DIJAGRAM KLASA – POSTIVANJE SOLID PRINCIPA

**Klasa: Kupac**

Single Responsibility Principle (SRP): Klasa Kupac je odgovorna samo za upravljanje informacijama vezanim za kupca. To uključuje funkcionalnosti poput registracije, prijave, i uređivanja profila kupca.

Open/Closed Principle (OCP): Klasa može biti proširena novim funkcionalnostima (npr., dodavanje metode za praćenje historije kupovina) bez izmjene postojećeg koda.

Liskov Substitution Principle (LSP): Ako bi postojale podklase klase Kupac, kao što su VIPKupac ili OsnovniKupac, one bi trebale moći zamijeniti klasu Kupac bez utjecaja na postojeće funkcionalnosti.

Interface Segregation Principle (ISP): Ako bi interfejsi bili korišteni, Kupac bi mogao implementirati samo one koji su relevantni za njegovu funkcionalnost, izbjegavajući nepotrebno nasljeđivanje metoda koje ne koristi.

Dependency Inversion Principle (DIP): Klasa Kupac bi trebala zavisiti od apstrakcija, a ne od konkretnih implementacija, što se može postići korištenjem interfejsa za komunikaciju s bazom podataka.

**Klasa: Manifestacija**

Single Responsibility Principle (SRP): Klasa Manifestacija upravlja svim aspektima vezanim za manifestacije, uključujući kreiranje, uređivanje, i prikaz informacija o manifestacijama.

Open/Closed Principle (OCP): Ova klasa može lako proširiti svoje funkcionalnosti (npr., dodavanje funkcionalnosti za komentare ili ocjene) bez mijenjanja postojećeg koda.

Liskov Substitution Principle (LSP): Slično kao i kod klase Kupac, ako bi postojale specijalizirane podklase (npr., MuzickaManifestacija, SportskiDogađaj), one bi trebale moći zamijeniti Manifestacija klasu.

Interface Segregation Principle (ISP): Manifestacija bi trebala implementirati specifične interfejse za svoje operacije umjesto jednog velikog interfejsa, što pomaže u održavanju organizacije koda i smanjenju spregnutosti.

Dependency Inversion Principle (DIP): Trebala bi koristiti interfejse ili apstraktne klase za pristup podacima, osiguravajući time da visokonivojske operacije ne zavise direktno od detalja nižeg nivoa implementacije.

**Klasa: Karta**

Single Responsibility Principle (SRP): Klasa Karta se bavi kreiranjem, validacijom i otkazivanjem karata, svaka operacija je jasno definisana unutar klase.

Open/Closed Principle (OCP): Može se proširiti, na primjer, dodavanjem metoda za grupne popuste ili promocije bez izmjene postojećeg koda.

Liskov Substitution Principle (LSP): U slučaju podklasa kao što su VIPKarta ili GrupnaKarta, one bi trebale funkcionisati u kontekstu gdje se koristi originalna Karta.

Interface Segregation Principle (ISP): Klasa bi trebala implementirati interfejse koji su specifično dizajnirani za operacije koje izvršava, bez nepotrebnih zavisnosti.

Dependency Inversion Principle (DIP): Korištenjem interfejsa za pristup podacima, Karta ostaje nezavisna od konkretnih implementacija baza podataka ili drugih skladišta podataka.

**Klasa: Mjesto**

Single Responsibility Principle (SRP): Klasa Mjesto ima jednu osnovnu odgovornost – upravljanje informacijama i statusom pojedinačnih mjesta (broj reda, broj kolone, status zauzetosti). Ova odgovornost uključuje operacije kao što su rezervisanje i oslobađanje mjesta.

Open/Closed Principle (OCP): Klasa može biti proširena novim funkcionalnostima, poput dodavanja atributa za specifične vrste sjedišta (npr. sjedišta prilagođena osobama s posebnim potrebama) bez potrebe za mijenjanjem postojećeg koda.

Liskov Substitution Principle (LSP): Ako bi postojale podklase klase Mjesto, kao što su VIPMjesto ili StandardnoMjesto, one bi trebale moći zamijeniti Mjesto bez utjecaja na postojeće funkcionalnosti (npr. rezervacija i oslobađanje).

Interface Segregation Principle (ISP): Mjesto bi trebalo implementirati interfejse koji se tiču specifično njenih funkcionalnosti, omogućavajući time da ne implementira metode koje su nebitne za njen kontekst.

Dependency Inversion Principle (DIP): Klasa Mjesto bi trebala zavisiti od apstrakcija (npr. interfejs za upravljanje statusom mjesta), a ne od konkretnih implementacija, što povećava fleksibilnost i održivost koda.

**Klasa: SlobodnaMjesta**

Single Responsibility Principle (SRP): Klasa SlobodnaMjesta se fokusira na upravljanje kolekcijom dostupnih mjesta za određenu manifestaciju. To uključuje funkcionalnosti kao što su prikazivanje slobodnih mjesta za određene datume i događaje.

Open/Closed Principle (OCP): Klasa se može lako proširiti dodatnim funkcionalnostima poput filtiranja ili pretraživanja mjesta po različitim kriterijumima bez izmjene postojećeg koda.

Liskov Substitution Principle (LSP): Ako bi se razvile specijalizovane podklase (npr., za različite vrste događaja), one bi trebale moći zamijeniti SlobodnaMjesta u svim kontekstima gdje se koristi.

Interface Segregation Principle (ISP): SlobodnaMjesta bi trebala implementirati samo one interfejse koji su relevantni za upravljanje slobodnim mjestima, izbjegavajući nepotrebne zavisnosti od metoda koje ne koristi.

Dependency Inversion Principle (DIP): Klasa treba zavisiti od apstrakcija za dohvat podataka o mjestima, što može uključivati interfejse ili apstraktne klase koje definišu metode za rad s mjestima na različitim događajima.