

Strukturalni paterni

Facade - ?

Adapter – Ovaj patern bi se u našem slučaju mogao iskoristiti tako da umjesto gettera i settera imamo adapter naslijeđen iz klase RegistrovaniKorisnik te da implementiramo interfejs koji ima metodu za vraćanje liste rezervacija.

Decorator – Mogli bi smo dodati da se mapa može prikazati na više načina (reljef, saobraćaj, satelit)

Bridge – Postoji mogućnost dobivanja raznih vrsta popusta u zavisnosti od uslova koje korisnik ispunjava. Korisnik (zaposlenik) želi da dobije informaciju o ukupnoj cijeni parkinga za određenog korisnika. Kako se svi popusti međusobno razlikuju mogli bismo napraviti klasu bridge i interfejs ICijena koji ima različite implementacije za svaki popust.

Composite – Potrebno je uvesti mogućnost pregleda slobodnih mjesta za različite kategorije vozila. Korisnik želi da pomoću jedne metode dobije dostupne parkinge za sve kategorije vozila. Možemo napraviti interfejs 'IKategorija' da na osnovu proslijeđenih informacija odabere odgovarajuću metodu koja vraća dostupna parking mjesta.

Proxy – u našem sistemu ovaj patern bi se mogao iskoristiti u svrhu provjere pristupnih podataka, ograničavanja prava pristupa određenim funkcionalnostima sistema u zavisnosti od toga da li je korisnik registrovan ili ne, da li je admin itd. Da bismo to postigli dodan je interfejs 'IProxyLogin' i klasa ProxyLogin koja kroz metodu login vrši identifikaciju korisnika na osnovu korisničkog imena.

Flyweight – Ovaj patern možemo iskoristiti da se omogući svim korisnicima da je defaultna kategorija vozila automobil i da se u tom slučaju, ako korisnik ne izabere drugu kategoriju, prikazuju samo dostupni parkinzi za automobile.

U naš sistem smo dodali Bridge i Composite pattern.