

Obrazac za zadaću na predmetu "**Uzorci dizajna**" ak. god. 2022./2023.

Ime i prezime studenta/ice: Alan Vranić

Matični broj: 0016130344

Dio A. Osnovni podaci o zadaći

R.br.	Pitanje	Odgovor	
1.	Grupa na seminaru:	G2	
2.	Broj i naziv zadaće:	2	Nadograđena brodska luka
3.	Procjena vremena za realizaciju bez decimala):	30 sati	
4.	Procjena % završenosti (bez decimala):	100 / 100%	
5.	Procjena bodova za izradu zadaće (1 decimala):	13 / (DZ2 - 13)	
6.	Želim prezentirati zadaću:	Ne (već sam prezentirao prvu)	
7.	Koji dijelovi iz opisa zadaće nisu realizirani:	Svi dijelovi iz opisa su realizirani.	
8.	Postoji li dio zadaće koji vrijedi posebno istaknuti i zašto:	Vjerujem da je zadaća napisana sa izuzetno čistom arhitekturom, te primjenjuje potrebne uzorke dizajna na pravim mjestima na kreativan način. Dnevnik brodske luke se može ispisati koristeći naredbu „D“.	
9.	Postoje li dijelovi zadaće koji imaju pogrešku u radu i koje:	Ne	
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. zadataće (kopirano i nepromijenjeno)

Naziv uzorka dizajna	Klase koje sudjeluju u uzorku dizajna	Opis razloga odabira uzorka dizajna
Singleton	BrodskoLuka, VirtualniSat	Brodsko luka ponaša se kao baza podataka sa svim učitanim brodovima, vezovima itd. Na sličan način, VirtualniSat pohranjuje trenutno vrijeme. Svi podaci, kao i virtualno vrijeme moraju biti dostupni na više mjesta kroz aplikaciju, stoga je ovo savršena primjena za Singleton ozorak dizajna. Također, potrebno je da postoji samo jedna brodska luka i samo jedan virtualni sat, stoga ovaj uzorak rješava i taj problem.
Proxy	IServis, Servis, ServisProxy	Proxy služi za više stvari, no u ovoj zadaći je primarni cilj iskoristiti proxy kao posrednika koji prije izvršavanja svake naredbe ispisuje virtualno vrijeme. Ponekad nakon neke naredbe ispiše i dodatne potvrdne podatke. U budućim zadaćama, proxy bi se lako mogao proširiti da bilježi ostale greške, vrši autentifikaciju ili validira ulazne podatke prije nego one dođu do naredbe koja se želi izvršiti. Također, proxy započinje rad virtualnog sata čim je instanciran.
Factory method	IParametriZauzetosti, IZauzetost, ParametriRasporedneZauzetosti, ParametriRezervacijskeZauzetosti, RasporednaZauzetost, RasporednaZauzetostKreator, RezervacijskaZauzetost, RezervacijskaZauzetostKreator, ZauzetostKreator	U 1. zadaći to je najveći uzorak dizajna i po meni najkorisniji. U zadaći se zauzetost može predložiti na dva načina – putem periodičnog rasporeda i putem zahtjeva rezervacije koji su nepredvidljivi. Ipak, oboje predstavljaju isti koncept – zauzeti period vremena na pojedinom vezu za pojedini brod. Odlučio sam koristiti Factory method kako bi olakšao instanciranje tih dviju klasa (putem zasebnih kreatora), te dodatno napravio sučelje IZauzetost koje ima definiranu metodu ZauzetURasponu. Na taj način, raspored i zahtjev rezervacije definira na svoj način što i kada znači biti zauzet u nekom rasponu datuma, a korištenje tih klasa je iznimno jednostavno te je uvelike smanjilo količinu koda koju sam morao pisati, što se uvelike reflektiralo na kvalitetu zadataće. Također je uveden interface IParametriZauzetosti kako bi mogao instancirati klase IZauzetosti koje imaju različite argumente za konstruktor, a svejedno se pridržavaju svih pravila strukture ovog uzorka.

Dio B.2. Dokumentacija rješenja 2. zadatke

Naziv uzorka dizajna	Klase koje sudjeluju u uzorku dizajna	Status ¹	Opis razloga odabira uzorka dizajna
Singleton	BrodskoLuka	S	Navedeno u tablici iznad (nepromijenjeno)
Singleton	VirtualniSat	S	Navedeno u tablici iznad (nepromijenjeno)
Singleton	IspisivačTablice	N	Uvijek postoji samo jedan ispisivač tablice jer se u njemu prilikom učitavanja spremaju opcije prikaza tablice. Također, te opcije se mogu promijeniti tijekom izvršavanja programa. Na ovaj način, sve vezano uz ispis tablica i njihovo formatiranje je na jednom mjestu dostupnom kroz cijelu aplikaciju.
Proxy	Navedeno u tablici iznad (nepromijenjeno)	S	Navedeno u tablici iznad (nepromijenjeno)
Factory method	Navedeno u tablici iznad (nepromijenjeno)	S	Navedeno u tablici iznad (nepromijenjeno)
Template method	BrodCsvCitac, BrodskoLukaCsvCitac, CsvCitacAbstract, KanalCsvCitac, MolCsvCitac, MolVezCsvCitac, RasporedCsvCitac, VezCsvCitac, ZahtjevRezervacijeCsvCitac	N	<p>Primjena Template method uzorka je idealna u slučajevima kad imamo algoritam u kojemu je potrebno izmijeniti ili imati varijacije pojedinih koraka.</p> <p>U ovom slučaju, proces učitavanja CSV datoteka je 90% sličan. Jedini korak koji varira je korak parsiranja podataka, koju konkretne implementacije CSV čitača mogu naslijediti i nadjačati.</p> <p>Na ovaj način, dolazi do bolje iskorištenosti koda (DRY), i svaki čitač je zaslužan samo za svoj dio.</p>
Chain of Command	HitniPrijevoz, IHitniPrijevozHandler	N	Funkcionalnost, način implementacije i razlog odabira uzorka Chain of Command dostupan je ispod, u dijelu Dio C.2.
Visitor	IVezVisitor, IZauzetiVez, VezZauzetostPremaVrstamaVisitor	N	<p>Uzorak dizajna Visitor implementiran je jer je zahtjevan u zadatku. Visitor nam omogućava da odvajamo algoritme od objekata nad kojim oni operiraju.</p> <p>U našem slučaju, primjenom Visitora omogućili smo sumarno zbrajanje zauzetih vezova prema vrstama za određeno vrijeme. Visitor posjećuje objekte koji implementiraju sučelje IZauzetiVez što im dozvoljava da „prihvate“ Visitora, zbog čega on može pristupiti ključnim podacima koje koristimo za sumarno zbrajanje.</p>
Mediator	IMediator, ISubscriber, OdgovorLuckeKapetanije, PorukaVrhKanal, ZahtjevZaOdjavu, ZahtjevZaPrivezNaRezerviraniVez, ZahtjevZaPrivezNaSlobodanVez	N	<p>Mediator je najključniji uzorak dizajna u 2. zadatku. Uzorak dizajna Mediatora omogućuje nam da se Kanali ponašaju kao posrednici između svih korisnika na toj frekvenciji.</p> <p>Na ovaj način, brodovi i brodska luka mogu si međusobno slati poruke, bez da</p>

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

			<p>direktno znaju jedan za drugog. Brodska luka uvijek sluša na svim frekvencijama. Cilj je da pojedini brod i brodska luka mogu komunicirati, no svi ostali posrednici će također dobiti te poruke.</p> <p>Bitno je napomenuti da u ovom slučaju Mediator ima i ulogu uzorka Observer. Odnosno, nakon što brod pošalje poruku Mediatoru (kanalu), on će obavijestiti sve svoje pretplatnike da je stigla poruka, te istu proslijediti. Svi brodovi i brodska luka odlučuju što će raditi sa dobivenim porukama, što stvara veliku modularnost.</p>
--	--	--	---

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

Svi dodani uzorci su novi, odnosno, nema promjene starih uzoraka.

Dio C.2. Opis funkcionalnosti za uzorak dizajna Chain of Responsibility

Hitni Prijevoz

Za implementaciju funkcionalnosti uzorka Chain of Responsibility osmislio sam slučaj u kojem je potrebno organizirati hitni prijevoz putnika. U bilo kojem momentu, u brodsku luku mogu pristići putnici koji zahtijevaju hitni prijevoz bilo kojim od trenutno privezanih brodova.

Hitni putnik može biti službenik ili imigrant.

Brod mora proći više kvalifikacija kako bi bio sposoban prevoziti takve putnike. Prvo, dolaze u obzir samo privezani brodovi u brodskoj luci. Zatim, brod mora imati slobodan kapacitet posvećen samo hitnim putnicima koji se može s vremenom popuniti. Posljednje, brod ovisno o vrsti odlučuje želi li prevoziti službenike, imigrante ili oboje.

Sintaksa naredbe izgleda ovako:

HP [broj sluzbenika] [broj imigranata]

Primjeri:

- HP 100 0
- HP 25 50
- HP 75 1000

Brodsko luka prolazi kroz sve privezane brodove i svakome dodjeljuje svog nasljednika (handlera). Kroz lanac odgovornosti šalje se objekt (HitniPrijevoz) koji vodi računa o tome koliko još hitnih putnika nema mjesta.

Ova implementacija je savršena za uzorak Chain of Responsibility zbog navedenih 5 razloga:

- Svaki brod (kada na njega dođe red) mora odgovoriti na pitanje želi li on prevesti putnike određene vrste ili ne (**želimo li procesirati zahtjev**)
- Svaki brod dinamično vodi računa o broju hitnih putnika koje prenosi i modificira kontekstualni objekt HitniPrijevoz (**dinamični podaci**)
- Ukoliko se popuni kapacitet broda ili jednostavno ne želi pojedinu vrstu putnika, on zove svog broda nasljednika (handler) koji će ispitati putnike, i tako lanac ide do kraja (**prosljeđivanje zahtjeva**)
- Može se desiti da već prvi brod zbrine sve putnike, pa se lanac zaustavlja (**rano zaustavljanje lanca**)
- Može se desiti da se dođe na kraj lanca i niti jedan putnik (ili samo dio njih) bude zbrinut (**dolazak do kraja lanca / nemogućnost izvršavanja zahtjeva**)

Nakon izvršavanja lanca, aplikacija ispisuje zaključak izvršenja, odnosno, jesu li svi putnici uspješno bili zbrinuti, ili ispisuje broj i vrstu onih koji nisu.

Nakon što se brod odveže sa luke, smatra se da se vratio na svoje odredište i njegov trenutni broj hitnih putnika resetira se na nulu.

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

