

Obrazac za zadaću na predmetu "Uzorci dizajna"

Ime i prezime studenta/ice: Josip Petanjek

Matični broj: 0016124756

Dio A. Osnovni podaci o zadaći

R.br.	Pitanje	Odgovor	
1.	Grupa na seminaru:	G3	
2.	Broj i naziv zadaće:	3	Iznajmljivanje električnih vozila
3.	Procjena vremena za realizaciju bez decimala):	36 sati	
4.	Procjena % završenosti (bez decimala):	90 / 100%	
5.	Procjena bodova za izradu zadaće (1 decimala):	13 / 15	
6.	Žalim prezentirati zadaću:	NE	
7.	Koji dijelovi iz opisa zadaće nisu realizirani:	Nisam siguran da li su svi tablični ispisi napravljeni po željenoj strukturi, na primjer struktura računa.	
8.	Postoji li dio zadaće koji vrijedi posebno istaknuti i zašto:	Ne.	
9.	Postoje li dijelovi zadaće koji imaju pogrešku u radu i koje:	Ispisivanje strukture najnižih grana je možda neispravno. Postoji mogućnost da se program izbaci iz funkcije zbog nepoznatih razloga.	
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, sa kolegama nisam dijelio rješenja niti koristio njihova.	

Dio B. Dokumentacija rješenja zadaće

Naziv uzorka dizajna	Klase koje sudjeluju u uzorku dizajna	Opis razloga odabira uzorka dizajna
Singleton	NacinRada; Tvrtka; VirtualnoVrijeme; SustavRacuna	„Tvrtka“ se koristi za spremanje svih učitanih podataka, „NacinRada“ se koristi za određivanje načina rada, „VirtualnoVrijeme“ se koristi za postavljanje i čitanje virtualnog vremena, „SustavRacuna“ za praćenje računa tvrtke
Factory Pattern	AktivnostDatotekaProduct; CjenikDatotekaProduct; KapacitetiDatotekaProduct; LokacijeDatotekaProduct; OsobaDatotekaProduct; TvrtkaDatotekaProduct; VozilaDatotekaProduct; ConcreteCreator; DatotekaCreator; DatotekaProduct	Jer je potrebno stvoriti instance svih klasa iz datoteka, ali na različite načine. Pomoću „Product“ definiramo sučelje odnosno funkciju koja nam vraća listu svih instanci objekata, sa „ConcreteProduct“ klasama implementiramo to sučelje te definiramo kako će se objekti čitati iz datoteka. Sa „Creator“ sučeljem odnosno metodu koja vraća objekt tipa „Product“ te ovo isto sučelje implementiramo u „ConcreteCreator“ tako da se vraća pravilan objekt.
Bridge	IzvoditeljImplementatorAbstraction; IzvoditeljAktivnosti; IAktivnost; KrajPrograma; NajamVozila; PregledRaspolozivihMjesta; PregledRaspolozivihVozila; Prijelaz; RacunIspis; Stanje; VracanjeVozila; ZaradaNajam; PlacanjeDugovanjaKorisnika; FinancijskoStanje; RacuniKorisnika;	Jer sve aktivnosti imaju zajedničko sučelje („IAktivnost“), a ovaj uzorak nam omogućuju da izvedemo bilo koju klasu koja implementira to sučelje. Dakle, sučelje sa funkcijom „izvedi()“, koju aktivnosti implementiraju, „IzvoditeljImplementatorAbstraction“ nam definira funkciju koja prima objekt koji implementira sučelje „IAktivnost“, te funkciju za izvođenje same aktivnosti, a „IzvoditeljAktivnosti“ ju proširuje te implementira te pokreće aktivnosti.
Iterator	Iterator; IteratorContainer; CompositeIterator	Za slijedni pristup elementima agregiranog objekta, da se možemo kretati kroz Composite.
Facade	Pronalazitelj; FacadeSustavPronalaska	Za pronalaženje veza između računa i lokacija kroz uniformno sučelje koje oni implementiraju.
Builder	ConcreteBuilder; TvrtkaBuilder; TvrtkaDirector	Za stvaranje tvrtke, odnosno njezinog stabla, jer je to složeni objekt pa moramo odvajati njegovu konstrukciju.
Composite	OrganizacijskaJedinica; OrganizacijskaJedinicaComposite	Sastavljanje organizacijskih jedinica u stablo, te njihova obrada kroz implementaciju dekoratora i agregiranje kroz iterator.
Decorator	CompositeDecorator; CompositeInterface; CompositeConcreteDecorator	Proširuje composite bez da mijenja njegovo sučelje, koristi se za ispis njegovih atributa, kao što neke aktivnosti traže.
State	State; Slobodno; Unajmljeno; Neispravno; NaPunjenju	Za praćenje stanja vozila, dakle klasa Vozilo koristi State objekt.
Chain of command	RacunHandler; RacunEvidencija; RacunLokacije; RacunOsobe; RacunRequest; RacunResponse;	Za izvršavanje evidencije računa te pronalazak istih. Poziva se preko Facade sustava pronalaska.
Model View Controller	Controller; InteraktivniPogled; IzlazPogled; SkupniPogled; Pogled;	Koristi se za uspostavljanje troslojne strukture, pogledi nasljeđuju sučelje Pogled te primjereno preko Controller ispisuju u zadani pogled.

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

Novi uzorci:

- State - Za praćenje stanja vozila, dakle klasa Vozilo koristi State objekt.
- Chain of command - Za izvršavanje evidencije računa te pronalazak istih.
- Model View Controller - Koristi se za uspostavljanje troslojne strukture.

Nadogradnje:

- Composite – ispravljen ispis kumulativnih podataka broja sati i broja unajmljenih vozila. Ispravljen ispis najniže organizacijske jedinice.
- Decorator – nadodani dekoratori za osobu i račun.
- Bridge – nadodane nove aktivnosti.

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

