



*Univerzitet u Sarajevu*  
**Elektrotehnički fakultet**  
*Sarajevo*

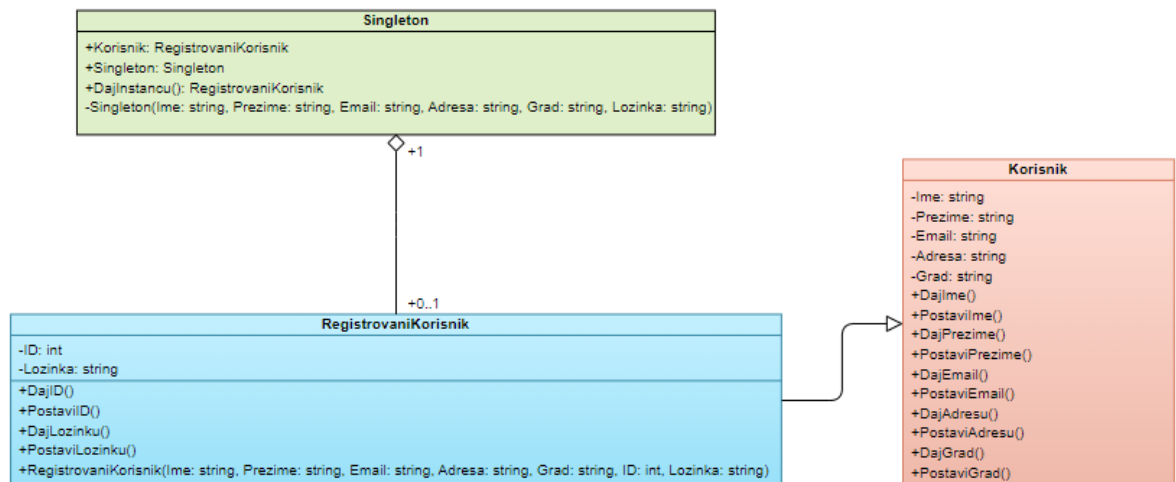
# **KREACIJSKI PATERNI**

## **-AMA Cosmetics-**

## 1. SINGLETON PATTERN

Ovaj patern osigurava da postoji samo jedna instanca određene klase. Singleton patern pruža mogućnost da se klasa instancira samo jednom i da osigura globalni pristup kreiranoj instanci klase.

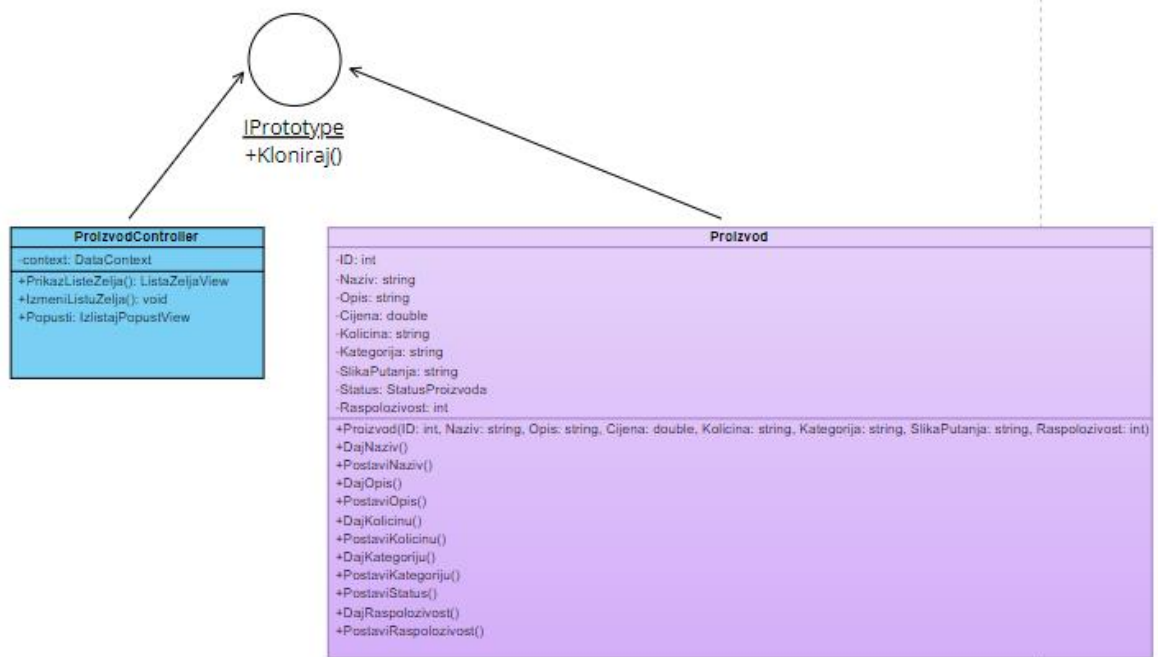
U našem sistemu ovaj patern se može primijeniti kod registracije korisnika. Napraviti ćemo jednu Singleton klasu, gdje će se nalaziti svi upiti, odakle ćemo dobavljati podatke, odnosno klasa u kojoj se nalazi konekcija na bazu. Time ćemo omogućiti da će postojati samo jedna konekcija na bazu, što je vrlo bitno da ne bi došlo do greške zbog pokušavanja otvaranja više konekcija ka jednoj bazi istovremeno.



## 2. PROTOTYPE PATTERN

Prototype patern služi za kloniranje objekata koji zauzimaju previše resursa. Ovaj patern pruža mogućnost kreiranja objekata koristeći neki postojeći prototip postojećeg objekta. Prednost ovog patern je što se mogu kreirati prilagođeni objekti bez poznavanja detalja kako je objekat kreiran ili njihove klase.

U našem sistemu se ovaj patern može primijeniti na proizvodima, među kojima se može naći veliki broj sličnih objekata. Napraviti ćemo interfejs koji će klonirati instance objekata, tj. sadržavati samo metodu Kloniraj( ).



### 3. FACTORY METHOD PATTERN

Factory Method Patern pruža mogućnost kreiranja objekata na način da se odluči koja se podklasa instancira.

Ukoliko bi se naš sistem u budućnosti proširio (npr. korisnik ima mogućnost da bira da li želi da obavi kupovinu u Bosni i Hercegovini ili u inostranstvu) ovaj patern bi bio od pomoći. U tom slučaju napravili bismo dvije nove klase poput: KorpaBiH i KorpaInostranstvo.

### 4. ABSTRACT FACTORY PATTERN

Abstract Factory patern omogućava stvaranje familija srodno povezanih objekata bez eksplicitnog specificiranja njihovih konkretnih klasa.

U našem sistemu ovaj patern možemo primijeniti za stvaranje različitih vrsta proizvoda, npr. proizvodi za lice i proizvodi za kosu.

## 5. BUILDER PATTERN

Svrha builder patterna je da smanji komplikovanje klasa koje su sklone nadograđivanju.

U našem sistemu ovaj patern možemo upotrijebiti kod klase *RegistrovaniKorisnik*. Sistem na početku traži neke osnovne informacije o korisniku, kao što su ime, prezime, adresa... , dok se prilikom plaćanja od korisnika zahtjeva više informacija.