



Univerzitet u Sarajevu
Elektrotehnički fakultet Sarajevo
Odsjek za računarstvo i informatiku



Deveti projektni zadatak

Objektno orijentisana analiza i dizajn

Ime i prezime: Din Švraka 18857
 Nejla Bečirspahić 18835
 Amina Pandžić 18808
Grupa: šest (6) / DNACityGuide
Datum: 30.05.2022.

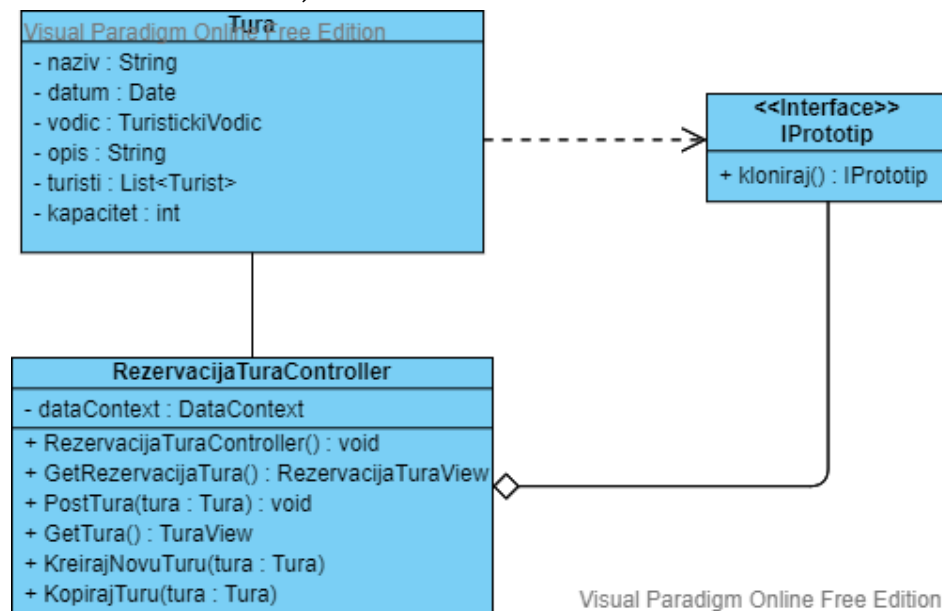
1. Kreacijski paterni

1.1. Singleton

Singleton patern je moguće primijeniti za prikaz mape grada Sarajeva u našoj aplikaciji, obzirom da će postojati jedinstvena mapa, koja će biti dostupna svim korisnicima i za koju nisu potrebne izmjene tokom određenog vremenskog perioda. Kako bismo što bolje zaštitili informacije o atrakcijama koje su označene na našoj mapi, potrebno je uvesti određeni vremenski period nakon kojeg je omogućeno administratoru sistema da unese određene izmjene na naš objekat. Npr. nakon perioda od godinu dana, administratoru je omogućeno da na mapi označi neke nove turističke atrakcije, ili obriše postojeće sa mape.

1.2. Prototype

Prototype pattern je moguće implementirati za klasu Tura, gdje bismo mogli implementirati podklase koje ćemo praviti na osnovu imena ture. Tada bismo mogli efikasnije praviti kopije ovakvih objekata, i na mnogo efikasniji način vršiti minorne promjene koje su potrebne za određeni objekat. Npr. mogli bismo imati dva identična objekta, koji se razlikuju samo u tome koji turistički vodič je zadužen za turu, ili jedina razlika jeste u datumu obilaska, itd.

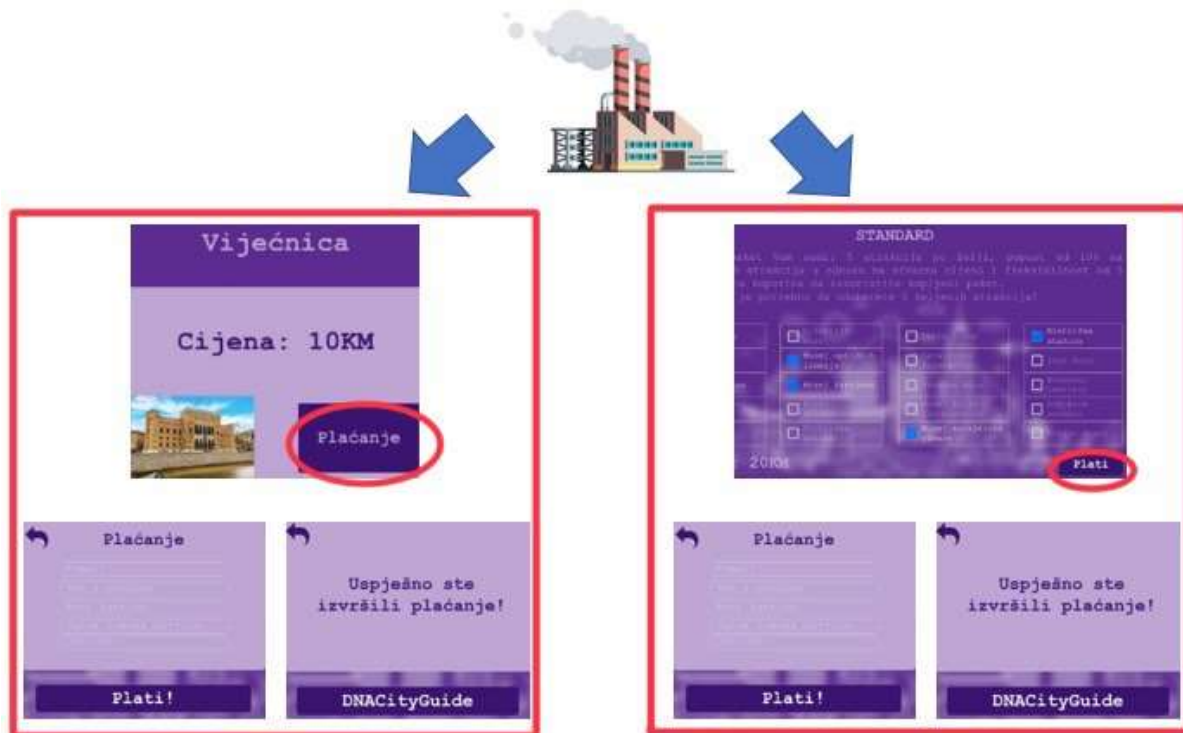


Slika 1. Struktura Prototype paternu

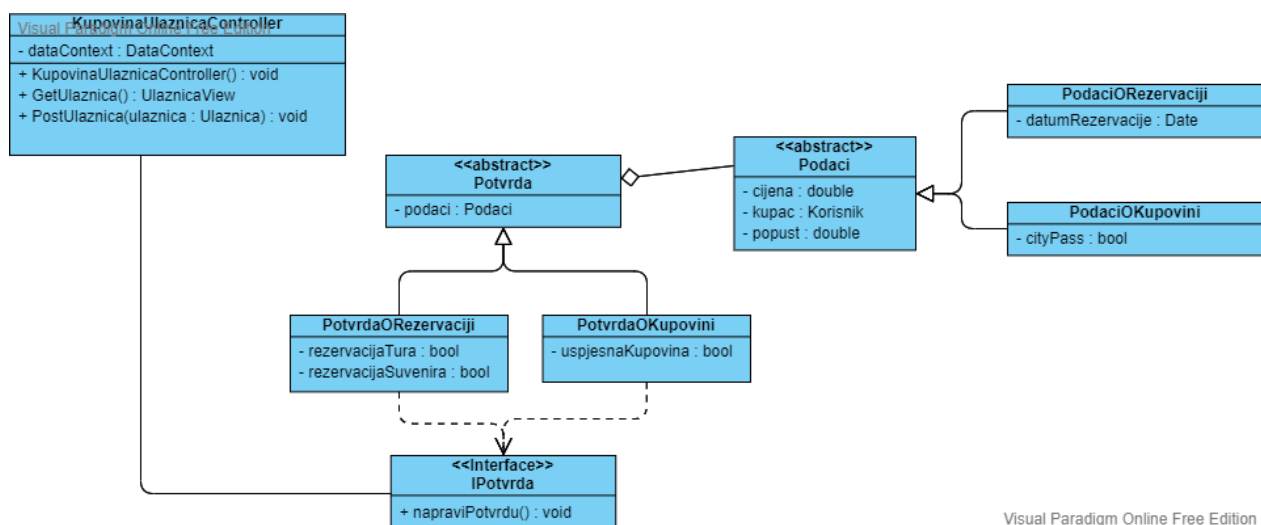
1.3. Factory method

Kreiranje potvrda je idealna prilika za korištenje Factory method paternu. Naime postoje dva tipa potvrda, a to su potvrda o rezervaciji i potvrda o kupovini, odnosno izvršenom plaćanju. Shodno tome, moguće su nam dvije verzije implementacije navedenog paternu. Obratimo pažnju na potvrdu o rezervaciji. U našoj aplikaciji su moguće rezervacije tura i rezervacije suvenira, dakle imamo dvije različite vrste rezervacija, a istu formu potvrde.

Slično se dešava i sa potvrdom o plaćanju/kupovini. Moguće je kupiti CityPass paket, ali i ulaznice za svaku atrakciju posebno pri čemu je potvrda o plaćanju/kupovini karakteristika i jedne i druge opcije. Navedena dva primjera su izuzetno dobra za implementiranje Factory method paterna jer je omogućeno instanciranje objekata koji nude iste mogućnosti koristeći istu metodu.



Slika 2. Ilustracija Factory method paterna



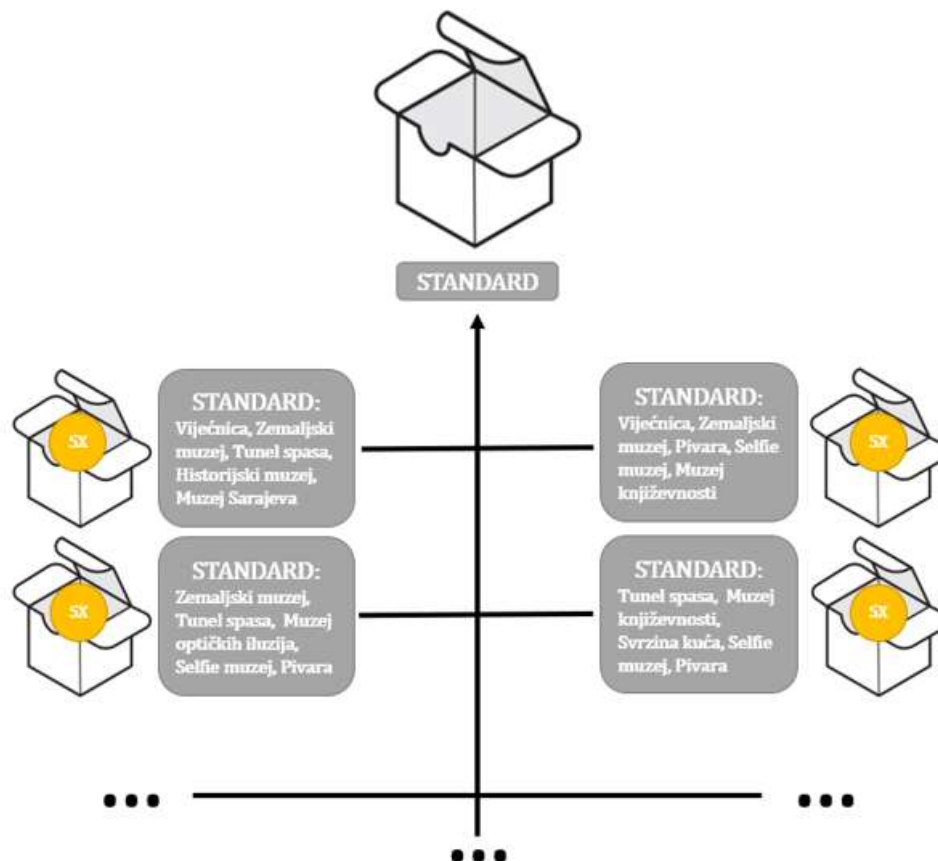
Slika 3. Struktura Factory method paterna

1.4. Abstract Factory

Navedeni patern možemo iskoristiti prilikom rezervacije kupovine suvenira i dobijanja odgovarajućeg koda kojeg će turisti predložiti uživo radi dobijanja popusta na licu mjesta prilikom kupovine. Kodovi se razlikuju domaćim i stranim turistima, odnosno imamo hijerarhiju klasa gdje bi apstraktna klasa bila naprimjer kod, a naslijeđeni objekti kod_za_stranca i kod_za_BiH.

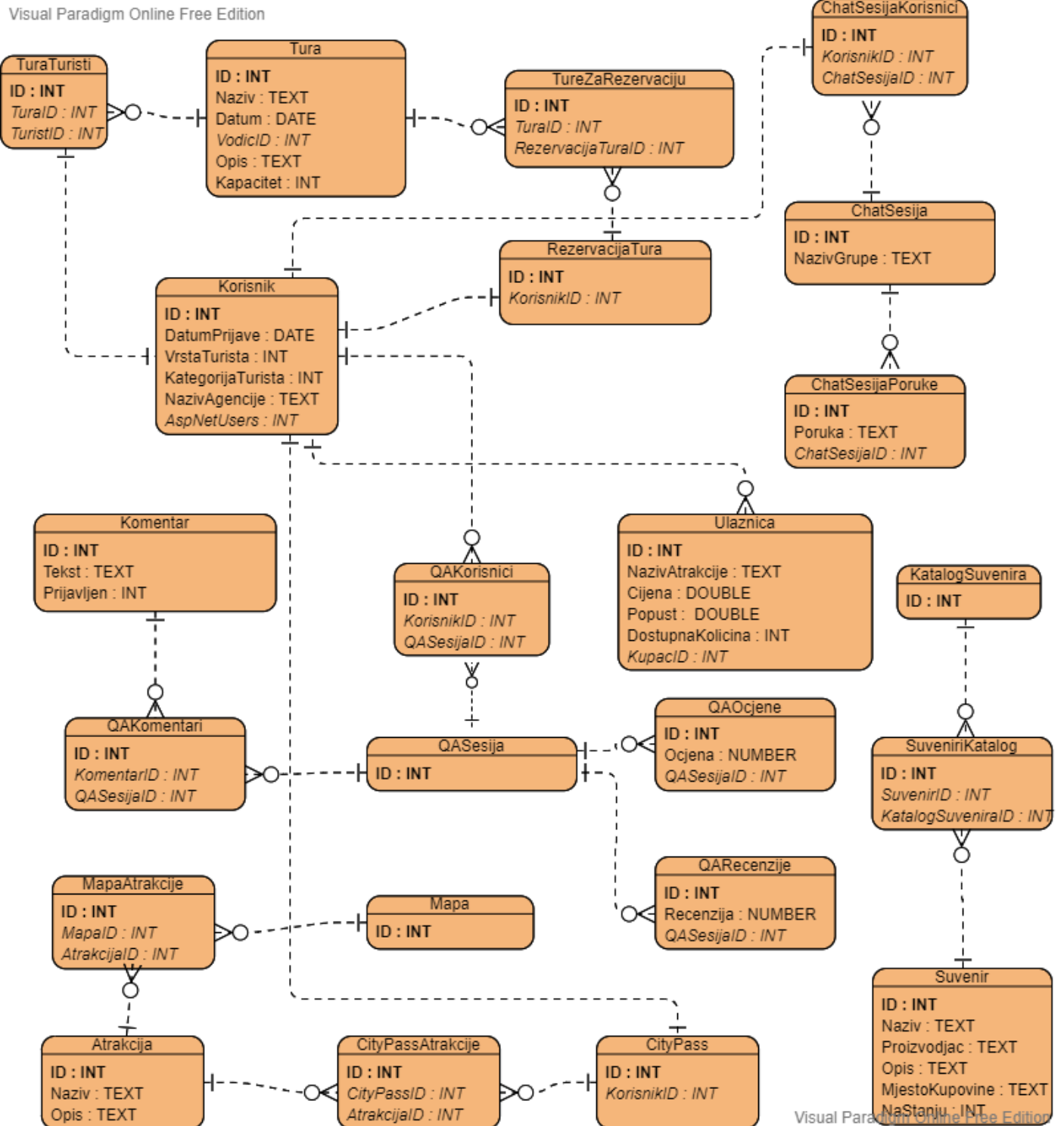
1.5. Builder

Patern Builder možemo povezati sa City Pass opcijom, ali indirektnim putem, odnosno njegova implementacija se tiče STANDARD, odnosno VIP paketa. Dakle, primjenjujemo ga na dva mjesta, u dva različita paketa pri čemu svaki od paketa sada posmatramo zasebno i neovisno jedan od drugog. Objasniti ćemo primjenu ovog patern na STANDARD paketu, a primjena Builder patern na VIP paketu je analogna u potpunosti. Ukoliko imamo jedan paket, cijena tog paketa je fiksna, a korisnik (klijent) može sam da bira šta sve želi da se nalazi u tom paketu, odnosno koje sve atrakcije želi da posjeti za ponuđenu cijenu paketa pri čemu je ograničen na broj odabranih atrakcija. Na ovaj način svaki turist ima mogućnost da napravi sebi jedinstven i karakterističan paket koji najviše odgovara njegovim zahtjevima.



Slika 4. Ilustracija Builder patern

2. Entity relationship dijagram



Slika 5. Entity relationship dijagram