

# Strukturalni paterni

## 1.ADAPTER

Adapter patern služi da se postojeći objekti prilagode za korištenje na neki novi način u odnosu na postojeći rad, bez mijenjanja same definicije objekta. Na taj način se obezbjeđuje da će se objekti i dalje moći koristiti na način kako su dosad upotrebljavani, ali u isto vrijeme se omogućava njihovo prilagođavanje novim uslovima. Nakon razmatranja našeg sistema, uočile smo mogućnost korištenja adapter patrena prilikom učitavanja vozačke dozvole. Dovoljno bi bilo da korisnici skeniraju vozačku, a da se podaci sa vozačke unesu u sistem pomoću adaptera. Nakon toga bi sistem radio sa klasičnim podacima, koji bi bili učitani iz vozačke.

## 2.FACADE

Fascade patern služi da se korisniku olakša korištenje kompleksnih sistema. Klijent vidi samo krajnji objekat, dok je njegova struktura skrivena. Ovo omogućava da korisnik radi sa sistemom, bez da detaljno poznaje kako sistem funkcioniše. Facade patern u našem sistemu je primjenjen, s obzirom na to da korisnik vidi samo određeni dio sistema koji je njemu potreban.

## 3.DECORATER

Decorator patern služi za omogućavanje različitih nadogradnji objektima koji imaju istu osnovu. Umjesto da se definiše veliki broj izvedenih klasa, dovoljno je omogućiti različito dekoriranje objekata, te se na taj način pojednostavljuje i rukovanje objektima, ali i samo implementiranje modela objekata. Mi smo ovaj patern iskoristile prilikom izajmljivanja vozila, gdje korisnik bira da li će iznajmiti profesionalnog vozača, ili će on sam upravljati vozilo ukoliko posjeduje vozačku dozvolu.

## 4.PROXY

Proxy patern služi za dodatno osiguravanje objekata od pogrešne ili zloupotrebe. Primjenom ovog paternu omogućava se kontrola pristupa objektima, te se onemogućava manipulacija objektima ukoliko neki uslov nije ispunjen, odnosno ukoliko korisnik nema prava pristupa traženom objektu. Ovaj patern ćemo iskoristiti na način da pregled historije iznajmljivanja imaju samo taj klijent i osoba koja je prijavljena kao administrator sistema. Na taj način obezbjeđujemo da svaki klijent može pregledati samo svoju historiju iznajmljivanja.

## 5.COMPOSITE

Composite patern služi za kreiranje hijerarhije objekata. Koristi se kada svi objekti imaju različite implementacije nekih metoda, no potrebno im je svima pristupiti na isti način, te se time pojednostavljuje implementacija. Ovaj patern nismo nigdje iskoristili, ali bi se eventualno mogao iskoristiti u slučaju proširenja sistema. Kada bi, pored administratora, imali drugo uposleno lice, mogli bismo pomoću ovog paternu pregledati plate različitih vrsta uposlenih. Kako je jedni uposlenik administrator, ovaj patern ne koristimo.

## 6.FLYWEIGHT

Flyweight patern se koristi kako bi se onemogućilo bespotrebno stvaranje velikog broja instanci objekata koji svi u suštini predstavljaju jedan objekat. Samo ako postoji potreba za kreiranjem specifičnog objekta sa jedinstvenim karakteristikama, tada se on instancira, dok se u ostalim slučajevima koristi postojeća instanca. Ovaj patern možemo iskoristiti kada se iznajmljuje isto vozilo više puta. Svaki put kada se iznajmljuje vozilo će se koristiti ista instanca tog vozila, umjesto da se kreira novi objekat.

## **7.BRIDGE**

Bridge patern služi kako bi se apstrakcija nekog objekta odvojila od njegove implementacije. Ovaj patern je veoma važan jer omogućava ispunjavanje Open-Closed Solid principa, odnosno uz poštovanje ovog paternu omogućava se nadogradnja modela klasa u budućnosti te osigurava da se neće morati vršiti određene promjene u postojećim klasama. U našem sistemu ovaj patern bi se mogao iskoristiti u obračunavanju popusta za iznajmljivanje, mogli bi imati iznajmljivanje sa popustom i iznajmljivanje bez popusta koje bi koristili pri izračunavanju cijene iznajmljivanja.