

Obrazac za zadaću na predmetu "**Uzorci dizajna**"

Ime i prezime studenta/ice: \_\_\_\_\_David Koprek\_\_\_\_\_

Matični broj: \_\_\_\_\_0016122816\_\_\_\_\_

**Dio A. Osnovni podaci o zadaći**

R.br.	Pitanje	Odgovor	
1.	Grupa na seminaru:	2	
2.	Broj i naziv zadaće:	1	Zadaća 3. Iznajmljivanje električnih vozila
3.	Procjena vremena za realizaciju bez decimala):	15 sati	
4.	Procjena % završenosti (bez decimala):	60 / 100%	
5.	Procjena bodova za izradu zadaće ( 1 decimala):	9/ (DZ1 - 12, DZ2 – 13, DZ3 – 15)	
6.	Žalim prezentirati zadaću:	NE	
7.	Koji dijelovi iz opisa zadaće nisu realizirani:	Ispis u tablicama, ispis kumulativa za brojčane vrijednost u aktivnostima 6 i 7, ispis u datoteku, MVC uzorak, ispis grešaka u crvenoj boji	
8.	Postoji li dio zadaće koji vrijedi posebno istaknuti i zašto:	Kod aktivnosti 7 i 8 <b>moraju</b> se upisati datumi (u formatu kao u primjeru) jer nema provjere i program se ruši	
9.	Postoje li dijelovi zadaće koji imaju pogrešku u radu i koje:	Spomenut upis datuma, kod aktivnosti 0-4 datum se mora napisati s navodnicima na početku i kraju	
10.	Da li ste koristili posebne biblioteke u realizaciji zadaće izvan spomenutih na nastavi:	Ne	
11.	Da li ste koristili programska rješenja ili dijelove programskog koda od drugih kolega:	Ne	

## Dio B. Dokumentacija rješenja zadaće

Naziv uzorka dizajna	Klase koje sudjeluju u uzorku dizajna	Opis razloga odabira uzorka dizajna
Singleton	Data	Dobivanje jedne instance koja će sadržavati sve podatke učitane iz datoteka. Na ovaj se način može pristupiti tim podacima iz bilo kojeg dijela aplikacije.
Singleton	VirtualTime	Pamćenje virtualnog vremena prebačeno u Singleton kako bi mu se moglo pristupiti i mijenjati bez da se prosljeđuje kroz klase.
Factory Method	Loader, LoaderFactory, OrganizationalUnitObjectLoader, OrganizationalUnitObjectLoaderFactory	Zbog toga jer program ne može biti siguran iz kojih će sve datoteka morati učitavati podatke (ponajprije se odnosi na mogućnost izostanka aktivnosti.txt) i činjenice da svaka datoteka ima različite podatke ovaj se uzorak dizajna činio praktičnim. Nadograđen s loader-om za učitavanje sadržaja <i>tvrtka.txt</i> datoteke.
<b>Composite</b>	Location, OrganizationalUnit, OrganizationalUnitInterface	Odabran je Composite uzorak jer on najbolje može predstaviti strukturu (hijerarhiju) tvrtke – kao stablo s granama (OrganizationUnit) i listovima (Location).
<b>State</b>	<b>Vehicle, VehicleState, VehicleStateBroken, VehicleStateCharging, VehicleStateFree, VehicleStateRented</b>	<b>Kako bi se izrazila različita stanja u kojima vozilo može biti i na temelju kojih mijenja ponašanje implementiran je State uzorak.</b>
<b>Iterator</b>	<b>Iterable, Iterator, DepthIterator, OrganizationalUnit</b>	<b>Za prolazak kroz stablo uzorka Composite implementiran je Iterator uzorak.</b>
<b>Chain of Responsibility</b>	<b>AbstractHandler, IHandler, CreateBillHandler, PrintBillHandler</b>	<b>Za rukovanje aktivnostima vezanih uz račune (kreiranje/izdavanje i ispis računa korisnika) koristi se Chain of Responsibility kojemu se pošalje aktivnost (zahtjev) koji onda „putuje“ do rukovatelja (Handler-a) koji s njom zna rukovati.</b>

## Dio C. Opis promjena u odnosu na prethodnu zadaću (osim kod 1. zadaće)

Dorađeno je učitavanje i čitanje parametara zbog dodanih parametara (dugovanje, izlaz). Implementiran uzorak **State** – dodane klase i nadograđena klasa *Vehicle* kao dio uzorka. Implementiran uzorak **Iterator** – dodane klase. Implementiran uzorak **Chain of Responsibility** – dodane klase.

*ActivityExecutor* promijenjen da koristi nove uzorke i nadograđen za nove aktivnosti 9,10 i 11.

Nadograđena klasa *Person* – polja *Debt* i *HadRent*.

Nadograđena klasa *PersonLoader* – postavljanje duga na nulu kod učitavanja.

Nadograđena klasa *Database* – pamćenje maksimalnog dopuštenog duga.

Promijenjena klasa *Program* – formatiranje ispisa i provjere parametara.

Nadograđena klasa *ActivityLoader* za podršku novim aktivnostima 9,10 i 11.

## Dio D. Dijagram klasa s naglašavanjem klasa koje sudjeluju u pojedinom uzorku dizajna

