

Zadaci za vežbe iz predmeta

Softverski obrasci i komponente

Dependency injection

1 ZADATAK

Date su abstraktne klase **AbstractHero** i **AbstractVehicle** gde prva označava šta treba super heroj da poseduje od metoda, a **AbstractVehicle** prikazuje sta njegovo prevozno sredstvo može da uradi.

Napisati abstraktnu klasu **AbstractStory** koja će definisati abstraktnu metodu **tell_story()** i klasu **HeroVehicleStory** koja će nasleđivati klasu **AbstractStory** i koja u sebi sadrži dva atributa **hero** i **vehicle**. Klasa **HeroVehicleStory** treba da redefiniše metodu **tell_story()** u kojoj će se na konzoli prikazati koji je super heroj i koje je vozilo trenutno instancirano.

- U klasi **HeroVehicleStory** implementirati da se nove instance za heroja i vozilo dobijaju preko **AbstractFactory** šablona.
- Iskoristiti **injector** biblioteku za reimplementaciju ovog zadatka.
- Omogućiti da se koristeći odgovarajući dekorator umeću odgovarajući objekti super heroja i vozila za klasu **HeroVehicleStory**.
- Napraviti da postoji samo jedan heroj po klasi (**Singleton**)

2 ZADATAK

Implementirati 1. zadatak sa prvih vežbi **Kreacioni obrasci**. Mesto **AbstractFactory** koristiti **injector** biblioteku i **Dependency Injection** šablon. Za poziv metode koja kreira prozor može se koristiti **CallableProvider**.

3 ZADATAK

Napraviti implementaciju prodavnice, **Shop** klasa. Svaka prodavnica ima servis za plaćanje **IBillingService**, koji ima metodu **IResult charge_order(IOrder order, ICreditCard card)** za izvršavanje plaćanja. Svaki **IBillingService** sadrži implementaciju **ITransactionLog** za logovanje važnih informacija o plaćanju i **ICreditCardProcessor** koji zapravo procesira plaćanje. Metoda koja procesira plaćanje ima sledeći potpis **ChargeResult charge(ICreditCard card, float amount)**. Napraviti više implementacija datih interfejsa, a pomoću **Dependency Injection** šablona birati koja će se klasa koristiti u **Shop** klasi.