

# KREACIJSKI PATERNI

Kreacijski paterni u dizajnu softvera su ključni za rješavanje problema stvaranja objekata na fleksibilan i efikasan način. Oni omogućavaju kontrolu nad stvaranjem objekata, olakšavaju promjene u načinu kreiranja objekata, i pomažu u upravljanju kompleksnošću sistema. U ovom dokumentu ćemo objasniti potrebu za dodavanjem kreacijskih paterna u postojeći sistem za online rezervaciju i prodaju karata. Fokusirat ćemo se na primjenu Factory Method i Singleton paterna, te ćemo objasniti gdje i kako bi se ostali kreacijski paterni mogli primijeniti.

## Primjena Kreacijskih Paterna

### Factory Method Pattern

Factory Method patern omogućava stvaranje objekata pomoću metoda koji vraćaju instance različitih klasa, čuvajući pritom apstrakciju i izbjegavajući direktno korištenje konkretnih klasa.

*Primjer:*

U našem sistemu možemo koristiti Factory Method patern za kreiranje različitih tipova karata (VIP, standardna, studentska, itd.) bez direktnog instanciranja tih klasa.

### Singleton Pattern

Singleton patern osigurava da postoji samo jedna instanca određene klase i pruža globalnu tačku pristupa toj instanci. Ovo je korisno za upravljanje resursima kao što su konfiguracije, konekcije prema bazama podataka, ili upravljanje sesijama korisnika.

*Primjer:*

U našem sistemu možemo koristiti Singleton patern za upravljanje konekcijama prema bazi podataka.

## Ostali Kreacijski Paterni

### Builder Pattern

Builder patern omogućava postepenu izgradnju kompleksnih objekata. Umjesto kreiranja objekata kroz kompleksne konstruktore, koristi se niz koraka koji grade krajnji objekat.

*Primjena:*

Može se koristiti za kreiranje kompleksnih entiteta kao što su Manifestacija sa mnogim opcionalnim atributima (npr. naziv, datum, lokacija, lista izvođača).

### Prototype Pattern

Prototype patern omogućava stvaranje novih objekata kopiranjem postojećih, umjesto kreiranja novih instanci. Ovo je korisno kada je kreiranje objekta skupo ili kompleksno.

*Primjena:*

Može se koristiti za kreiranje duplikata Karte sa sličnim atributima.

## **Abstract Factory Pattern**

Abstract Factory patern omogućava kreiranje familije povezanih objekata bez specifikacije njihovih konkretnih klasa. Koristi se kada postoji potreba za kreiranjem različitih setova objekata.

*Primjena:*

Može se koristiti za kreiranje različitih Tipova Karata i Plaćanja.

Dodavanje kreacijskih paterna u postojeći sistem za online rezervaciju i prodaju karata donosi mnoge prednosti, uključujući fleksibilnost, ponovnu upotrebljivost koda, i lakšu održivost sistema. Primjena Factory Method i Singleton paterna pomaže u rješavanju specifičnih problema vezanih za kreiranje objekata u našem sistemu, dok se ostali kreacijski paterni mogu koristiti za rješavanje drugih složenih situacija. Na taj način, sistem postaje modularniji, lakši za proširenje i održavanje.