

## STRUKTURALNI PATERNI

- **Adapter pattern**

Adapter patern služi da se postojeći objekat prilagodi za korištenje na neki novi način u odnosu na postojeći rad, bez mijenjanja same definicije objekta. U našem slučaju ovaj pattern bi se mogao relizovati dodavanjem adaptera za kupovinu karata na autobuskoj stanici tj. šalterskoj prodaji. Prodavač bi bio loginovan kao kupac, samo sa različitim pogledima i metodama u odnosu na običnog kupca.

- **Facade pattern**

Ovaj patern koristimo kada nije potrebno koristiti i poznavati cijeli sistem već koristiti određeni dio funkcionalnosti. Konkretno, vozač može dodavati alternativne rute na Google mapama dodavanjem koordianata početne i krajnje tačke puta , pa time nije potrebno da poznaje realizaciju kompletne mape, već se koristi samo njen dio.

- **Decorator pattern**

Ovaj patern koristimo kada želimo modifikaciju vršiti pomoću postojećih klasa, bez dodavanja većeg broja novih. Ovaj patern je iskorišten u kreiranju vožnje. Pri kreiranju klase vožnja kaskadno učestvju klase ruta, veza, stanica i termin. Prema tome omogućena je modifikacija vožnje na različitom nivou.

- **Bridge pattern**

Pattern omogućava da se iste operacije primjenjuju nad različitim podklasama. Bridge pattern je realizovan kod pretrage vožnji. Metode za pretragu vožnje su iste, ali su implementirane različito za različite korisnike.

- **Proxy pattern**

Proxy patern služi za dodatno osiguravanje objekata od pogrešne ili zlonamjerne upotrebe. Heširanjem e-maila i passworda korisnika sistema, osiguravamo korisnike od krađe njihovih profila.

- **Composite pattern**

Omogućuje pozivanja iste metode nad različitim objektima sa različitim implementacijama. Ovaj patern nije relizovan u našem projektu u potpunosti. Iako je zadovoljena hijerarhija klasa, nije omogućeno pozivanje iste metode nad različitim objektima. U slučaju potrebe ovaj pattern je moguće realizovati.

- **Flyweight pattern**

Ovaj patern podrazumijeva mogućnost ponovne upotrebe istih objekata. Prilikom kreiranja ili modifikacije vožnji moguća je ponovna upotreba već kreiranih stanica i veza između istih. Time je omogućena manja potrošnja memorije.