**SOLID PRINCIPI ZA ULAZNICE.COM**

1. **SRP (Single Responsibility Principle)**

Ovaj princip nam je zadovoljen budući da svaka od navedenih klasa u dijagramu radi samo jedan posao. Operacije unutar klasa su jednostavne.

1. **OCP (Open – Closed Principle)**

Klasa *Placanje* je zastupljena u iduće 3 klase. To su *Gotovina*, *Vaucer* i *Kartica*, te njihova nadogradnja neće zahtjevati neko veće modifikovanje jer one zavise od klase *Placanje.* Klasa *Placanje* se može nadograditi jer ona ima samo jedan atribut a to je iznos koji se treba naplatiti – *saldo.* Klasa *Korisnik* sadrži samo informacije o korisniku i ne utiče na druge klase. Klase *Karta* i *Lokacija* zavise jedna od druge budući da je lokacija eventa sadržana na ulaznici. One se mogu modifikovati a ne nadograditi.

1. **LSP (Liskov Substitution Principle)**

Sigurno je da će se u budućnosti dodati i druge klase koje utiču na nivoe pristupa aplikaciji tako da će klasa *Korisnik* postati apstraktna i neće se moći instancirati primjerak te klase već će ona biti naslijeđena kroz klase *Administrator, ContentCreator, Moderator* i sl. Stoga, trenutno za naš sistem ovaj princip **(ni)je zadovoljen**\*.

1. **ISP (Interface Segragation Principle)**

Trenutno ne vidim nijedan mogući interfejs. Možda u klasama koje su zadužene za plaćanje *(Placanje, Gotovina, Vaucer i Kartica)* pa recimo metoda *validiraj* bi se mogla preinačiti u interfejs koji će implementirati gore navedene klase. Trenutno ovaj princip nije zadovoljen.

1. **DIP (Depedency Inversion Principle)**

Klasa Korisnik će u dogledno vrijeme postati apstraktna pa ćemo imati apstrakciju na nivou modela klase dijagrama. Stoga, trenutno za naš sistem ovaj princip **(ni)je zadovoljen**\*.