

Kreacijski paterni

U grupu kreacijskih paterna se ubrajaju paterni: Singleton, Prototype, Factory Method, Abstract Factory i Builder.

Singleton patern

Uloga Singleton paterna je da osigura da se klasa može instancirati samo jednom i da osigura globalni pristup kreiranoj instanci klase.

Ovaj patern bi se mogao iskoristiti kod klase 'Direktor' (naša aplikacija ne posjeduje ovu klasu) jer je direktor jedna osoba koja se ne mijenja određeni vremenski period a ima interakciju sa svim podacima iz baze.

Prototype patern

Uloga Prototype paterna je da kreira nove objekte klonirajući jednu od postojećih prototip instanci (postojeći objekat).

Ovaj patern ćemo iskoristiti pri kreiranju rasporeda od strane doktora. Doktori će dobiti isti početni raspored, a nakon toga će oni popunjavati raspored u odnosu na svoju dostupnost tokom radnog vremena. Također, ovaj patern se može koristiti i pri kreiranju/kloniranju smjena unutar poliklinike.

Factory method patern

Uloga Factory Method paterna je da omogući kreiranje objekata na način da podklase odluče koju klasu instancirati. Različite podklase mogu na različite načine implementirati interfejs. Factory Method instancira odgovarajuću podklasu (izvedenu klasu) preko posebne metode na osnovu informacije od strane klijenta ili na osnovu tekućeg stanja.

Ovaj patern se može iskoristiti unutar izdavanja recepta pacijentu. Naime, mogu se napraviti dvije skupine lijekova: jeftini i skupi. Svaka od navedenih *fabrika* bi podržavala razne metode (ReceptProtivGlavobolje, ReceptZaKožneBolesti i sl.). Fabrika bi pri pozivu ove metode vratila SkupiReceptProtivGlavobolje recept čija bi metoda *dajLijek* vratila neki skuplji lijek, metoda *Upotreba* vratila jedan način upotrebe i sl. Važno je da postoji jedna metoda za kreiranje recepata, a da su metode za davanje lijeka i upotrebe iste za različite recepte.

Abstract Factory patern

Abstract Factory patern omogućava da se kreiraju familije povezanih objekata/produkata. Na osnovu apstraktne familije produkata kreiraju se konkretne fabrike (factories) produkata različitih tipova i različitih kombinacija. Patern odvaja definiciju (klase) produkata od klijenta.

Zbog toga se familije produkata mogu jednostavno izmjenjivati ili ažurirati bez narušavanja strukture klijenta.

Ovaj patern bi bio iskoristiv kada bismo napravili klasu ProizvođačiLijekova koja ima listu svih proizvođača lijekova koji bi sarađivali sa našom poliklinikom, klasu MedicinskaOprema koja ima listu medicinske opreme i sl. To znači da bi nam trebala apstraktna klasa Saradnici sa nasljednicama ProizvođačiLijekova, MedicinskaOprema i sl.

Builder patern

Uloga Builder patern je odvajanje specifikacije kompleksnih objekata od njihove stvarne konstrukcije. Isti konstrukcijski proces može kreirati različite reprezentacije.

Patern bi mogli iskoristiti kada bi imali polikniku u više gradova (ne samo jednu u jednom gradu). Pri ulasku na našu stranicu, korisnik bi mogao da odabere grad koji mu najviše odgovara i na osnovu odabira, KojiGradBuilder klasa, koja implementira IBuilder interfejs, bi generisala početnu stranicu sa odjelima i uslugama u odabranom gradu.