

# KREACIJSKI PATERNI

## 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 možemo iskoristiti ako bi kreirali klasu Direktor jer je direktor jedna osoba koja se ne mijenja određeni vremenski period.

## Prototype patern:

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

Nas sistem se sastoji od klasa kod kojih je većina atributa za većinu instanci slične tako da je ovaj prototip već iskorišten kod klase Osoba. Jedna osoba može biti i Pacijent i Uposlenik pri čemu za informacije o toj osobi čuvaju u instanci klase Osoba.

## Factory method patern:

Uloga Factory method paterna je da omogućiti kreiranje objekata na način da podklase odluče koju klasu instancirati.

Obzirom da su klase Uposlenik i Pacijent izvedene iz klase osnovne klase Osoba, kreiranje instanci ove klase možemo jednostavno prebaciti na drugo mjesto u programu tako da bismo olaksali i sebi i pojednostavili dodavanje novih tipova korisnika ukoliko to bude potrebno u kasnijem razvoju i dorađivanju aplikacije.

## Abstract factory patern:

Abstract factory patern omogućava da se kreiraju familije povezanih objekata.

U našem sistemu ovaj sablon nije moguće primijeniti.

## Builder patern:

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

Patern bi mogli iskoristiti kada bi imali ordinaciju u više gradova ili država (ne samo u jednom gradu). Pri ulasku na našu stranicu korisnik bi odabrao grad i na osnovu odabranog grada klasa GradBuilder, koja implemrentira IBuilder interfejs, bi generisala stranicu sa uslugama dostupnim u tom gradu na jeziku koji je službeni za tu državu.