

Kreacijski patterni

*boldirane pattern-e ćemo sigurno primijeniti

1. Singleton pattern

Singleton pattern koristi se kada želimo da neka klasa ili objekat bude instanciran samo jednom, te kada je važno osigurati jedinstvenu tačku pristupa toj instanci. U našem slučaju izbor najprodavanijeg proizvoda mjeseca- Kupac na početnoj stranici aplikacije može vidjeti najprodavaniji proizvod mjeseca. Proizvod mjeseca se bira jednom u mjesec dana. Zaključili smo da klasa proizvod mjeseca može imati samo jednu instancu, a vrstu proizvoda možemo mijenjati.

2. Prototype pattern

Ovaj pattern koristi se kada je potrebno kopirati ili klonirati objekte – bilo da je riječ o površinskom ili dubokom kopiranju. U kontekstu našeg sistema, njegova primjena bi bila nad klasama Proizvod i Cvijet. Ukoliko želimo da u proizvod koji može biti buket ili poklon kutija dodamo određenu količinu cvijeća iste vrste, samo primijenimo operaciju kloniranja nad instancama klase Cvijet.

3. Factory method pattern

Factory Method pattern koristi se kada želimo kreirati objekte iz više različitih izvedenih klasa. Ovaj obrazac se implementira putem interfejsa *iProduct* i klasa poput *ProductA*, *ProductB* i slično, koje implementiraju taj interfejs. Njegova primjena bi bila moguća kroz mehanizam generisanja izvještaja. Sistem može kreirati tri različite vrste izvještaja – dnevni, mjesečni i godišnji – u zavisnosti od potreba zaposlenika. U tom slučaju, apstraktna klasa *Izvještaj* može poslužiti kao osnova za instanciranje odgovarajuće varijante izvještaja.

4. Abstract factory pattern

Ovaj obrazac koristi se za kreiranje grupe međusobno povezanih objekata, bez oslanjanja na konkretne klase, te omogućava razdvajanje definicije objekata od njihove upotrebe. U našem sistemu, potencijalna primjena ovog patterna mogla bi biti kroz klasu *Popust*, iz koje bi nasljeđivale specifične klase. Te izvedene klase bi predstavljale različite vrste popusta – na primjer, one koje se aktiviraju unosom koda iz email-a, te one koje važe za posebne prilike i značajne datume.

5. Builder pattern

Builder pattern koristi se za razdvajanje procesa kreiranja kompleksnih objekata od njihove konkretne reprezentacije. Omogućava da se isti proces izgradnje koristi za stvaranje različitih varijanti objekata. U našem sistemu, ovaj pattern može biti koristan u situacijama gdje imamo složene objekte sa opcionim dijelovima. Na primjer, korisnik može odabrati dodatne elemente prilikom kreiranja narudžbe – kao što je dodavanje personalizovane kartice uz poklon – bez utjecaja na osnovnu strukturu ili cijenu narudžbe. Ako se kartica ne odabere, narudžba se generiše automatski sa podrazumijevanim postavkama.