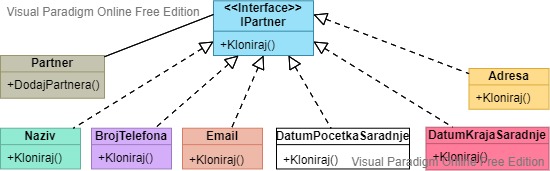
# KREACIJSKI PATERNI

1. *SINGLETON PATERN*

* Singleton patern pruža mogućnost da se klasa instacira 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.
* Singleton patern se može koristiti kada :
* postoji više servera koji mogu uslužiti zahtjev korisnika.
* se serveri mogu uključivati/isključivati.
* je potreban jedan objekat koji zna sve o stanju mreže i koji vrši usmjeravanje saobraćaja.
* U našem sistemu ovaj patern se može upotrijebiti za admina.On bi bio jedinstvena klasa ,koji bi znao sve o stanju mreže i koji vrši preusmjeravanje saobraćaja.

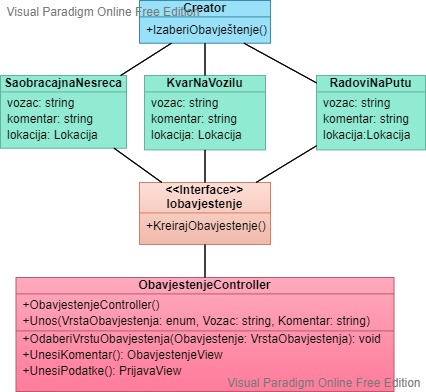
1. *PROTOTYPE PATERN*

* Prototype patern pruža mogućnost kreiranjaobjekata koristeći neki postojeći prototip postojećeg objekta. Prednost ovog paterna je što se mogu kreirati prilagođeni objekti bez poznavanja detalja kako je objekat kreiran ili njihove klase.Također, još jedna prednost ovog paterna je i što kloniranje već postojećeg objekta olakšava u slučaju da su veliki troškovi za kreiranje novog objekta i/ili je kreiranje objekta resursno zahtjevno.
* Ova vrsta paterna se može koristiti kod npr unosa partnera u sistem. Isti objekti se unose u sistem(naziv, adresa,broj telefona, datumi saradnje....) a detalje koji se razlikuju unosi administrator.



1. *FACTORY METHOD PATERN*

* Factory Method Patern pruža mogućnost kreiranja objekata tako da podklase odluče koju klasu instancirati.Patern instancira odgovarajuću izvedenu klasu,tj podklasu na osnovu tekućeg stanja ili od strane klijenta.
* U sistemu ovu vrstu paterna možemo primijeniti kod vozača koji kada šalje obavještenja može odabrati jednu od tri vrste obavještenja(saobraćajna nesreća, kvar na vozilu i radovi na putu). Potrebni su nam interfejs Iobavještenje, te klasu Creator sa metodom IzaberiObavjestenje koja odlučuje koju klasu instancirati.



1. *ABSTRACT PATERN*

* Abstract Patern pruža mogućnost da se kreira familija povezanih objekata i na osnovu toga se kreiraju konkretne fabrike različitih kombinacija i tipova.
* Ovu vrstu paterna možemo koristiti kod popusta. S obzirom da popust na svaku vožnju na svaku vožnju mogu dobiti samo ugostiteljski objekti , trebamo napraviti apstraktnu klasu iz koje se nasljeđuju dvije klase: za ugostiteljske objekte i za ostale korisnike. U zavisnosti koja je vrsta klijenta određujemo i vrstu popusta. Ovakav način nam olakšava dodavanja drugih vrsta popusta za druge vrste korisnika.

1. *BUILDER PATERN*

* Builder patern odvaja specifikacije kompleksnih objekata od njihove stvarne konstrukcije.
* Builder patern možemo iskoristiti tokom proces naručivanja. U našem sistemu imamo više pogodnosti za korisnike (majka sa djecom, osobe sa invaliditetom...), te zbog toga imamo i više vrsta vozila,koji ispunjavaju te zahtjeve. Tokom naručivanja vozila, ako korisnik izabere neke od mogućnosti, sistem će tražiti najbliže slobodno vozilo koje ispunjava sve zahtjeve korisnika.Npr. ako je u pitanju majka sa djetetom sistem će dodijeliti auto sa sjedalicom.