

# Kreacijski paterni

U ovom sistemu dva paterna koje je pogodno implementirati su:

- Singleton,
- Factory Method

U nastavku je dato objašnjenje gdje i zašto bi se primjenili ovi paterni.

## Singleton

Singleton patern je koristan za klasu Korisnik u situacijama kada u aplikaciji u svakom trenutku postoji samo jedan trenutno aktivan (ulogovan) korisnik. Korištenjem Singletona možemo omogućiti globalni pristup toj jednoj instanci korisnika, bez potrebe da se objekat stalno prosleđuje kroz funkcije ili klase. Samim tim možemo omogućiti samo jednu konekciju sa bazom u pokrenutoj aplikaciji na uređaju. Ovim bi postigli očuvanje resursa, umjesto da se svaki put otvara nova konekcija, koristi se jedna zajednička instanca.

Još jedan primjer upotrebe bi bio korištenjem za logiku slanja Newslettera da bi izbjegli kreiranje više instanci odjednom. Ovako samo jedan objekt šalje newsletter svima i spriječava se da više modula paralelno šalje isti sadržaj.

## FactoryMethod

Factory Method je dizajnerski obrazac koji omogućava kreiranje objekata bez navođenja njihove konkretne klase, već preko metode koja odlučuje koji tip objekta da instancira na osnovu prosljeđenih parametara. Može se primijeniti na više načina i na različitim klasama. Kao prvo može se primijeniti na klasama Korisnik, Recept i pri kreiranju instanci i izmjena već postojećih instanci. Bez ovog paterna za kreiranje svake instance ili modifikacije iste morali bi pozvati svaku pojedinačnu klasu, dok ovako na osnovu prosljeđenih parametara pozivamo samo metodu. Time izbjegavamo ponavljanje unutar koda i "čišći" kod.