

Obrazac za zadaću na predmetu "Uzorci dizajna"

Ime i prezime studenta/ice: _____ Luka Pintarić _____

Matični broj: _____ 44892-R _____

Dio A. Osnovni podaci o zadaći

R.br.	Pitanje	Odgovor	
1.	Grupa na seminaru:	G2	
2.	Broj i naziv zadaće:	3	Nogometno prvenstvo – nastavak 2
3.	Procjena vremena za realizaciju bez decimala):	___30___ sati	
4.	Procjena % završenosti (bez decimala):	___90___ / 100%	
5.	Procjena bodova za izradu zadaće (1 decimala):	___13___ / (DZ3 - 15)	
6.	Žalim prezentirati zadaću:	NEMA PREZENTACIJE	
7.	Koji dijelovi iz opisa zadaće nisu realizirani:	Svi novododani zahtjevi su realizirani. Možda postoje neke provjere kod učitavanja koje nisu realizirane.	
8.	Postoji li dio zadaće koji vrijedi posebno istaknuti i zašto:	Kod obrade događaja koja je implementirana pomoću ChainOfResponsibility nije specificirano da li je potrebno tu obradu vršiti kod učitavanja ili kod pripreme podatak za ispis. U mojem rješenju implementirano je kod učitavnja, jer mi je to imalo više smisla.	
9.	Postoje li dijelovi zadaće koji imaju pogrešku u radu i koje:	Nisam primijetio tijekom testiranja	
10.	Da li ste koristili tuđi programski kod u realizaciji zadaće izvan spomenutih izvora na nastavi:	Nisam.	
11.	Da li ste koristili programska rješenja ili dijelove programskog koda od drugih kolega:	Nisam.	

Dio B. Dokumentacija rješenja zadaće

Naziv uzorka dizajna	Klase koje sudjeluju u uzorku dizajna	Status ¹	Opis razloga odabira uzorka dizajna
Singleton	NogometnaLigaPodaci	P	NogometnaLigaPodaci predstavlja klasu koja je spremište podataka koji će biti korišteni u radu aplikacije. Klasa Program (predstavlja klijenta) joj prosljeđuje odgovarajuće nazive datoteka s podacima, a NogometnaLigaPodaci čita podatke iz datoteka te pomoću factoryja UcitavanjeCreator sprema podatke u liste. Ova klasa jest singleton jer je potrebna samo jedna njezina instanca koja se dobiva preko statične metode koja kada je prvi put pozvana kreira instancu, a nadalje ju samo vraća.
Factory method	UcitavanjeCreator, UcitavanjeCreatorDogadaj, UcitavanjeCreatorIgrac, UcitavanjeCreatorKlub, UcitavanjeCreatorSastavUtakmice, UcitavanjeCreatorUtakmica	S	Factory method jest UcitavanjeCreator i njegove konkretizacije. Ovaj factory method iz danog polja atributa kreira objekte koji će biti spremljeni u listama singletona. Dakle, služi za učitavanje podataka. Uzorak se koristi jer postoji više datoteka istog formata, no njihovi podaci se spremaju u različite tipove objekata.
Builder	DogadajBuilder	S	DogadajBuilder služi kako bi se kreirao objekt Dogadaj prilikom učitavanja podataka u singleton. Dogadaj jest objekt koji prilikom svog kreiranja može zahtijevati od 3 do 6 atributa u više varijanta. Kako bismo smanjili broj konstruktora koje bi Dogadaj trebao sadržavati koristimo buildera. On će postaviti vrijednosti obaveznih atributa i putem svojih metoda će omogućiti postavljanje vrijednosti izbornih atributa. Koristi ga konkretni kreator UcitavanjeCreatorDogadaj.
Facade	UcitavanjeFacade	S	Uzorak facade je odabran kako bi se pružilo sučelje za podsustav učitavanja podataka (u njemu se nalazi factory method UcitavanjeKreator) iz datoteka. Na taj način dobivena je slojevitost sustava, tj. odvojili smo podsustav učitavanja od klijenta što je vrlo korisno ako sustav u budućnosti postane složen.
Composite	KluboviComponent Klub Osoba Trener Igrac PrvenstvoComponent Prvenstvo Kolo Utakmica	S	Projekt sadrži dva Composite uzorka. Composite je odabran kako bi se prikazala hijerarhija među objektima (npr. Klub-Igrac, gdje jedan klub može sadržavati više igrača). Također sve komponente nasljeđuju istu nadklasnu pa se stoga njihovi objekti mogu tretirati uniformno.

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

	SastavUtakmice Dogadaj		
Visitor	PrikazVisitor PrikazVisitorBodovi PrikazVisitorDogadaj PrikazVisitorKarton PrikazVisitorRezultat PrikazVisitorStrijelac PrikazVisitorSastavi	P	Primjenom uzorka Visitor strukturi koja sadrži zapis podataka (to su u ovom slučaju composite hijerarhija i njezini objekti) dodajemo nove operacije bez da mijenjamo klase te strukture. Svaki konkretni visitor obrađuje određenu vrstu statistike za prikaz i ako je potrebno u budućnosti dodati neku novu vrstu statistike tada nije potrebno mijenjati entitetske klase nego samo dodajemo novi konkretni visitor.
Observer	DogadajiSubject DogadajObserver DogadajObserverSemafor	S	Observer kao uzorak određuje ovisnost više objekata o jednom koji kada mijenja stanje, svi koji ovise o njemu su obaviješteni o toj promjeni te sukladno tome izvršavaju određenu operaciju. U našem slučaju ovaj uzorak se koristi za prikazivanje i promjenu stanja semafora. Kada klijent promjeni stanje subjecta on obavještava observee te oni ažuriraju semafor. Trenutno postoji samo jedan observer u sustavu, ali u budućnosti ako će biti potrebe da ih bude više tada smo u velikoj prednosti.
State	Igrac State IgraState ZamjenaState ZutiKartonState IzvanIgreState	N	Uzorak state dodan je kako bi klasi/entitetu Igrac dodali stanja u kojima se on u nekom trenutku može nalaziti. Na taj način ćemo nekom objektu Igrač dinamički promijeniti ponašanje a izgledat će kao da objekt mijenja klasu.
Chain of responsibility	DogadajHandler DogadajHandlerAutogol DogadajHandlerCrveni DogadajHandlerGol DogadajHandlerGolPenal DogadajHandlerKraj DogadajHandlerPocetak DogadajHandlerZamjena DogadajHandlerZuti DogadajVerifikator	N	Chain of responsibility uzorak dodan u svrhu obrade događaja. Na taj način objekt koji zahtjeva obradu događaja ne mora znati sve primatelje koji se nalaze u lancu. Potrebno je samo dohvatiti referencu na prvi objekt u lancu, a zahtjev se tada propagira automatski i stiže do primatelja koji zahtjev može obraditi. Verifikacija određenog događaja odvija se upotrebom prethodno navedenog uzorka state.
Singleton	DogadajVerifikator GeneratorRasporeda	N	DogadajVerifikator predstavlja objekt za koji je potrebna samo jedna instanca tijekom izvođenja. Ovaj objekt zadužen je za kreiranje instance lanca odgovornosti DogadajHandler. Dohvaćanjem instance DogadajVerifikator drugi objekti po potrebi mogu dohvatiti referencu na lanac DogadajHandler. Kako bi klijentska strana dohvatila algoritam za generiranje rasporeda utakmica, u tu svrhu je kreirana klasa GeneratorRasporeda i potrebna je samo jedna instanca ovog objekta pa je odabran uzorak Singleton.

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

Dodana je klasa GeneratorAlgoritama i u klasi Program implementirane su metode za generiranjem, pregled, postavljanje i ispis rasporeda.

Dodan je novi visitor za prikaz sastava prije i poslije utakmice PrikazVisitorSastavi. U tu svrhu kreirana je nova pomoćna klasa za prikaz sastava IgracPostava.

U singletonu NogometnaLigaPodaci dodane su neke nove metode i novi atributi za spremanje generiranih rasporeda.

Klasa Igrac koja je dio kompozita malo je izmjenjena jer ju je bilo potrebno povezati sa State objektima.

Dodane su još neke dodatne provjere kod učitavanja podataka. Dodatno je ovaj dio poboljšán jer sada provjera događaja kod učitavanja se odvija kroz DogadajHandler.

Dio C.2. Opis funkcionalnosti za uzorak dizajna Decorator

Nije implementirano

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

