

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

Ime i prezime studenta/ice: Nikola Muše

Matični broj: 0016131266 (48735)

**Dio A. Osnovni podaci o zadaći**

R.br.	Pitanje	Odgovor	
1.	Grupa na seminaru:	G2	
2.	Broj i naziv zadaće:	3	Zadaća 3. Nogometno prvenstvo – nastavak 2
3.	Procjena vremena za realizaciju bez decimala):	40 sati	
4.	Procjena % završenosti (bez decimala):	100 / 100%	
5.	Procjena bodova za izradu zadaće ( 1 decimala):	15 / (DZ3 - 15)	
6.	Želim prezentirati zadaću:	NEMA PREZENTACIJE	
7.	Koji dijelovi iz opisa zadaće nisu realizirani:	Sve je realizirano.	
8.	Postoji li dio zadaće koji vrijedi posebno istaknuti i zašto:	NE	
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. Dokumentacija rješenja zadaće

Naziv uzorka dizajna	Klase koje sudjeluju u uzorku dizajna	Status <sup>1</sup>	Opis razloga odabira uzorka dizajna
Singleton	Prvenstvo.cs	P	Kroz cijelo trajanje izvođenja programa svim klasama treba biti dostupna samo jedna instanca klase. Želimo osigurati da uvijek svim klasama bude dostupna jedinstvena instanca.
Factory Method	U prvom Factory Methodu: -IProduct.cs -Creator.cs -ConcreteCreator.cs -UcitajUtakmice.cs -UcitajSastave.cs -UcitajKlubove.cs -UcitajIgrace.cs -UcitajDogadaje.cs U drugom Factory Methodu: -IProductIspis.cs -CreatorIspis.cs -ConcreteCreatorIspis.cs -IspisStrijelaca.cs -IspisRezultataUtakmica.cs -IspisKlubova.cs -IspisKartona.cs	P	Koristim ga prilikom kreiranja. Koristim ga za kreiranje objekata bez specificiranja njihovih konkretnih klasa. U prvom slučaju ga koristim kod učitavanja datoteka. Prvo provjeravam oznaku datoteke i onda ovisno o njoj određujem vrstu učitavanja. Factory method pruža sučelje za kreiranje objekata, a podklasama mijenjam način na koji će objekti biti stvoreni. To radimo jer ne može klasa predvidjeti klasu za objekte koje mora kreirati. U drugom primjeru ga koristim za ispis ljestvica ovisno o tome koju tablicu korisnik želi ispisati.
Builder	DogadajBuilder.cs OdredivanjeDogadaja.cs IDogadajBuilder.cs	S	Koristim ga kod kreiranja događaja zbog toga što je kreiranje događaja složeniji proces. U datoteci popisa događaja svi atributi nisu obavezni i onda ovisno o broju atributa događaj se „build-a“ drugačije. Ovisno o vrsti događaja nekada su obavezna tri atributa, nekad 5, a nekad 6.
Composite	U prvom Compositu: -Clanovi.cs -Klub.cs -Osoba.cs -Trener.cs -Igrac.cs U drugom Compositu: -INogometnoPrvenstvo.cs -NogometnoPrvenstvo.cs -Kolo.cs -Utakmica.cs -Sastav.cs -Dogadaj.cs	P	Prvi Composite koristim za kreiranje strukture objekata koji se sastoji od klubova i osoba (trener i igrači). Drugi Composite koristim za kreiranje nešto složenije strukture objekata koje čine prvenstvo, kolo, utakmica, sastav i događaj.
Facade	FacadeUcitaj.cs	S	Facade koristim za učitavanje datoteka. Facade prilikom učitavanja datoteka koristi Factory Method.
Visitor	IVisitorObradaAktivnosti.cs VisitorObrada.cs IProductIspis.cs IspisStrijelaca.cs IspisRezultataUtakmica.cs IspisKlubova.cs IspisKartona.cs	S	Koristim ga za obradu podataka za ispis strijelaca, bodova klubova, kartona i rezultata (aktivnosti). Ispis se radi u Factory Method-u za ispis tablica.
Observer	ISubjectIspis.cs IObserverIspis.cs	S	Koristio sam Observer u svrhu ispisa tablice događaja iz unesene aktivnosti. Tablica ispisa

<sup>1</sup> N – dodan u 2. zadaći, P – promijenjen u 2. zadaći, S – bez promjena u 2. zadaći

	SubjectDogadaj.cs ObserverDogadaj.cs		podrazumijeva one događaje neke utakmice prema unesenom kolu te klubovima koji igraju.
State	ContextSastavaIgraca.cs StateIgra.cs StateIskljucen.cs StatePricuva.cs StateSastavaUtakmice.cs StateZamijenjen.cs	N	Koristio sam State u svrhu dodavanja stanja u sastavu. Stanja mogu biti : u igri (StateIgra), u pričuvi (StatePricuva), isključen (StateIskljucen) i zamijenjen (StateZamijenjen).
Chain of responsibility	IObradaDogadaja.cs ObradaGola.cs ObradaKartona.cs ObradaZamjeneIgraca.cs OsnovniHandlerObrade.cs	N	Koristio sam Chain of Responsibility za provjeru događaja iz datoteke kako bi se izbacili oni događaji u kojima su sudjelovali igrači sa nedozvoljenim stanjima za taj događaj.

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

- U Singletonu je dodan atribut za važeći raspored kako bi se tom rasporedu moglo uvijek pristupiti.
- Prvi Factory Method je promijenjen na način da se za učitane sastave dodaju pripadna stanja, a za učitane događaje se rade dodatne provjere (Chain of responsibility i State).
- U drugom Composite uzorku je za sastav dodan atribut stanja.

## **Dio C.2. Opis funkcionalnosti za uzorak dizajna Decorator**

**Decorator nije implementiran.**

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

