

Paterni ponašanja

Strategy patern:

Doktor bi mogao odabrati neki algoritam za sortiranje medicinskih kartona preko klase Nalaz. Treba dodati klase koje implementiraju algoritme (IStrategy interfejs) u kojima je definisana metoda sortiranja koja je ista za sve algoritme. IStrategy interfejs nasljeđuju sve klase koje implementiraju različita sortiranja. Na ovaj način, doktor bi pristupao sortiranim medicinskim kartonima.

Observer patern:

U našem projektu, ovaj patern smo iskoristiti kada korisnik odabere slobodan termin za preglet te taj zahtjev biva odobren. Klasa Raspored će implementirati interfejs IObserver, na poziv funkcije postaviStatusTermina će dodati odobreni termin u bazu. Korisnik bi dobio notifikaciju da je uspješno zakazao termin.

Ovaj patern mi također mogli implementirati kada bi dodali mogućnost da se korisnici preplate da dobijaju notifikacije o svim dešavanjima na poliklinici (promjena osoblja, cijene, usluge, radno vrijeme). To bi uradili tako što bi dodali interfejs IObserver koji će imati metodu posaljiObavijestjenje. U klasi Pacijent nam treba lista korisnika koji su se preplatili za novosti i metoda obavijestiPacijente.

State patern:

Ovaj patern bi mogli iskoristiti kada bi na osnovu trenutnog broja zakazanih (i odobrenih) termina od strane korisnika formirali određena stanja. Svaki korisnik sa jednim aktivnim terminom ne ostvaruje nikakav *popust/privilegije* i nalazi se na stanju Pacijent1, dok za svaka dva termina, pacijent prelazi u stanje Pacijent2, Pacijent3... Ove izmjene bi se izvršavale automatski u zavisnosti od broja trenutno zakazanih termina.

Iterator patern:

Ovaj patern možemo iskoristiti za prolazak kroz listu doktora poliklinike unutar klase Doktor. Da bi se implementirao ovaj patern, potrebno je kreirati klasu Iterator koja će implementirati interfejs iterator. Pomenuta klasa bi imala atribut listu tipa Doktor i indeks trenutnog elementa i metode dajSljedećeg() i dajTrenutnog().

TemplateMethod patern:

Ovaj patern možemo iskoristiti da različite vrste pregleda daju različite rezultate nekih metoda, a način unosa u karton je isti. Bazna klasa Pregled ima default implementaciju za unos u karton, default implementaciju opisa pregleda i druge metode, a različite klase pregleda imaju različite implementacije za neke od tih metoda, nikad za sve.