliko postoji potreba za kreiranjem specifičnog objekta sa jedinstvenim karakteristikama (tzv. specifično stanje), vrši se njegova instantacija, dok se u svim ostalim slučajevima koristi postojeća opća instanca objekta (tzv. bezlično stanje).

Pomenuti patern ćemo primijeniti u našoj aplikaciji na način da ne pravimo duboke kopije sala koje se pridružuju terminima pri kreiranju termina za predavanja/vježbi.