

Kreacijski šabloni

1. Singleton šablon

Singleton šablon osigurava a da se klasa može instancirati samo jednom i da osigura globalni pristup kreiranoj instanci klase. Postoje objekti čija je samo jedna instanca potrebna i nad kojima je potrebna jedinstvena kontrola pristupa. Instanciranje više nego jednom može prouzrokovati probleme kao što su nekorektno ponašanje programa, neadekvantno korištenje resursa ili nekonzistentan rezultat.

U našem projektu bi klasa Administrator mogla biti realizovana kao singleton klasa, tada bi konstruktor klase bio proglašen privatnim, dodan statički atribut tipa Administrator kao i statička metoda `getInstanca(): Administrator`. Međutim, kako se naš tim opredijelio za mogućnost postojanja više od jednog administratorskog računa, ovaj šablon nećemo implementirati u našem sistemu.

2. Prototype šablon

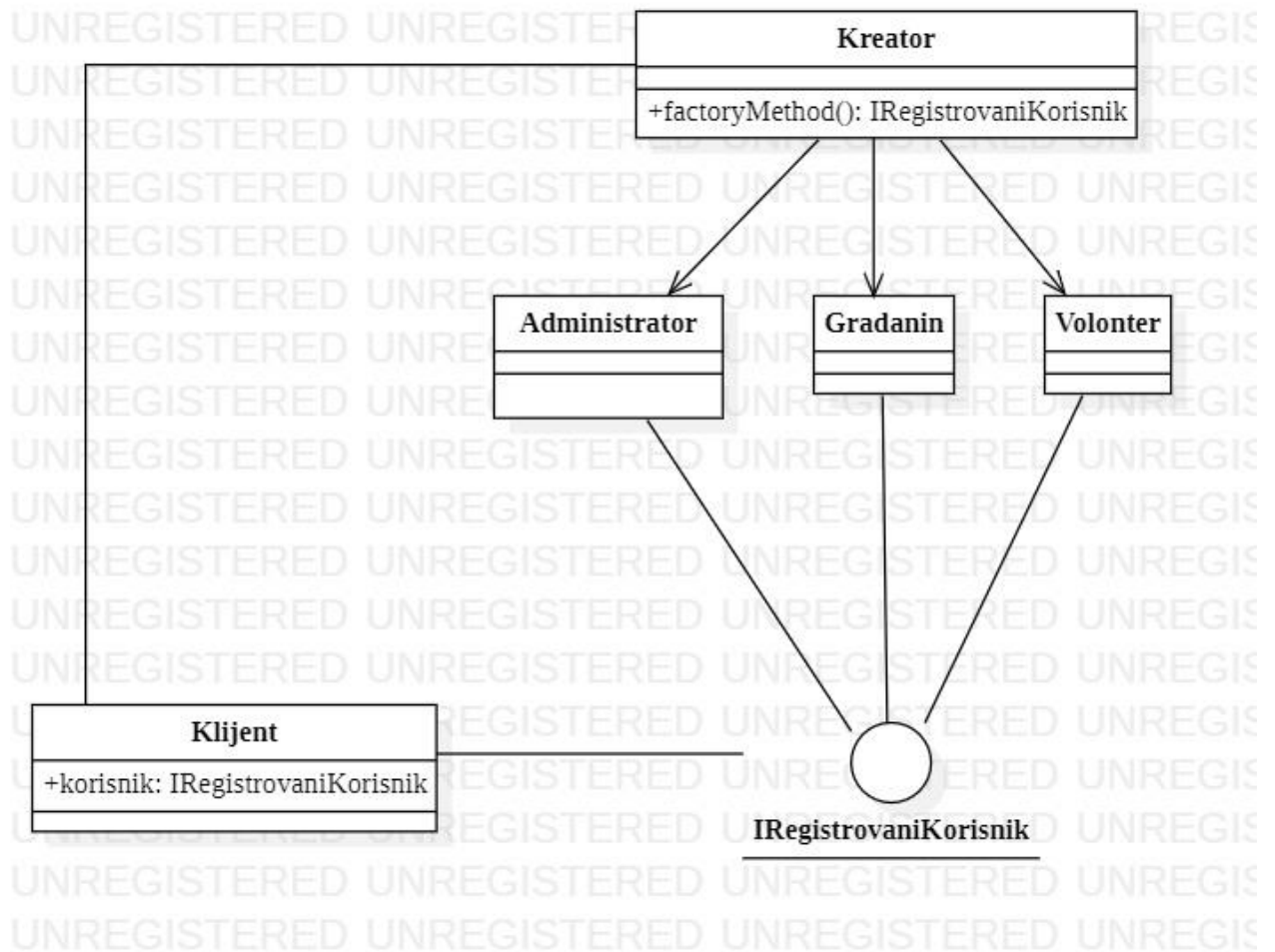
Prototype šablon kreira nove objekte klonirajući jednu od postojećih prototip instanci. Dakle, ovaj šablon omogućava jednostavnije kreiranje novih instanci klasa kod kojih je veliki broj atributa identičan za većinu instanci.

Naš sistem se sadrži od klasa kod kojih je većina atributa za većinu instanci različita, čak i svi različiti, jer su u pitanju klase koje modeliraju jedinstvenu osobu, zahtjev ili žalbu pa niti ima potrebe za kloniranje, niti ima smisla isto činiti.

3. Factory Method šablon

Factory Method šablon omogućava kreiranje objekata na način da podklase odluče koju klasu instancirati. Različite podklase mogu na različite načine implementirati interfejs.

U Solidarity sistemu ovaj šablon može biti iskorišten kod klasa koje su izvedene iz klase `RegistrovaniKorisnik`. Instanciranje klasa može se prebaciti na jedno mjesto u programu, što će omogućiti lakše dodavanje novih vrsta korisnika ukoliko bude potrebe za time prilikom budućeg razvoja aplikacije.



4. Abstract Factory šablon

Abstract Factory šablon se koristi pri radu sa familijama različitih objekata. Na osnovu apstraktne familije objekata kreiraju se konkretne fabrike produkata različitih tipova i različitih kombinacija.

U našem sistemu ovaj šablon nije moguće primjeniti.

5. Builder šablon

Builder šablon odvaja specifikaciju kompleksnih objekata od njihove stvarne konstrukcije. Isti konstrukcijski proces može kreirati različite reprezentacije.

Za potrebe naše aplikacije nije potrebno koristiti builder šablon prilikom kreiranja objekata pošto su klase u sistemu relativno proste, odnosno nije potrebno odvojeno kreiranje dijelova objekta. Međutim, za slučaj da je klasa RegistrovaniKorisnik ili

podklase izvedene iz nje kompleksnije, tada bi učinkovitije bilo attribute grupisati u dijelove i onda iz tih dijelova sklopiti objekat.