# KREACIJSKI PATTERNI

## Prototype pattern

*Uloga Prototype patterna je da kreira nove objekte klonirajući jednu od postojećih prototip instanci.*

U našem sistemu ovaj pattern možemo iskoristiti za klasu Stol.Recimo kada se kreira nova Rezervacija, ili se Stol dodaje u određeni lokal i zatreba nam identičan objekat klase Stol možemo samo pozvati metodu clone().

Factory Method

*Uloga Factory Method patterna je da omogući kreiranje objekata na način da podklase odluče koju klasu instancirati.*

U našem sistemu ovaj pattern možemo iskoristiti za podklase PlacanjeUzivo i KarticnoPlacanje. Klijent može da izabere da plati gotovinom i u tom slučaju se šalje obavijest uposleniku koji je zadužen za njegov sto. Ako klijent izabere da plati putem kreditne kartice, otvaraju mu se polja za unos broja kreditne kartice, datuma isteka i CVV-a.

## Singleton patern

*Singleton pattern se koristi kako bi se osiguralo da se klasa može instanicarati samo jednom i da osigura globalni pristup kreiranoj instanci klase.*

U našem sistemu ovaj pattern možemo iskoristiti za klasu Musterija, kako bismo izbjegli probleme prilikom kreiranja rezervacija ili narudžbi ako bismo to radili preko 2 odvojene instance ove klase. Ovi objekti se koriste za logiranje te ih je zbog toga potrebno samo jednom instancirati. Tako da bi se kreiranje nove instance Kupac vršilo prilikom logovanja u sami sistem.

## Abstract Factory pattern

Uloga Abstract Factory patterna je omogućavanje da se kreiraju familije povezanih objekata.

Ovaj pattern nećemo primijeniti jer naša aplikacija nema veliki broj srodnih objekata/klasa.

## Builder pattern

Uloga Builder patterna je odvajanje specifikacije kompleksnih objekata od njihove stvarne konstrukcije.