

Obrazac za zadaću na predmetu "**Uzorci dizajna**" ak. god. 2023./2024.

Ime i prezime studenta/ice: Marko Mušica

Matični broj: 0016142667

Dio A. Osnovni podaci o zadaći

R.br.	Pitanje	Odgovor	
1.	Grupa na seminaru:	G2	
2.	Broj i naziv zadaće:	2	Dostava paketa po područjima
3.	Procjena vremena za realizaciju bez decimala):	24 sati	
4.	Procjena % završenosti (bez decimala):	80 / 100%	
5.	Procjena bodova za izradu zadaće (1 decimala):	9 / (DZ2 - 13)	
6.	Želim prezentirati zadaću:	DA	
7.	Koji dijelovi iz opisa zadaće nisu realizirani:	Dostava radi na način iz zadaće 1, znači nije implementirana nova dostava. Osim ta 2 nova načina dostave još nisu obavljene sve provjere. Ne radi naredba VS jer dostava u segmentima nije implementirana. Bitno je napomenuti da je ukrcavanje paketa prema lokaciji napravljeno samo nova dostava nije.	
8.	Postoji li dio zadaće koji vrijedi posebno istaknuti i zašto:	NE	
9.	Postoje li dijelovi zadaće koji imaju pogrešku u radu i koje:	Za osobu koja ne postoji u inicijalnim datotekama ne dobivaju se obavijesti jer ne postoji u sustavu.	
10.	Da li ste koristili tuđi programski kod u realizaciji zadaće izvan spomenutih izvora na nastavi:	NE	
11.	Da li ste koristili programska rješenja ili dijelove programskog koda od drugih kolega:	NE	

Dio B.1. Dokumentacija rješenja 1. zadatka (kopirano i nepromijenjeno)

Naziv uzorka dizajna	Klase koje sudjeluju u uzorku dizajna	Opis razloga odabira uzorka dizajna
SINGLETON	TerminalCommandHandler	Podaci u ovoj klasi trebaju biti dostupni skoro svakoj klasi u cijeloj zadaći i nema potrebe da se više puta instanciraju. Da nisam koristio SINGLETON, morao bi u svakoj klasi i metodi imati parametar za atribut ove klase.
SINGLETON	PackageRepository	Razlog isti kao i u prvom redu tablice, samo za spremanje paketa.
SINGLETON	PackageTypeRepository	Razlog isti kao i u prvom redu tablice, samo za spremanje vrsta paketa.
SINGLETON	VehicleRepository	Razlog isti kao i u prvom redu tablice, samo za spremanje vozila.
FACTORY METHOD	<p>DataExtractor, PackageDataExtractor, PackageTypeDataExtractor, VehicleDataExtractor</p> <p>Objekti koje se kreiraju: DataReader (Abstract, ne kreira se) PackageDataReader PackageTypeDataReader VehicleDataReader</p>	Svaki dokument koji čitamo je tipa „csv“, što znači da je proces čitanja isti, ali su uvjeti provjere valjanosti i učitavanje u memoriju malo drugačiji. U ovu svrhu koristio sam factory method da učitam i spremim podatke koje pročitam svaki konkretan DataReader i usput ulovim neke iznimke koje bi iskočile u slučaju greške s datotekom.
BUILDER	<p>PackageBuilder, PackageBuilderImpl, PackageBuilderDirector</p> <p>Klasa koja se „builda“: Paket</p>	Klasa Paket ima previše atributa i imao sam dosta uvjeta kreacije što je u kodu rezultiralo sa dosta viška linija, ogromnim konstruktorom i kršenjem Single responsibility principa. Osim toga odabrao sam kreaciju u builder uzorku dizajna jer sam trebao dobivati pakete s različitim atributima (npr. vrsta paketa).

Dio B.2. Dokumentacija rješenja 2. zadatka

Naziv uzorka dizajna	Klase koje sudjeluju u uzorku dizajna	Status ¹	Opis razloga odabira uzorka dizajna
BUILDER	PackageBuilder, PackageBuilderImpl, PackageBuilderDirector Klasa koja se „builda“: Paket	S	Klasa Paket ima previše atributa i imao sam dosta uvjeta kreacije što je u kodu rezultiralo sa dosta viška linija, ogromnim konstruktorom i kršenjem Single responsibility principa. Osim toga odabrao sam kreaciju u builder uzorku dizajna jer sam trebao dobivati pakete s različitim atributima (npr. vrsta paketa).
SINGLETON	TerminalCommandHandler	S	Podaci u ovoj klasi trebaju biti dostupni skoro svakoj klasi u cijeloj zadaći i nema potrebe da se više puta instanciraju. Da nisam koristio SINGLETON, morao bi u svakoj klasi i metodi imati parametar za attribute ove klase.
SINGLETON	RepositoryManager	P	Razlog isti kao i prošli SINGLETON samo za repozitorije svakog bitnijeg objekta koji se koristi u sustavu.
FACTORY METHOD	DataSaver PackageDataSaver PackageTypeDataSaver PersonDataSaver AreaDataSaver PlaceDataSaver StreetDataSaver VehicleDataSaver Objekti koji se kreiraju su liste: Package PackageType Person Area -> Location Place -> Location Street -> Location Vehicle	P	Svaki dokument koji čitamo je tipa „csv“, što znači da je proces čitanja isti, ali su uvjeti provjere valjanosti i učitavanje u memoriju malo drugačiji. U ovu svrhu koristio sam Factory method da učitam i spremim podatke koje pročitam svaki konkretan DataSaver i usput ulovim neke iznimke koje bi iskočile u slučaju greške s datotekom. Ovakvi podaci se tada spremaju u odgovarajući repozitorij
FACADE	DataFacade DataFacadeImpl Klase podsustava: ParameterLoader ParameterHandler RepositoryManager Svaka DataSaver Klasa Klijent klasa zove Facade	N	Klasa Klijent imala je previše ovisnosti kod inicijalizacije i spremanja podataka. Ova inicijalizacija također se mora provesti u pravom redoslijedu pa sam zbog ovakvih uvjeta logiku i ovisnosti o više klasa odlučio odvojiti u FACADE. Trebao sam kreirati jednostavno sučelje koje koristi složeni podsustav.
OBSERVER	Observer Person Subject Paket	N	Trebao je sustav koji šalje obavijesti korisnicima čijima su paketi primljeni, dostavljeni itd. Zbog načina rada „preplate“ odabran je OBSERVER.
STATE	VehicleContext Vehicle VehicleState	N	Vozila ovisno o stanju u kojem su mogu raditi različite stvari. Da bi se spriječila upotreba beskonačnih if-ova i pojednostavila izmjena stanja vozila odlučio sam koristiti STATE uzorak.

¹ N – dodan u 2. zadaći, P – promijenjen u 2. zadaći, S – bez promjena u 2. zadaći

	ActiveVehicleState BrokenVehicleState InactiveVehicleState		
VISITOR	DataDisplayVisitor DrivesDataDisplayVisitor VehicleDataDisplayVisitor VehicleDisplay Vehicle	N	Visitor je korišten jer postoji više različitih ispisa istog objekta – Vehicle. Da sve ovakve ispise stavljam u klasu Vehicle "zagadio" bi tu klasu s ispisima i različitim metodama za provjere i ispise.
COMPOSITE	Location Area Place Street	N	Area može biti "Composite" koji u sebi još sadržava ulice ili gradove koji jednako tako mogu sadržavati ulice. Ovo znači da imamo stablastu strukturu i zato sam odlučio koristiti Composite. Area, Place, Street su svi Location od kojih Area i Place imaju svoje Liste Locationa kako bi se lakše došlo do ulice/grada.
PROXY	PackageSubscriber PackageSubscriberProxy PackageSubscriberImpl	N	Htio sam kontrolirati pretplatu na sustav obavijesti samo korisnicima koji imaju ulogu admin. Za ovo koristio sam PackageSubscriberProxy koji nakon takve provjere zove stvarni PackageSubscriberImpl koji tada obavlja logiku pretplaćivanja korisnika.

Dio C.1. Opis promjena u odnosu na prethodnu zadaću

Imam promjene u FACTORY methodu i SINGLETONU. Broj singletona je skraćen na način da nije svaka **Repository** klasa singleton nego postoji jedan singleton koji sadržava objekte za svaku **Repository** klasu. Ovaj novi singleton naziva se **RepositoryManager**.

FACTORY METHOD se mijenjao na način da više ne postoje apstrakcije DataReader i DataExtractor nego postoji Apstrakcija **DataSaver**. Svaki konkretni DataSaver koji nasljeđuje ovu apstrakciju mora implementirati **createDataList()** metodu koja ovisno o generičkom tipu korištenom vraća listu potrebnih proizvoda. Ova lista se tada spremi u repozitorij koji smo odredili također generičkim tipovima. Proizvodi bi u ovom slučaju bile liste objekata, tj. Liste našeg modela: Person, Paket, Vehicle, Location

Dio C.2. Opis funkcionalnosti za uzorak dizajna Proxy

Naredbe koje su uvedene služe za pretplaćivanje korisnika u sustav obavijesti. Za PROXY uzorak dizajna uvedene su tri nove naredbe:

ADMIN Korisnik – Korisnik postaje admin

PROXY Korisnik – Korisnik je pretplaćen na svaki paket

UNPROXY Korisnik – Korisniku se oduzme pretplata na svaki paket.

Primjer:

ADMIN Filip Orao – Korisnik Filip Orao postaje Admin.

PROXY Filip Orao – Korisnik Filip Orao postaje pretplaćen na svaki paket.

UNPROXY Filip Orao – Korisniku se oduzme pretplata na svaki paket.

Proxy provjerava stanje korisnika prije nego što se dozvoli pretplata korisnika kao "Observera" na svaki paket. Ako je korisnik ADMIN onda se naredbom PROXY pretplati na svaki paket u sustavu. Ako korisnik nije admin, javlja se greška za svaki paket u sustavu. Jednako tako naredba UNPROXY oduzima pretplatu korisnika na svaki paket u sustavu s istim javljanjem greške.

Sada u sustavu možemo vidjeti da korisnik dobiva/ne dobiva obavijesti za svaki paket.

```
ADMIN Filip Orao
Dec 09, 2023 10:16:04 AM org.foi.uzdiz.mmusica.Klijent askForUserInput
INFO: Naredba uspjesno izvršena!
PROXY Filip Orao
Dec 09, 2023 10:16:11 AM org.foi.uzdiz.mmusica.Klijent askForUserInput
INFO: Naredba uspjesno izvršena!

Paket CROVŽ0001 -> PREUZET
Mato Medved: Paket --> CROVŽ0001: Preuzet
Pero Kos: Paket --> CROVŽ0001: Preuzet
Filip Orao: Paket --> CROVŽ0001: Preuzet
Paket CROVŽ0002 -> PREUZET
Mato Miš: Paket --> CROVŽ0002: Preuzet
Filip Orao: Paket --> CROVŽ0002: Preuzet
Paket CROVŽ0003 -> PREUZET
Silva Sova: Paket --> CROVŽ0003: Preuzet
Pero Paun: Paket --> CROVŽ0003: Preuzet
Filip Orao: Paket --> CROVŽ0003: Preuzet
Paket CROVŽ0004 -> PREUZET
Filip Orao: Paket --> CROVŽ0004: Preuzet
Paket CROVŽ0005 -> PREUZET
Kiki Kovač: Paket --> CROVŽ0005: Preuzet
Filip Orao: Paket --> CROVŽ0005: Preuzet
Paket CROVŽ0006 -> PREUZET
Miki Mrav: Paket --> CROVŽ0006: Preuzet
Filip Orao: Paket --> CROVŽ0006: Preuzet
Paket CROVŽ0007 -> PREUZET
Miki Mrav: Paket --> CROVŽ0007: Preuzet
Filip Orao: Paket --> CROVŽ0007: Preuzet
Paket CROVŽ0008 -> PREUZET
Miki Mrav: Paket --> CROVŽ0008: Preuzet
Filip Orao: Paket --> CROVŽ0008: Preuzet
Paket CROVŽ0009 -> PREUZET
Miki Mrav: Paket --> CROVŽ0009: Preuzet
Filip Orao: Paket --> CROVŽ0009: Preuzet
Paket CROVŽ0010 -> PREUZET
Miki Mrav: Paket --> CROVŽ0010: Preuzet
Filip Orao: Paket --> CROVŽ0010: Preuzet
```

Paketima kojima već pripada ne pridružuje se više puta. Za ostale pakete u kojima nije pošiljatelj ili primatelj sada prima obavijesti. Isti način rada vrijedi i za naredbu UNPROXY

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

