Univerzitet u Sarajevu
Elektrotehnički fakultet u Sarajevu
Odsjek za računarstvo i informatiku

Paterni ponašanja

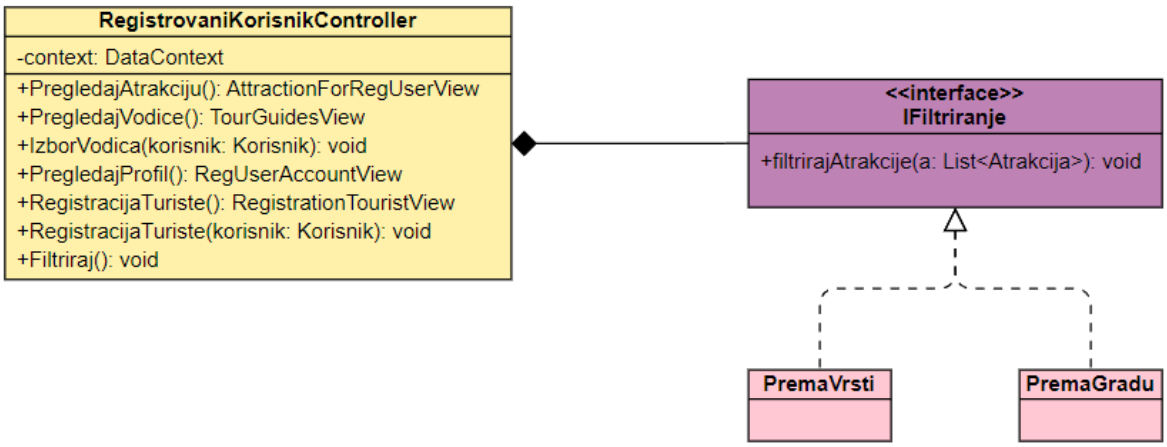
Objektno orijentisana analiza i dizajn

Članovi tima:
Nađa Alijagić
Hana Biševac
Anida Nezović

1. Strategy patern

Uloga Strategy paterna jeste da izdvaja algoritam iz matične klase i uključuje ga u posebne klase. Ovaj patern omogućava klijentu izbor jednog od algoritma iz familije algoritama za korištenje.

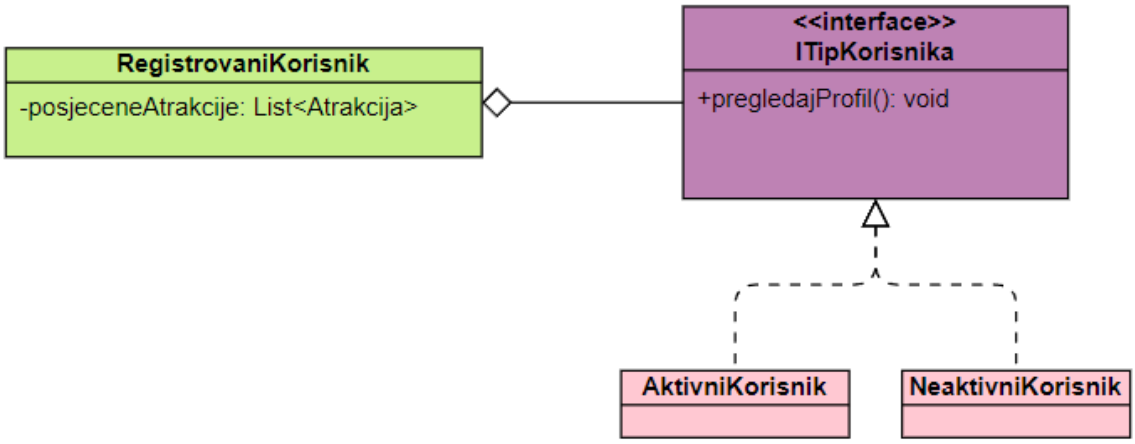
Kako se Strategy patern koristi da olakša dodavanje strategija kako će se nešto ponašati, mi ćemo ga iskoristiti prilikom filtriranja atrakcija prema vrsti ili prema gradu.



2. State patern

State patern je dinamička verzija Strategy paterna. Objekat mijenja način ponašanja na osnovu trenutnog stanja. Ovaj patern uklanja probleme sa razbacanim if iskazima u programu koji bi se koristili za ispitivanje koje je stanje.

U našem sistemu, imamo dvije ideje kako bismo mogli primijeniti ovaj patern. Prva ideja jeste da omogućimo da svaki registrovani korisnik može izabrati jedan od dva mode-a profila. Jedan mode bi bio privatni, a drugi javni. Dakle, imali bismo interfejs *IMode* iz kojeg su naslijeđene klase *PrivatniMode* i *JavniMode*. Sam interfejs bi bio povezan sa *RegistrovaniKorisnikController*. Druga ideja jeste da imamo dva tipa registrovanih korisnika. Prvi tip bi bili oni koji su aktivni i imaju mnogo posjećenih mjesta, a drugi bi bili oni manje aktivni. Implementirali smo drugu ideju.



3. Template method patern

Template method pattern se zasniva u odvajanju pojedinih koraka neke metode ili algoritma u podklase. Time omogućujemo da taj korak ima različitu implementaciju za različite podklase. Ovaj patern bismo mogli iskoristiti na način malo izmjenimo sistem i da ograničimo da neregistrirani korisnik nema pristup svim detaljima i zanimljivosti atrakcije. U klasi Korisnik bismo implementirali metodu prikažiDetalje(), koja bi prikazivala radno vrijeme i lokaciju atrakcije. Unutar te metode bi se nalazila dodatna metoda prikažiZanimljivosti(). Unutar klase Registrovani korisnik, ta metoda bi bila implementirana tako da se korisniku prikazuju sve poznate zanimljivosti o znamenitosti, dok bi NeregistrovaniKorisnik implementirao metodu tako da mu se ispirše da se registruje ili prijavi na sistem za dodatne informacije.

4. Observer patern

Observer pattern omogućuje uspostavljanje relacije između objekata, tako da se pri promjeni stanja jednog objekta šalje obavijest svim objektima koji su vezani za njega. U našem sistemu bi se ovaj pattern mogao primijeniti na način da da se svim korisnicima prijavljenim za turu šalju informacije o prijavljenoj turi, npr o predviđenom vremenu završetka ture, te bi obavijestili korisnika ukoliko bi došlo do neke izmejen za turu.

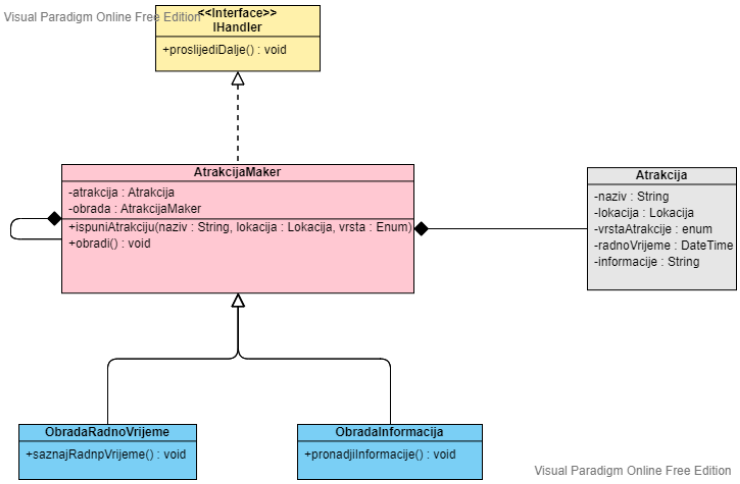
5. Iterator patern

Iterator patern omogućava sekvencijalni pristup elementima kolekcije bez poznavanja kako je kolekcija strukturirana. Ovaj patern bismo mogli iskoristiti

ako bi omogućili korisniku da pregleda svoje fotografije kao album, s tim da je slaganje fotografija u album po abecednom redu imena lokacija, po starosti fotografije ili po nekom slučajnom (random) prikazu.

6. Chain of responsibility patern

Chain of responsibility pattern je patern ponašanja koji nam omogućuje prosljeđivanje zahtjeva duž lanca handler-a. Nakon primanja zahtjeva, svaki handler odlučuje ili obraditi zahtjev ili ga proslijediti sljedećem handleru u lancu. Ovaj pattern bi mogli iskoristiti ako Administrator želi da automatski doda novu Atrakciju, napisat će osnovne informacije o njoj (naziv atrakcije, lokacija, vrsta atrakcije), a AtrakcijaMaker će dodati i popuniti ostale attribute koji su potrebni.Uz pomoć lokacije i naziva, pronaći će radno vrijeme atrakcije i osnovne informacije o istoj.



7. Medijator patern

Medijator patern definira objekt koji kapsulira kako skup objekata međudjeluje. Ovaj patern se smatra paternom ponašanja zbog načina na koji može promijeniti ponašanje programa. Mi smatramo da naša sistem nema potrebu za ovim paternom. Međutim, mogao bi se implementirati kada bi naš sistem tražio unos godina te bi se tada mogle staviti restrikcije na prijavljivanje maloljetnih korisnika na ture bez pratnje staratelja ili neke vrste punomoći (koja bi zaštitila vodiča u slučaju bilo kakvih nezgoda). Punoljetni korisnici se mogu prijavljivati na ture slobodno bez ikakvih restrikcija.