

Kreacijski paterni

Singleton patern

Uloga Singleton patern je da osigura da se klasa može instancirati samo jednom i da osigura globalni pristup kreiranoj instanci klase. U našoj aplikaciji sve klase su singleton klase. Kada god korisniku treba instanca ove klase, može se poslužiti statičkom metodom getInstance preko koje pristupa singleton klasi.

Prototype patern

Uloga Prototype patern je da kreira nove objekte klonirajući jednu od postojećih prototip instanci (postojeći objekat). Naša aplikacija nema potrebe za kloniranjem objekata, te ga stoga nismo implementirali.

Factory Method patern

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

Factory Method patern je implementiran jer se na osnovu odabranog vremena potrebnog za kuhanje vrši filtriranje i izbacuju se odgovarajući recepti.

Abstract Factory patern

Abstract Factory patern omogućava da se kreiraju familije povezanih objekata/produkata. Na osnovu apstraktne familije produkata kreiraju se konkretne fabrike produkata različitih tipova i različitih kombinacija.

Ovaj patern se može iskoristiti dodavanjem novih klasa za tipove recepata. Recimo da imamo klase *Vegetrijanska*, *Veganska* i *Ostala* jela. Da je naša aplikacija pružala mogućnost razlikovanja korisnika na osnovu hrane koje jedu i da smo korisnicima dodali opciju da imaju personalizirane preporuke za obrok, koristili bismo ovaj patern.

Builder patern

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

Ovaj patern možemo iskoristiti za preporuke jela na osnovu lokacije. Pri ulasku na stranicu korisnik bi imao mogućnost da upiše trenutnu lokaciju i na osnovu toga bi se vršio prikaz preporučenih recepata. Naprimjer, ako se korisnik trenutno nalazi u Italiji, na početnoj stranici će dobit preporuke talijanskih jela.