

Strukturalni paterni

Merjem Bećirović

Stefani Kecman

Bakir Bajrović

1. Adapter patern:

U našem sistemu smo predvidjeli da zaposlenik sistema, na kraju svake utrke, rezultate ručno unosi kroz predviđeni interfejs, demonstriran na jednom od ranijih taskova. Međutim, bilo bi korisno kada bi on mogao preuzeti tabelu rezultata sa nekog provjerenog izvora i ubaciti je u sistem za obradu. S obzirom da naš prvobitni sistem prihvata pojedinačne podatke, u ovom bi slučaju bilo potrebno uvesti adapter, koji će algoritamski čitati podatke iz tabele i interpretirati ih na način prihvatljiv našem sistemu, pri čemu mi ostavljamo i prvobitnu mogućnost ručnog unosa.

2. Facade patern:

Običnog korisnika sistema će zanimati tabelarni prikaz statistike, leaderboard ekipa, vozača i slično. Iza njih se nalazi kompleksan sistem unosa i obrade podataka koji se može grupisati na sljedeći način: unos vozača i unos ekipa (prije početka sezone), unos rezultata ostvarenih do datog momenta, unos i sabiranje novih bodova ako je jedna utrka završena, ažuriranje sume bodova za vozača i za ekipe, algoritam za sortiranje poretka na tabelama nakon unosa dodatnih bodova (tj. promjena prikaza mjesta na ljestvici ukoliko se ona zaista dogodi). Nijedan ovaj korak ne zanima običnog korisnika, a neophodan je da se pruži mogućnost prikaza statistike i ljestvica, koja zanima svakog posjetioca. Samim tim, tabele koje oni vide su samo fasada i krajnji rezultat mnogobrojnih i kompleksnih procesa, koji su u funkciji različitih kontrolera i koji koriste metode raznih klasa.

3. Decorator patern:

F1 vozači su modelirani klasom Vozac, njenim atributima i metodama. Jedna od tih metoda je dodajBodove. Mi smatramo, bez obzira na pravilnik iz stvarnog života, da bi se u slučaju nepredvidivih okolnosti koje idu na štetu vozača, a na koje on nije mogao utjecati, bodovi trebali na određeni način skalirati. Ako se vozač povrijedi ili razboli (kao što je bilo slučajeva zbog covid pandemije), prinuđen je da ne učestvuje u utrkama, radi čega gubi bodove. Ideja koju imamo je da na neki način ublažimo taj gubitak bodova (s obzirom da ovaj patern nećemo implementirati, detaljan opis tog načina izostavljamo iz ovog segmenta). Dakle, mi bismo za

samo pojedinačne instance klase Vozac, dodali poseban omotač, koji bi sa sobom nosio funkcionalnost skaliranja bodova, i koji bi se primijenio samo na datu instancu, bez mijenjanja funkcionalnosti cijele klase.

4. Bridge patern:

Sistem poznaje dvije vrste registrovanih korisnika, običnog i premium registrovanog korisnika. Oba imaju mogućnost kupovine karata za utrke Formule 1. Jedina razlika je u tome što premium korisnik ostvaruje popust na redovnu cijenu karte. Prilikom kupovine, zajedničko im je da biraju željenu utрку, zatim odaberu kategoriju sjedišta, sistem provjeri ima li dostupnih karata i, ukoliko ima, prikaže redovnu cijenu karte. To je most koji spaja proceduru kupovine kod ova dva tipa korisnika i on se u svakom slučaju izvršava. Razilaženje nastaje pri potvrdi kupovine, kada sistem provjerava da li je korisnik premium ili ne. Ako je riječ o premium korisniku, traži se unos njegovog premium koda, za verifikaciju, nakon čega se obračunava cijena koju će on platiti po formuli: $(1 - \text{popust}) * \text{redovna cijena}$ (napomena: popust nije u procentima već u segmentu $(0..1]$!) Ta se cijena prikazuje kao finalna. U suptornom, cijena za platiti je redovna cijena karte.

5. Composite patern:

I vozači i ekipe imaju svoje bodove, kao i metode za unos/obračun bodova. Ove klase će imati interfejs za tu svrhu, a razlikovat će se u tome što su bodovi ekipe drugačije definirani od bodova vozača. Bodovi jedne ekipe su suma svih bodova koje su ostvarili njeni vozači. Composite patern bi nam služio da tu razliku definiramo i korektno implementiramo pomenute funkcionalnosti.

6. Proxy patern:

Predviđeni način plaćanja karata za kupovinu je putem bankovnog računa. Pored provjera o novčanom stanju računa, potrebno je iz sigurnosnih razloga uvesti provjeru da li bankovni račun odgovara korisniku koji je trenutno ulogovan i vrši kupovinu, kao i da li je on verifikovan. Ovde bismo iskoristili proxy patern.

7. Flyweight patern:

Svaka utrka ima fiksni broj karata, raspoređen po unaprijed određenim kategorijama. U svakoj kategoriji ima više karata. Pri kupovini karte, mi ne pravimo razlike između individualnih karata iz iste kategorije. Dakle, ako korisnik kupuje kartu za kategoriju čija je jedna karta već instancirana, mi ćemo samo iskoristiti podatke od već napravljene karte iz te kategorije (uz promjenu broja karte koji će se prikazivati).