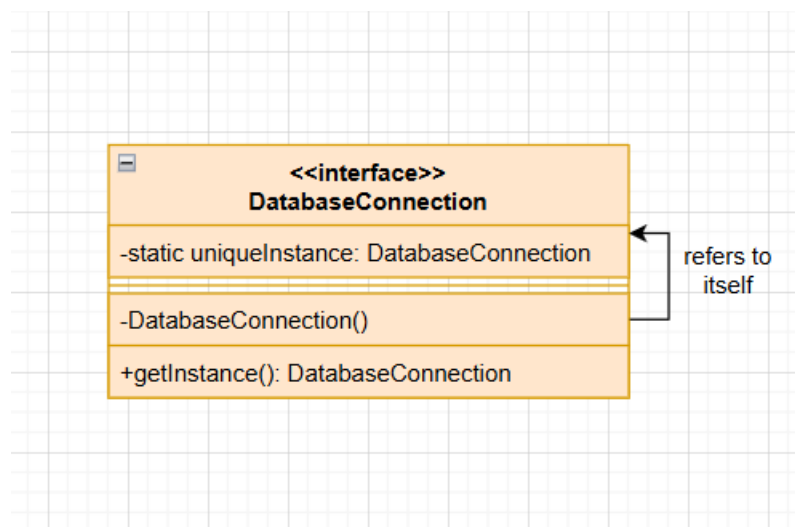


# KREACIJSKI PATERNI

Sistem za upravljanje nekretninama, kako bi ostao skalabilan, fleksibilan i jednostavan za održavanje i korištenje, treba da koristi dizajn paterne koji rješavaju specifične probleme u inicijalizaciji objekata. Kreacijski paterni omogućavaju da se instanciranje objekata izdvoji iz klasa koje ih koriste.

## 1. SINGLETON PATTERN

Uloga Singleton paterna je da osigura da se klasa može instancirati samo jednom i da osigura globalni pristup kreiranoj instanci klase. Koristi se i za odgođeno (lazy) instanciranje objekata. S obzirom na to da naš sistem zahtijeva pristup bazi podataka iz različitih dijelova koda, možemo implementirati Singleton klasu koja sadrži logiku za uspostavljanje veze sa bazom podataka i izvršavanje upita. Također se može koristiti i za kreiranje „Logger servisa“ koji bilježi systemske aktivnosti. Za prikaz na dijagramu izabrali smo prvi navedeni primjer – upravljanje konekcijom s bazom podataka (DatabaseConnection).



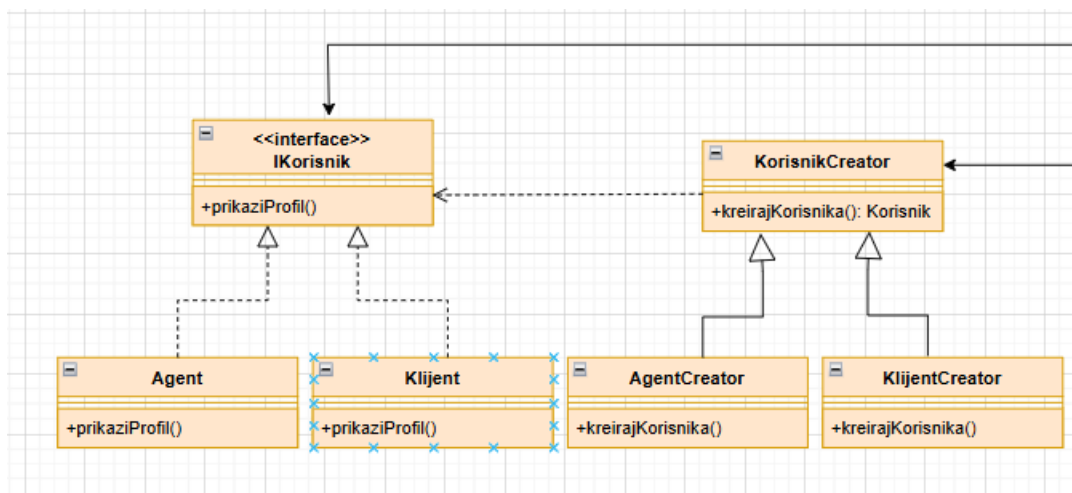
## 2. PROTOTYPE PATTERN

Prototype dizajn patern dozvoljava da se kreiraju prilagođeni objekti bez poznavanja njihove klase ili detalja kako je objekt kreiran. Ovaj patern omogućava da klase budu kopirane ili klonirane iz neke instance koja predstavlja prototip, umjesto da se kreiraju nove instance. Prototype patern ima mnoge prednosti, između ostalog ubrzava instanciranje veoma velikih klasa koje se dinamički učitavaju. U našem sistemu ovaj patern možemo koristiti kada želimo omogućiti da korisnik (agent/vlasnik nekretnine) može kopirati postojeću nekretninu da bi unio

novu sličnu (recimo isti opis, lokacija, struktura oglasa...) uz male izmjene. Drugim riječima, koristimo ga kada se radi na šablonima oglasa za nekretnine koje se često ponavljaju, npr. tipični stanovi u novogradnji.

### 3. FACTORY METHOD

Uloga Factory method paterna je da omogući kreiranje objekata na način da podklase odluče koju klasu instancirati. Ovaj patern odvaja zahtijevanje objekata od kreiranja objekata. Klijent ne treba da zna koji tip objekta je kreiran. Što se tiče primjene factory method paterna u našem sistemu, mogli bismo ga iskoristiti na dva načina. Prvi slučaj je kreiranje različitih tipova korisnika (Admin, Klijent, Agent...), a drugi slučaj je kreiranje specifičnih vrsta nekretnina (Stan, Kuća, PoslovniProstor...). Na dijagramu klasa možemo prikazati oba pristupa, međutim radi jednostavnosti prikazat ćemo samo prvi slučaj iz kojeg se bez velike muke može shvatiti princip upotrebe ovog dizajn paterna.



### 4. ABSTRACT FACTORY PATTERN

Abstract Factory patern omogućava da se kreiraju familije povezanih objekata/produkata bez specificiranja konkretnih klasa. Patern odvaja definiciju (klase) produkata od klijenta. Abstract Factory patern se koristi kada postoji hijerarhija objekata koja enkapsulira različite platforme sastavljene od grupe povezanih objekata. U našem sistemu ga možemo koristiti kada kreiramo porodicu povezanih objekata, npr kada agent/vlasnik nekretnine kreira nekretninu zajedno sa kontakt formom, galerijom slika i početnim statusom.

## 5. BUILDER PATTERN

Uloga builder paterna je odvajanje specifikacije kompleksnih objekata od njihove stvarne konstrukcije. Koristi se ako se objekti mogu podijeliti u skupove objekata koji se razlikuju samo po permutaciji njihovih dijelova. Najbolji primjer za builder patern unutar našeg sistema je slučaj filtriranja nekretnina. Ovaj patern je idealan za kreiranje filtera za pretragu gdje korisnik može postepeno dodavati kriterije: lokacija, cijena, površina, broj soba ...). Pruža čitljiv i fleksibilan način pravljenja složenih objekata, tj. omogućava korisniku da postepeno sastavi objekat bez konfuzije.