Kreacijski paterni

1. Singleton patern

Uloga Singleton paterna je da osigura da se klasa može instancirati samo jednom i da osigura globalni pristup kreiranoj instanci klase. Postoji više objekata koje je potrebno samo jednom instancirati i nad kojim je potrebna jedinstvena kontrola pristupa. U nasoj aplikaciji ovaj patern bi mogli koristiti za instanciranje klase za placanje. Jer dovoljno da jednom kreiramo tu instancu.

2. Prototype patern

Uloga Prototype paterna je da kreira nove objekte klonirajući jednu od postojećih prototip instanci (postojeći objekat). Ako je trošak kreiranja novog objekta velik i kreiranje objekta je resursno zahtjevno tada se vrši kloniranje već postojećeg objekata. Prototype dizajn patern dozvoljava da se kreiraju prilagođeni objekti bez poznavanja njihove klase ili detalja kako je objekat kreiran. U nasoj aplikaciji ovaj patern bi mogli iskoristiti za slucaj kada registrovani korisnik zeli da bude I korisnik koji predstavlja vlasnik terena, u tom slucaju kopiranje I castanje iz jednog tipa u drugi uz male promjene ce veoma posluziti, jer obije klase imaju broj identicnih atributa. Ovaj patern takodjer mozemo koristit I za kreiranje termina.

3. Factory Method Patern

Uloga Factory Method paterna je da omogući kreiranje objekata na način da podklase odluče koju klasu instancirati. Različite podklase mogu na različite načine implementirati interfejs. Factory Method instancira odgovarajuću podklasu(izvedenu klasu) preko posebne metode na osnovu informacije od strane klijenta ili na osnovu tekućeg stanja. Ovaj patern mozemo koristit kod kreiranje tenera gdje ce se u zavisnosti od lokacije I sportova mijenjati. Takodjer mozemo iskoristit pri kreiranju korisnika u zavisnosti da li je korsnik odabrao opciju “vlasnik tenera” pri registraciji

4. Abstract Factory Patern

Abstract Factory patern omogućava da se kreiraju familije povezanih objekata/produkata. Na osnovu apstraktne familije produkata kreiraju se konkretne fabrike (factories) produkata različitih tipova i različitih kombinacija. Patern odvaja definiciju (klase) produkata od klijenta. Zbog toga se familije produkata mogu jednostavno izmjenjivati ili ažurirati bez narušavanja strukture klijenta. U nasoj aplikaciji nema uocljivo mjesto gdje bi se ovaj patern mogao iskoristiti

5. Builder Patern

Uloga Builder paterna je odvajanje specifikacije kompleksnih objekata od njihove stvarne konstrukcije. Isti konstrukcijski proces može kreirati različite reprezentacije. Ovaj patern mozemo koristiti pri prikazivanju odredjenih dijelova/elemenata na ekranu u zavisnosti od tipa korisnika koji koristi aplikaciju (Admin ce imati opcije za editovanje tenera dok obicni korisnik nece), to nam daje ustedu resursa I koda.