

# Kreacijski paterni

## 1. Singleton pattern

Ovaj pattern će biti primijenjen u ovom projektu. Singleton pattern se može iskoristiti kako bi instancirali objekat koji pravi konekciju sa bazom podataka. Ovaj objekat bi trebao biti dostupan na globalnom nivou svim potencijalnim funkcijama i objektima koji traže pristup bazi podataka. Nema smisla otvarati više od jedne konekcije prema bazi podataka od jednom, pa zato i nema potrebe za više od jedne instance ove klase.

## 2. Prototype pattern

Ovaj pattern bi se mogao koristiti uz builder pattern. Korisnik bi mogao sa spašava prethodno kreirane filtere i da ih klonira i izmjenjuje. Klonirani filter bi imao ista svojstva kao i originalni filter i korisnik bi mogao da nadodaje nove ili mijenja i briše stare filtere bez ikakve interakcije sa originalnim filterom.

## 3. Factory method pattern

Ovaj pattern bi mogli iskoristiti u projektu pri slanju notifikacija. Notifikacije bi mogle imati vrlo slične sadržaje sam elementima koji se mijenjaju. Recimo mogli bi smo kreirati notifikaciju koja se korisniku šalje nakon potvrde kartičnog plaćanja. Jedini dio sadržaja koji je potrebno promijeniti u ovoj notifikaciji je u suštini količina uplaćenog novca i datum i vrijeme.

## 4. Abstract factory pattern

Ovaj pattern bi se mogao iskoristiti zajedno sa Factory method patternom da bi se kreirala metoda koja generiše notifikacije na osnovu određene email adrese i samog sadržaja ili tipa notifikacije. Dakle ovo bi bio neki generalni objekat proizvođač notifikacija koji može generisati više različitih vrsta notifikacija od kojih svaka zahtijeva neke pojedine elemente, ili generisati notifikaciju koja ne prati nikakvu predodređenu sadržajnu strukturu već ima sadržaj koji je određen namjenski. Kao primjer familije klasa mogli bismo imati familiju notifikacija koje su namijenjene krajnjim korisnicima i administratorima odnosno uposlenicima sistema. Notifikacije koje su namijenjene administratorima mogu imati dodatne informacije o stanju samog sistema koje su dodane na kraj notifikacije, a notifikacije koje su namijenjene korisnicima ne bi imale ove podatke.

## 5. Builder pattern

Ovaj pattern će biti primijenjen u ovom projektu. Kako bi korisnik pronašao određenu sobu korisnik mora izgraditi odgovarajući filter. Korisnik određuje koja mu svojstva odgovaraju i nadodaje filtere za sobe. Prikaz svih soba se mijenja ovisno o filteru.

Dva patterna za primjenu:

- Singleton pattern
- Builder pattern