

## Kreacijski paterni

Sistem Mediplan sadrži entitete poput korisnika, termina, usluga i nalaza. Radi bolje skalabilnosti, održavanja i organizacije koda, uvedeni su kreacijski paterni – obrasci za kontrolisano i fleksibilno kreiranje objekata. U projektu smo implementirale **Factory Method** i **Builder**, dok su ostali paterni predstavljeni kao potencijalno korisni.

### Factory Method

Factory Method je kreacijski patern koji definiše interfejs za kreiranje objekta, dok podklase odlučuju koju konkretnu klasu treba instancirati. Time se omogućava fleksibilno dodavanje novih tipova objekata bez potrebe za izmjenama postojećeg koda. U projektu je implementirana *KorisnikFactory*, koja omogućava kreiranje različitih tipova korisnika (npr. Pacijent, Doktor) na osnovu prosljeđenog tipa. Korišten je interfejs *KorisnikCreator* sa metodom *kreirajKorisnika()*, a konkretne klase *PacijentKreator* i *DoktorKreator* implementiraju ovu metodu za stvaranje specifičnih korisnika. Ovaj pristup pojednostavljuje dodavanje novih uloga u sistem i poboljšava čitljivost i testabilnost koda.

### Builder

Builder je patern koji omogućava postepeno kreiranje složenih objekata pomoću niza metoda, bez potrebe za velikim konstruktorima sa puno parametara. U projektu je implementirana klasa *TerminBuilder*, koja omogućava postepenu izgradnju objekta *Termin*. Putem metoda kao što su *zaPacijenta()*, *kodDoktora()*, *saDatumom()*, *saLokacijom()* i *saUslugama()* moguće je unositi podatke redom kojim su dostupni. Na kraju se koristi *build()* metoda da bi se kreirao gotov *Termin* objekat. Ovaj pristup je posebno koristan prilikom unosa termina putem formi ili API-ja, gdje se informacije prikupljaju u koracima. Patern omogućava fleksibilniju manipulaciju podacima i olakšava održavanje složenih struktura.

### Singleton

Singleton je patern koji osigurava da određena klasa ima samo jednu instancu i omogućava globalni pristup toj instanci. Koristan je za upravljanje centralizovanim resursima poput baze podataka, servisa za slanje e-mailova ili rasporeda termina. U projektu se može koristiti za servis koji obavlja slanje notifikacija, čime se osigurava da postoji samo jedan objekat koji upravlja tim resursom kroz cijeli sistem.

### Prototype

Prototype je patern koji omogućava kloniranje postojećih objekata. Umjesto da se objekti prave ispočetka, koristi se već kreirani prototip koji se zatim prilagođava po potrebi. U sistemu se može koristiti za ponavljanje šablonskih usluga ili paketa. Na primjer, ako bolnica nudi unaprijed definisane pakete pregleda, svaki novi termin se

može kreirati kloniranjem postojećeg prototipa paketa i izmjenom nekoliko parametara.

### **Abstract Factory**

Abstract Factory je patern koji omogućava kreiranje povezanih grupa objekata bez navođenja njihovih konkretnih klasa. Omogućava da se cjelokupna konfiguracija sistema mijenja jednostavno zamjenom fabrike. U kontekstu sistema Mediplan, ovaj patern bi omogućio kreiranje različitih varijacija sistema za specijalizirane klinike. Na primjer, jedna fabrika može kreirati objekte specifične za ginekološku ordinaciju (termini, nalazi, usluge), dok druga fabrika kreira iste tipove objekata ali za stomatološku ordinaciju. Time se postiže fleksibilnost pri prilagođavanju sistema različitim vrstama zdravstvenih ustanova.