



Univerzitet u Sarajevu  
Elektrotehnički fakultet Sarajevo  
Odsjek za računarstvo i informatiku



## Osmi projektni zadatak

Objektno orijentisana analiza i dizajn

Ime i prezime:                   Din Švraka 18857  
  Nejla Bečirspahić 18835  
  Amina Pandžić 18808  
Grupa:                           šest (6) / DNACityGuide  
Datum:                           27.05.2022.

## 1. Strukturalni paterni

### 1.1. Adapter

Adapter pattern nam može poslužiti prilikom odabira i kupovine ulaznica za neke atrakcije. Turisti sa državljanstvom Bosne i Hercegovine će imati mogućnost pristupa funkcionalnosti odabira željene atrakcije i kupovine iste. Budući da će i turisti stranci moći izvršiti istu funkcionalnost, to znači da je najlakše uvesti odgovarajući adapter koji će samo „pretvoriti“ domaćeg u stranog turistu, a odabir i kupovina ulaznica se neće mijenjati. Dakle uspješno će biti omogućeno prilagođavanje turista iz različitih kategorija za iste funkcionalnosti.

### 1.2. Facade

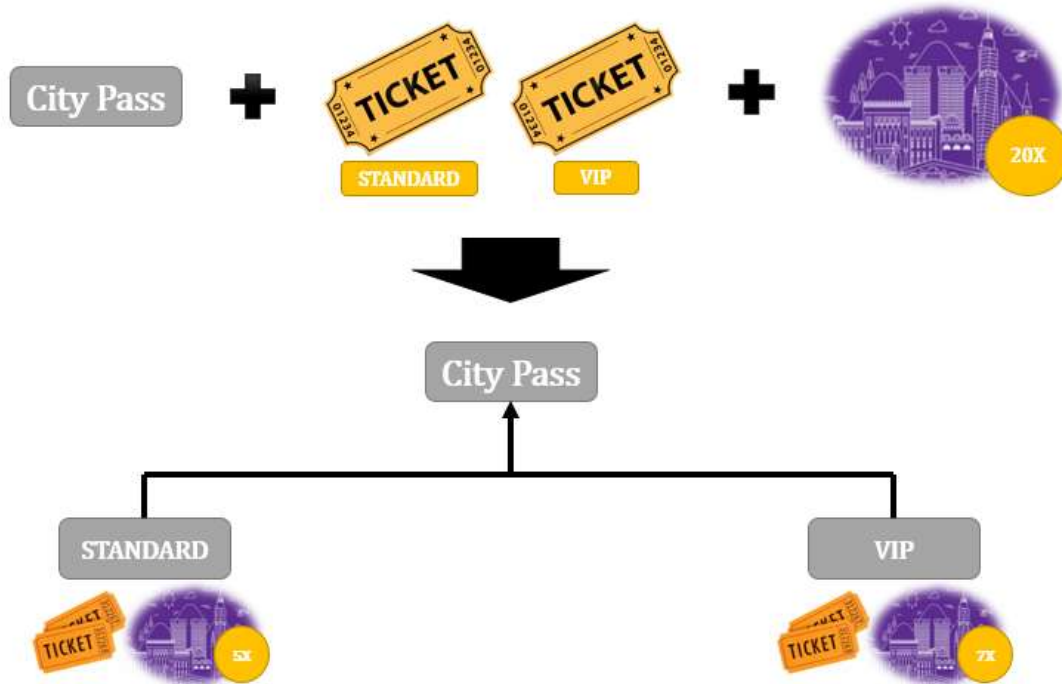
Facade pattern služi za osiguravanje pogleda visokog nivoa na podsisteme. Apstrakcije i implementacije tih podsistema su usko povezane. U našem projektu, uočavamo navedenu povezanost između klasa KupovinaUlaznicaController i RezervacijaTuraController. Upravo nad ovim klasama bismo mogli iskoristiti facade pattern, na način da funkcije KupovinaUlaznicaController i RezervacijaTuraController postaju podsistemi, a dodajemo klasu Facade koja objedinjuje ova dva podsistema. U klasu Facade dodajemo sve metode koje se koriste u navedenim klasama, a za attribute Facade klase imamo instance ova dva podsistema.

### 1.3. Decorator

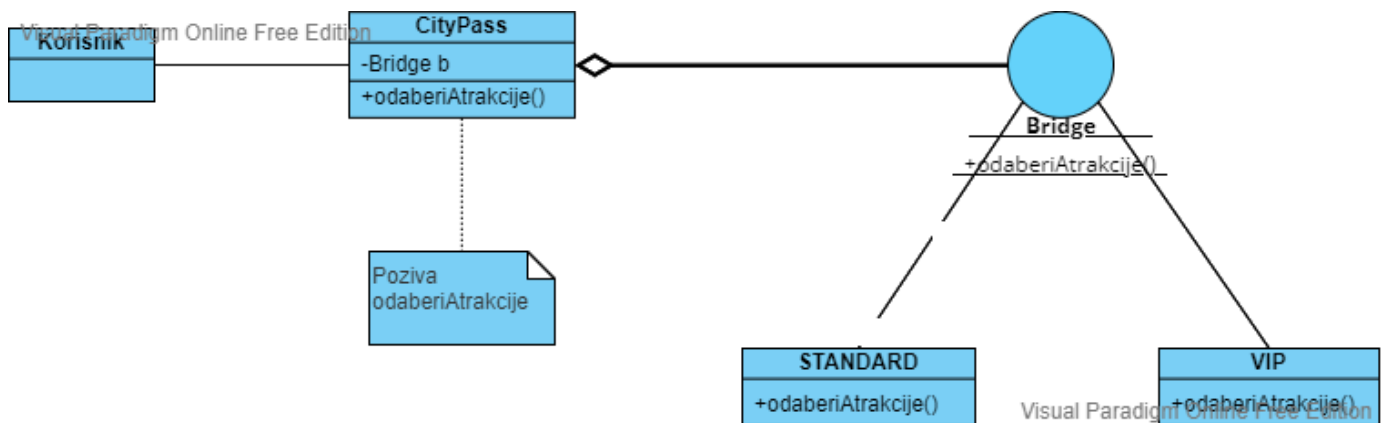
U ovoj aplikaciji će biti omogućeno administratoru da postavlja video snimak. Putem decorator patterna bismo mogli omogućiti administratoru editovanje video snimka putem aplikacije. Odnosno mogli bismo imati klasu VideoEdit u kojoj bi bilo omogućeno rotiranje, promjena veličine, skraćivanje videa itd, a s druge strane bismo mogli imati klasu VideoFilteri koja bi administratoru služila za primjenu nekih već predodređenih filtera nad video zapisom.

### 1.4. Bridge

Bridge patern možemo iskoristiti da bismo omogućili da se iste operacije primjenjuju nad različitim podklasama, što je ovdje konkretno biranje STANDARD ili VIP paketa te odabir 5, odnosno 7 atrakcija od ponuđenih 20 u sklopu jednog paketa. Time je postignuto izbjegavanje kreiranja novih metoda za već postojeće funkcionalnosti, a kupovina (plaćanje) je također jednako primjenjivo na obje mogućnosti. Ovo je veoma pogodno jer je upravo omogućena i jedna od osnovnih uloga i važnosti Bridge patterna, a to je mogućnost implementacije nove verzije softvera, pri čemu postojeća ostaje u funkciji. Ilustracija je data u nastavku, a pobliže objašnjava svrhu ovog patterna.



Slika 1. Ilustracija Bridge paternna



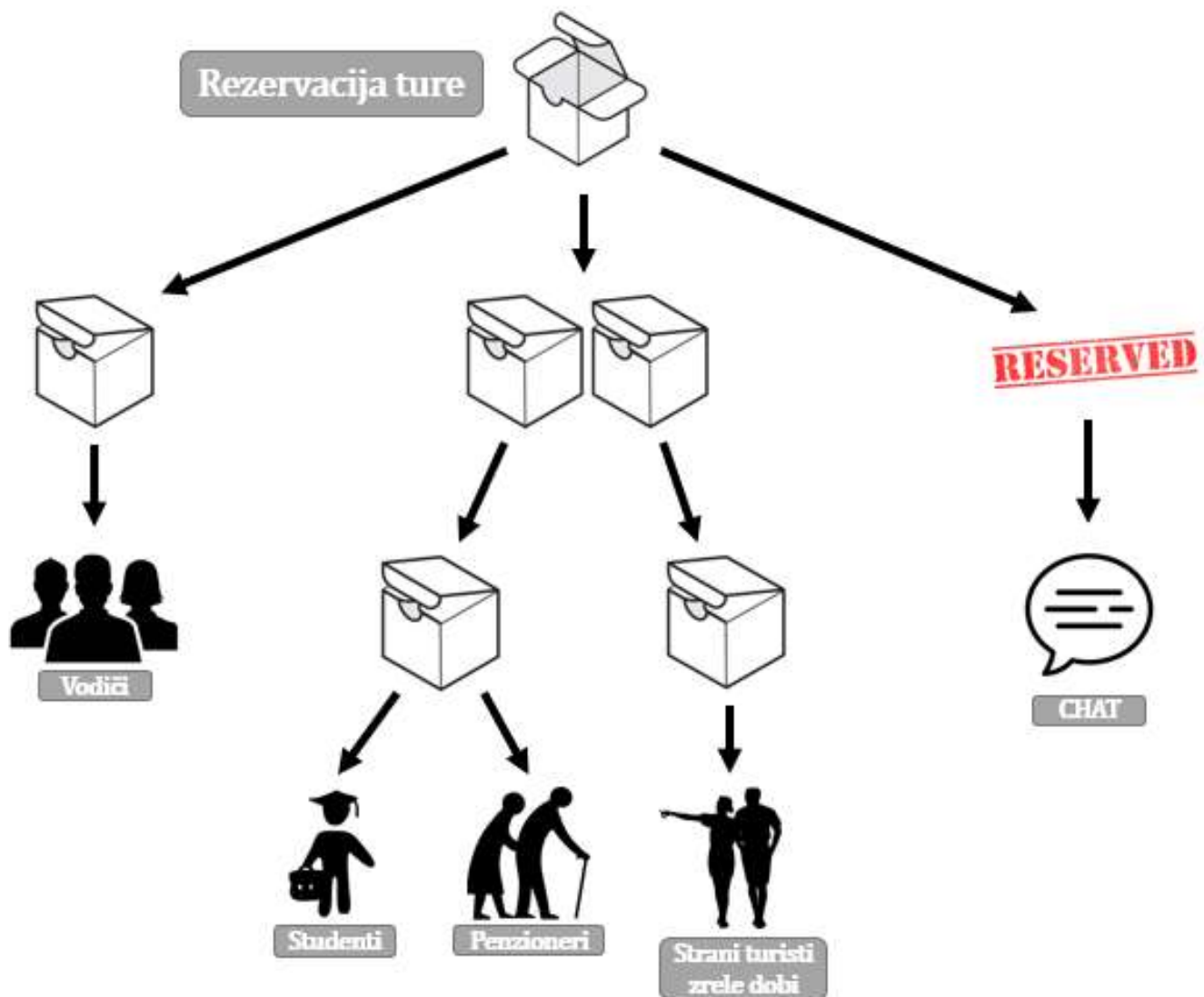
Slika 2. Struktura Bridge paternna

## 1.5. Proxy

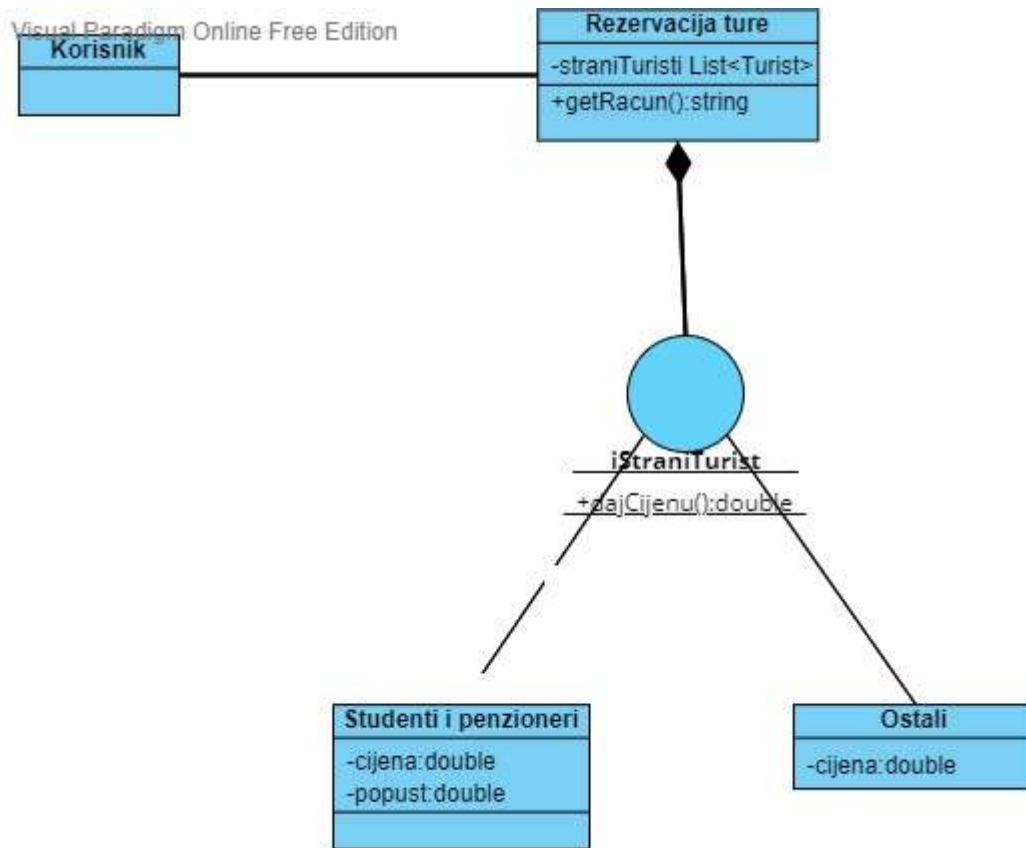
ProxyPattern možemo iskoristiti u interakciji korisnika sa QASesijom, obzirom da u ovoj klasi postoje određene funkcionalnosti kojima imaju pravo pristupa samo administratori sistema. Također, putem ProxyPatterna ovdje možemo regulisati koje vrste korisnika uopšte imaju pravo pristupiti QASesiji.

## 1.6. Composite

Kao što je i prikazano na slici 2. Composite patern nam je veoma važan budući da imamo svojevrsnu hijerarhiju klasa, odnosno omogućena su povezivanja istih metoda nad različitim objektima sa različitim implementacijama. Konkretno, u ovom slučaju su to vodiči, obračuni popusta za studente i penzionere, odnosno nekorigiranje cijena za ostale strane turiste, potvrde o (ne)uspješnim rezervacijama tura kao i implementacija CHAT sobe nakon prikazane potvrde. Upravo tako je formirana struktura stabla pomoću klasa, a individualni objekti (na slici ilustrovani kao listovi stabla) i kompozicije individualnih objekata (kutije kao korijeni stabla) jednako su tretirani.



Slika 3. Ilustracija Composite paterna

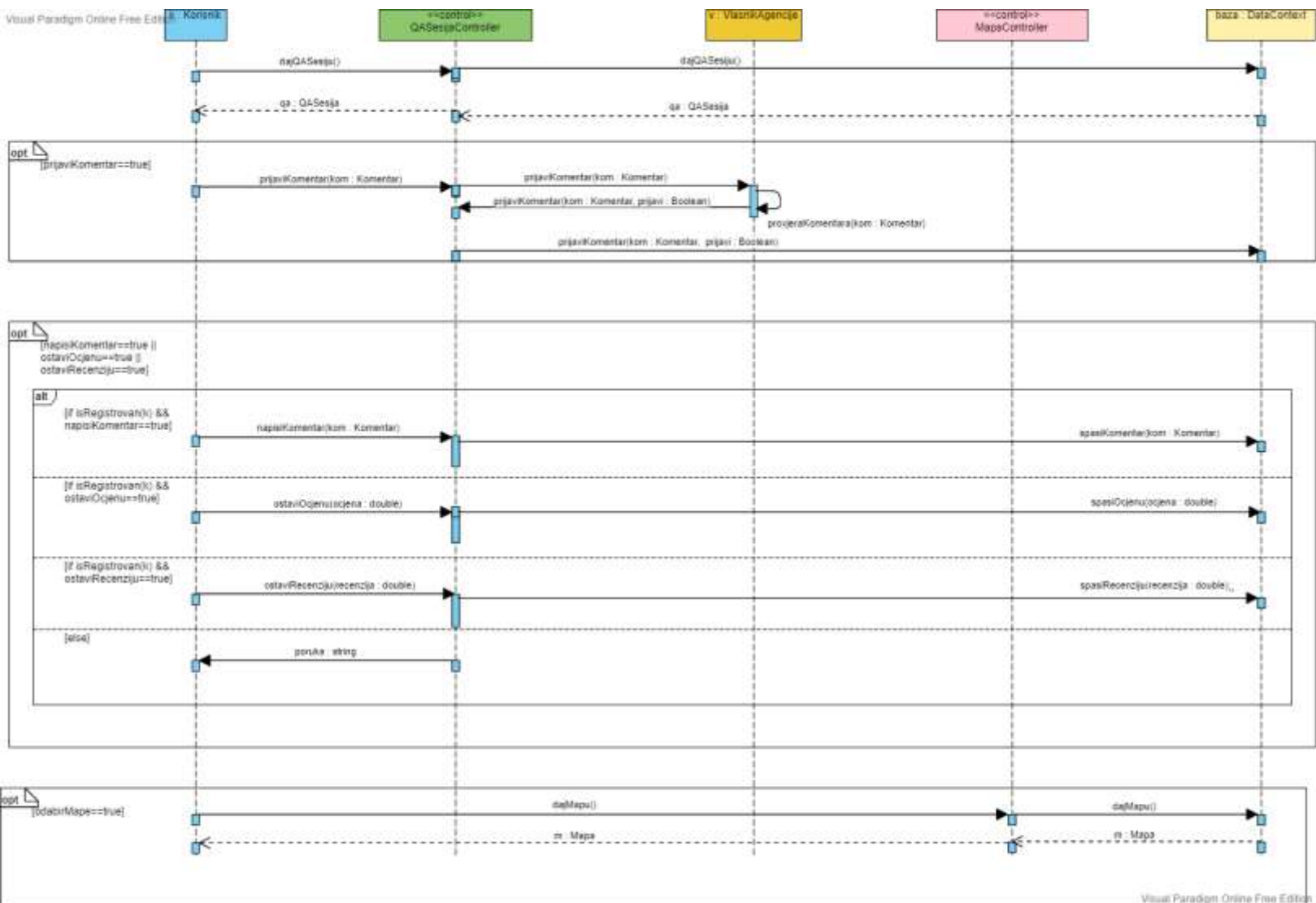


Slika 4. Struktura Composite paterna

### 1.7. Flyweight

Flyweight pattern bismo mogli iskoristiti za klasu Komentar. Odnosno mogli bismo imati određenu listu standardnih komentara, kao što su „Odlično iskustvo!“, „Bilo je zaista prelijepo!“ i slično. Pored toga što bismo na ovaj način uštedjeli na memoriji, imali bismo i prednost u tome što bi njihov atribut „prijavljen“ po defaultu bio false, i administrator ovakve komentare ne bi trebao provjeravati da li su neprikladnog sadržaja.

## 2. Dijagram sekvence



Slika 5. Dijagram sekvence