



# Objektno-orijentisano projektovanje

*Elektronski fakultet Niš*

Projektni obrasci



# Istorijski pregled razvoja projektnih obrazaca

- **1979**: Christopher Alexander, arhitekta – knjiga: “The Timeless Way of Building”, Oxford Press.
- **1987**: OOPSLA (Object Oriented Programming System), Orlando, Ward Cunningham i Kent Beck prezentuju primenu Alexander-ove ideje projektnih obrazaca i na OO softver.
- **1991**: Erich Gamma, Doktorska teza.



# Istorijski pregled razvoja projektnih obrazaca

- **1991:** James Coplien, *Advanced C++ Idioms*, knjiga.
- **1994:** PLoP Konferencije i knjige, 1994-do danas.
- **1994:** E.Gamma, R.Helm, R.Johnson and J.Vlissides :  
“Design Pattern: Elements of Reusable OO software”.



# Definicija projektnih obrazaca

- Ne treba svaki problem rešavati od nule...
- Treba iskoristiti već gotova rešenja od ranije.
- Projektni obrasci olakšavaju ponovno korišćenje uspešnih modela i arhitektura.



# Definicija projektnih obrazaca

- Christopher Alexander:
  - “Svaki projektni obrazac opisuje problem koji se stalno ponavlja u našem okruženju i zatim opisuje suštinu rešenja problema tako da se to rešenje može upotrebiti milion puta, a da se ne ponovi dva puta na isti način...”
- On je govorio o obrascima za zidanje gradova, ali sve ovo važi i kod OO softvera.



# Definicija projektnih obrazaca

- GoF:
  - “Projektni obrasci kod softvera predstavljaju opise objekata i klasa koje komuniciraju i prilagođene su rešavanju opšteg problema projektovanja u određenom kontekstu...”



# Elementi projektnih obrazaca

- **Ime obrasca**
  - U nekoliko reči opisuje problem.
- **Problem**
  - Opis situacije u kojoj se obrazac koristi.
- **Rešenje**
  - Opis elemenata koji čine projekat, njihove odnose, odgovornosti i saradnju.
- **Posledice**
  - Rezultati i ocene primene obrasca.



# Opis projektnih obrazaca (GoF)

1. Ime obrasca i klasifikacija	
2. Namena	
3. Poznat takođe kao	
4. Motivacija	
5. Primenjivost	
6. Struktura	
7. Učesnici	
8. Saradnja	





## Opis projektnih obrazaca (GoF)

9. Posledice	
10. Implementacija	
11. Primer koda	
12. Poznati primeri korišćenja	
13. Povezani obrasci	



[PATTERN-NAME]

Author

[YOUR-NAME] ([YOU@YOUR.ADDR]).

Last updated on [TODAY'S-DATE]

### Context

[PARAG-1]

[PARAG-2]

### Problem

[ONE-ASPECT]

[ANOTHER-ASPECT]

### Examples

### Forces

1.[FORCE-1]

2.[FORCE-2]

### Design

[PARAG-1]

[PARAG-2]

### An Implementation

[SOME-CODE]

### Examples

### Variants

[VARIANT]

[ANOTHER-VARIANT]

### See Also

[ANOTHER-REF]

```
IF    you find yourself in CONTEXT
      for example EXAMPLES,
      with PROBLEM,
      entailing FORCES
THEN for some REASONS,
      apply DESIGN FORM AND/OR RULE
      to construct SOLUTION
      leading to NEW CONTEXT and OTHER PATTERNS
```

## Templejt projektnih obrazaca

**Više templejta na adresama:**

- <http://hillside.net/patterns/template.html>
- [http://www.paterndepot.com/pages \(Templates\)](http://www.paterndepot.com/pages (Templates))



# Katalozi projektnih obrazaca

- **Katalog projektnih obrazaca je skup povezanih obrazaca iz odgovarajućeg domena.**
- **Postoje katalozi obrazaca za:**
  - Distribuirane sisteme
  - Veštačku inteligenciju
  - Sisteme za rad u realnom vremenu
  - Telekomunikacione sisteme
  - Adaptivno programiranje
  - ...

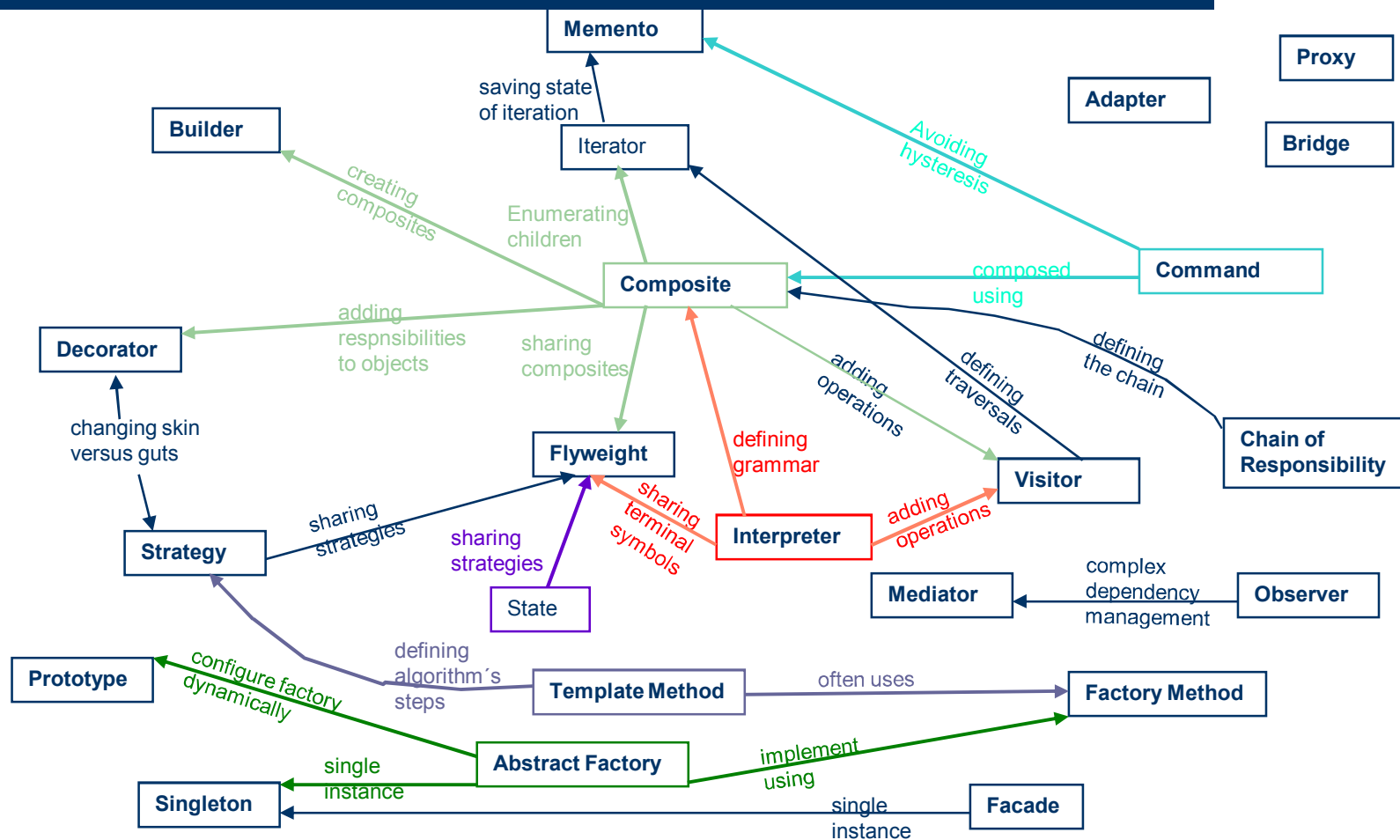


# GoF katalog

		Namena		
		Kreiranje	Struktura	Ponašanje
Domen	Klasa	Factory Method	Adapter (class)	Interpreter Template Method
	Objekat	Abstract Factory Builder Prototype Singleton	Adapter (object) Bridge Composite Decorator Facade Flyweight Proxy	Chain of Responsibility Command Iterator Mediator Memento Observer State Strategy Visitor



# Odnosi među obrascima





# Linkovi

- <http://hillside.net/patterns/>
- <http://www.netobjectives.com/design.htm>
- <http://umbc7.umbc.edu/~tarr/dp/dp.html>
- [http://www.links2go.com/topic/Design\\_Patterns](http://www.links2go.com/topic/Design_Patterns)
- [http://www.objenv.com/cetus/oo\\_patterns.html](http://www.objenv.com/cetus/oo_patterns.html)
- <http://www.patternscentral.com/>
- <http://www.tm.tue.nl/it/research/patterns>
- <http://www.mokabyte.it/1998/10/pattern.htm>
- <http://www.c2.com>
- <http://www.enteract.com/~bradapp/docs/patterns-intro.html>
- . . .

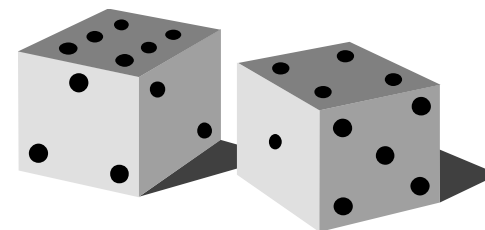


# Primeri primene projektnih obrazaca

- Igra - bacanje kockica
- Bankarski sistem

## Prvi primer

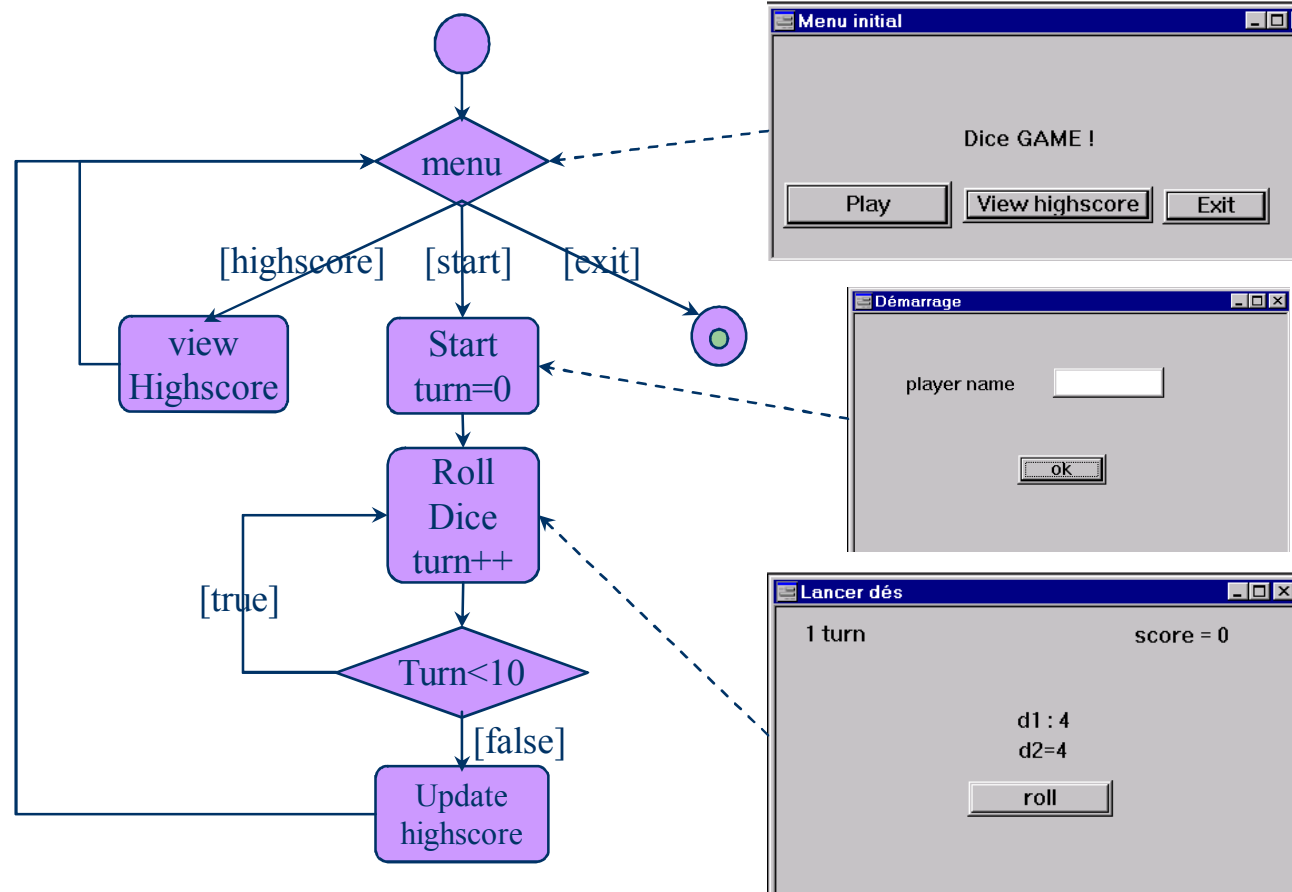
- Igra sa kockama
- Igrač baca 10 x 2 kocke
- Ako se dobije zbir = 7, skor = skor + 10 poena
- Na kraju, skor igrača se registruje u **highscore** tabeli.



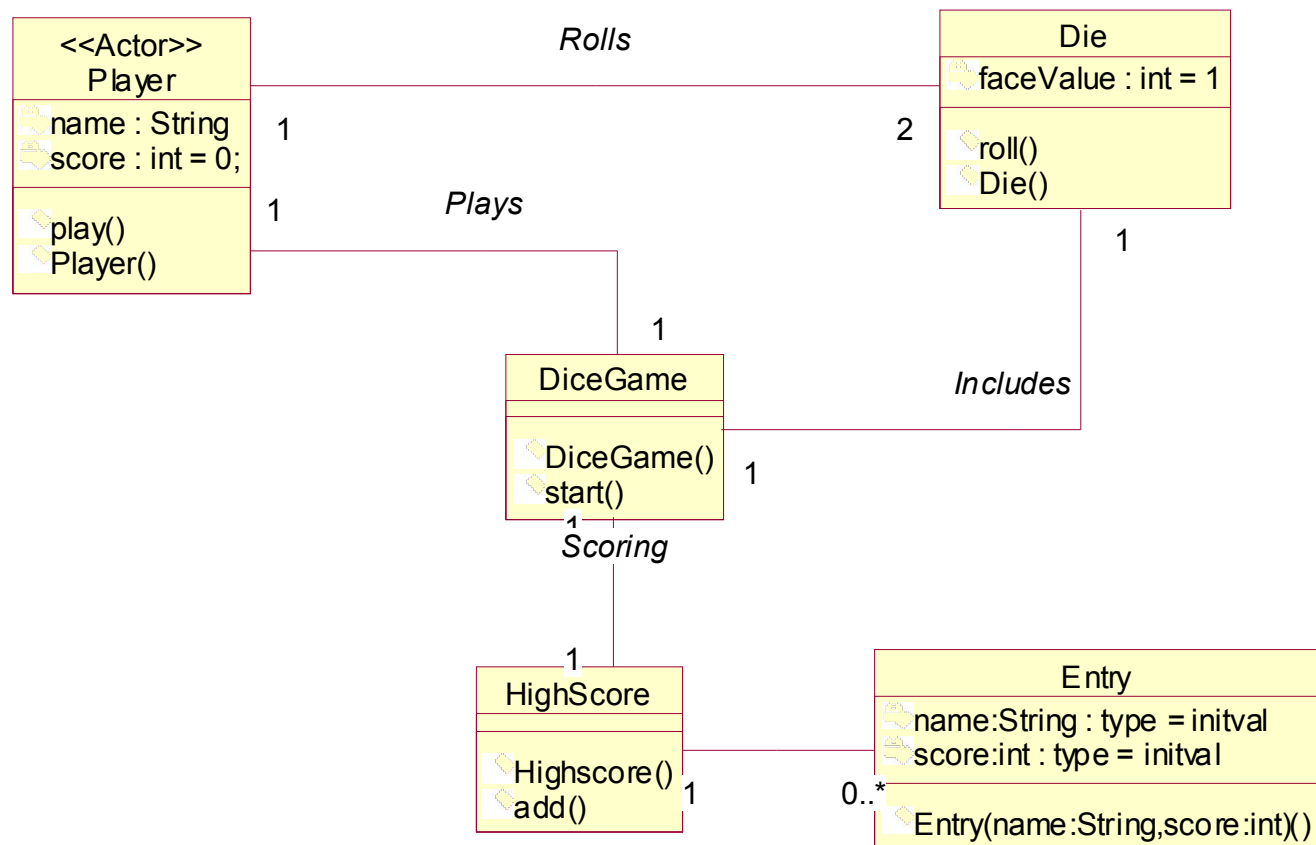




# Dijagram aktivnosti



# Faza analize...





## Faza projektovanja

- Projektovanje korisničkog interfejsa
- Rešavanje smeštaja rezultata iz highscore tabele (snimanje u datoteku ili u bazu podataka)
- Realizacija arhitekture po lejerima: Primena **Layer Architectural Pattern-a**



## Nivo (Layer)

- Organizacija sistema u nivoe omogućava strukturiranje i dekompoziciju sistema u grupe zadataka, pri čemu je svaka grupa jedan nivo apstrakcije.



# Layer :Struktura



# Layer: Struktura projektnog obrasca



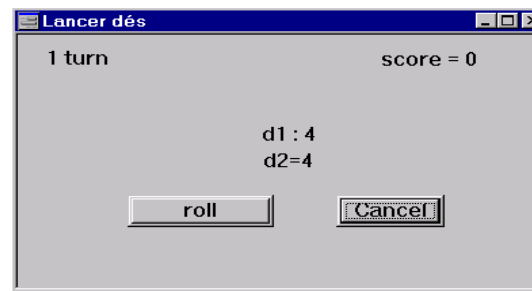
# Layer arhitekturni obrazac : Primeri

- JVM – Java Virtuelna Mašina i binarni code format
- API : Lejer koji enkapsulira niže nivoe
- Informacioni Sistemi
  - Presentacioni nivo, Logika aplikacije, Nivo domena aplikacije, Baza podataka
- Windows NT
  - Sistemski servisi,
  - Upravljanje resursima (Object manager, security monitor, process manager, I/O manager, VM manager, LPC),
  - Kernel (exception handling, interrupt, multipro synchro, threads),
  - HAL (Hardware Abstraction Level)
  - Hardware

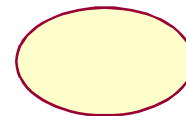


# Primena Layer Architecture obrasca

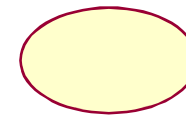
UI



Core

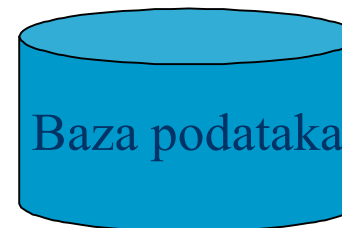


Play



View High Score

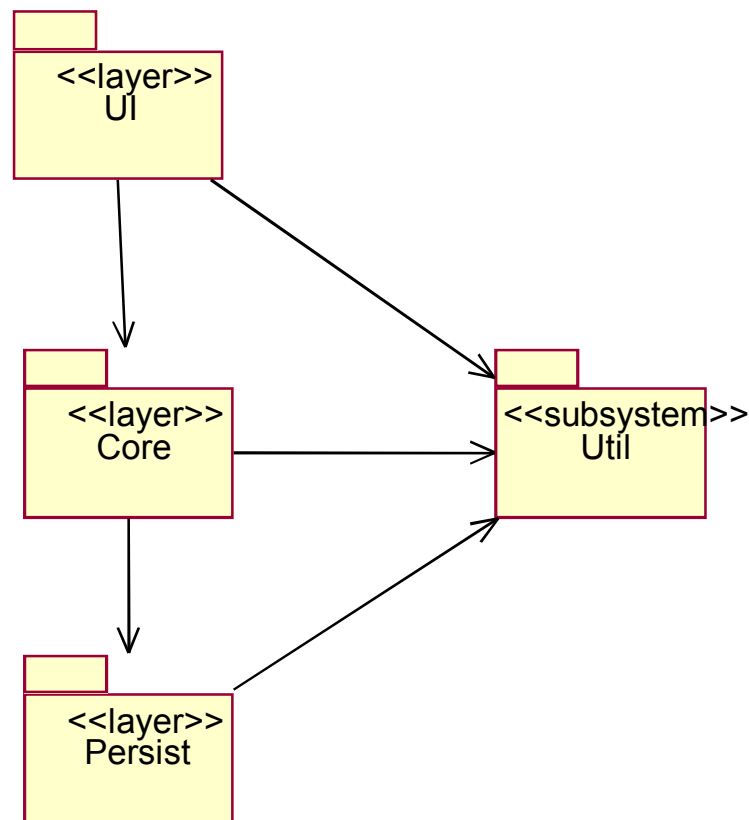
Perzistencija







# Primena Layer Architecture obrasca





## Layer « core »

- Sadrži klase koje definišu logiku rada aplikacije...
- Adaptacija klasa iz faze analize za implementaciju...
- Korišćenje Singleton projektnog obrasca...

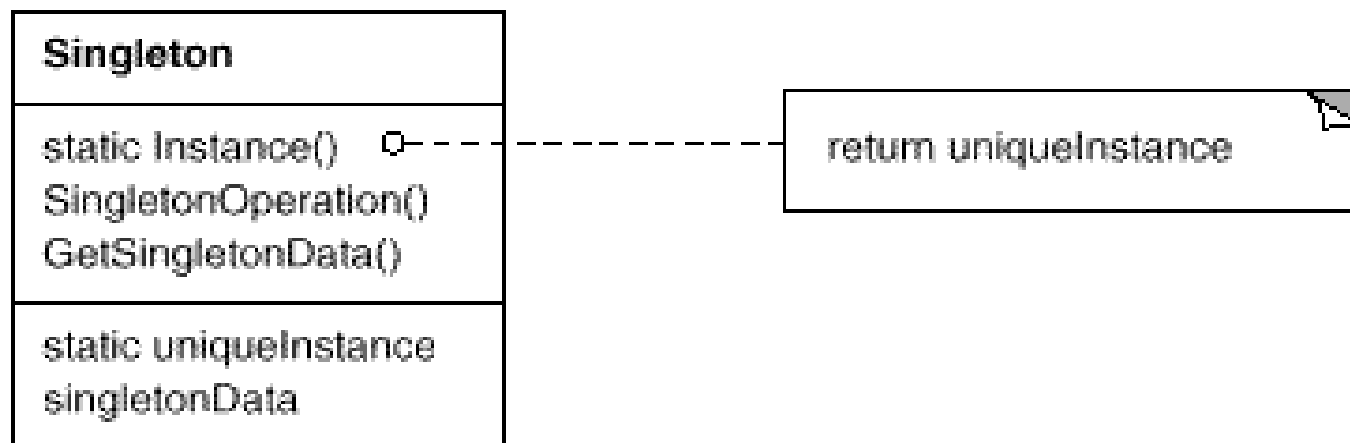


# Singleton obrazac

- Obezbeđuje da klasa ima samo jednu instancu, kao i globalni pristup toj instanci.

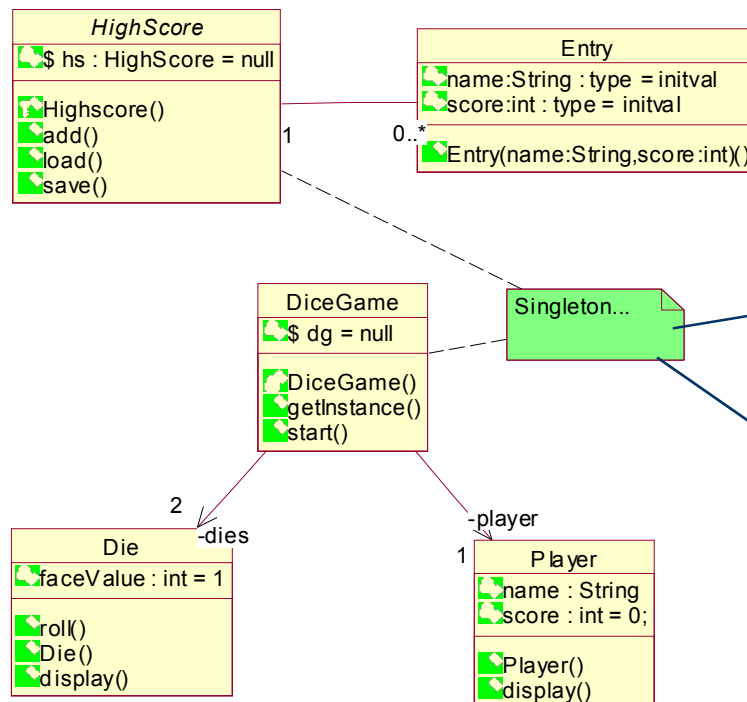


# Singleton Struktura

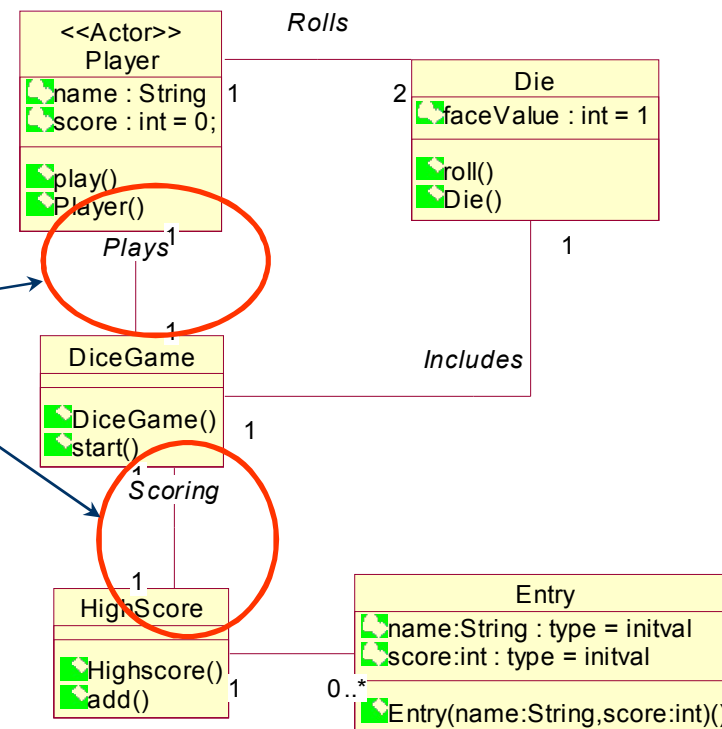


# Core « Layer »: dijagram klasa

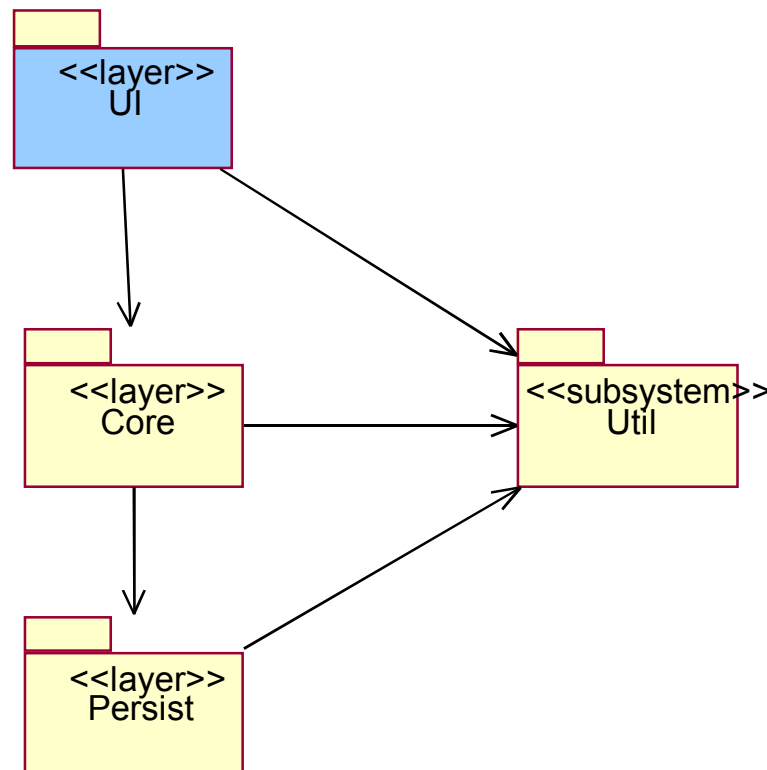
## Projektovanje



## Analiza

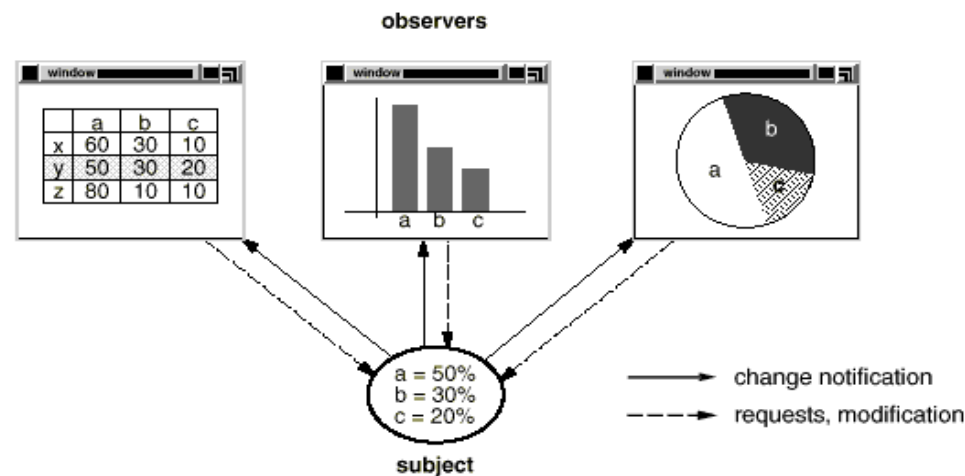


# Lejer Korisnički interfejs

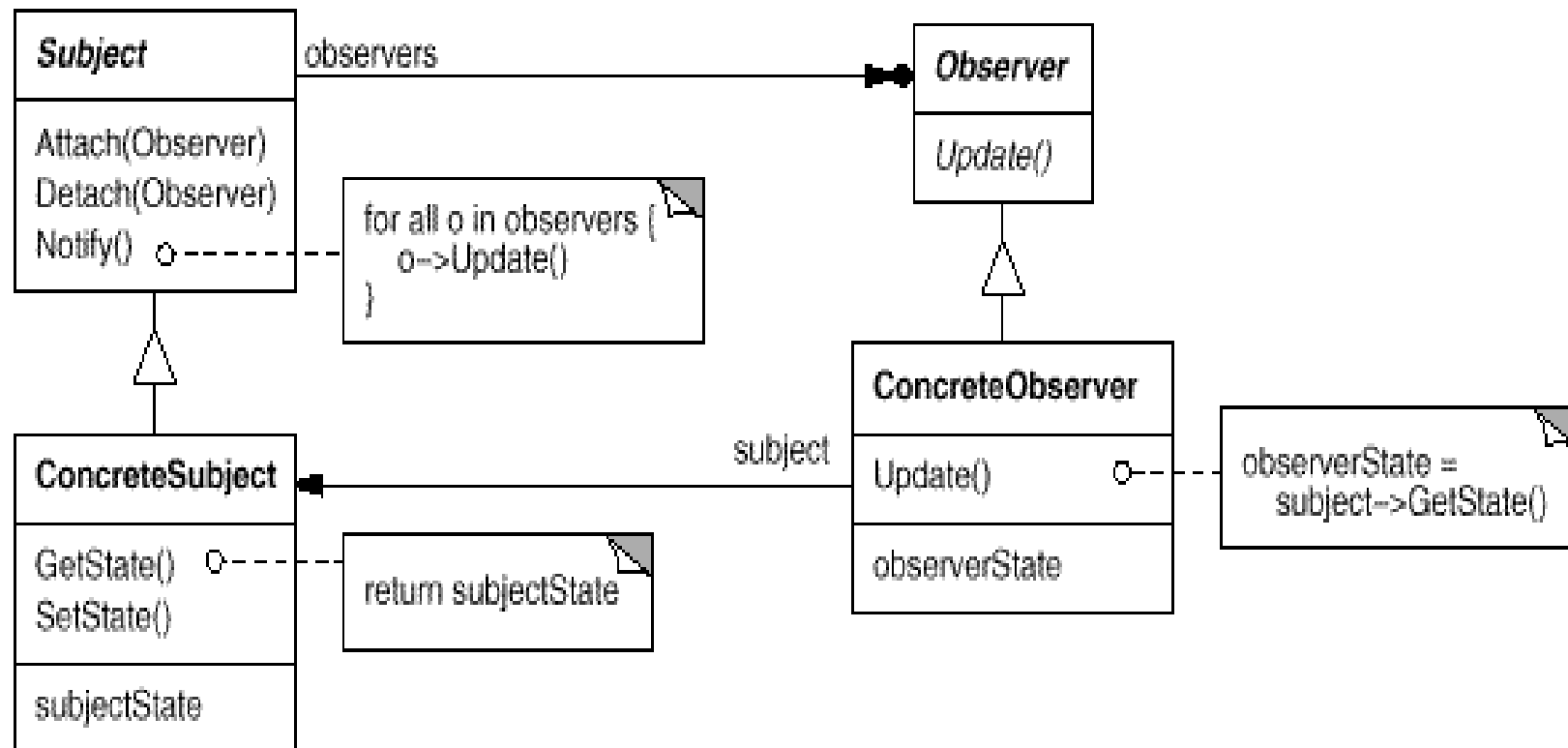


# Observer pattern

- **One-to-many** zavisnost između objekata: promena jednog objekta će izazvati obaveštavanje svih opservera

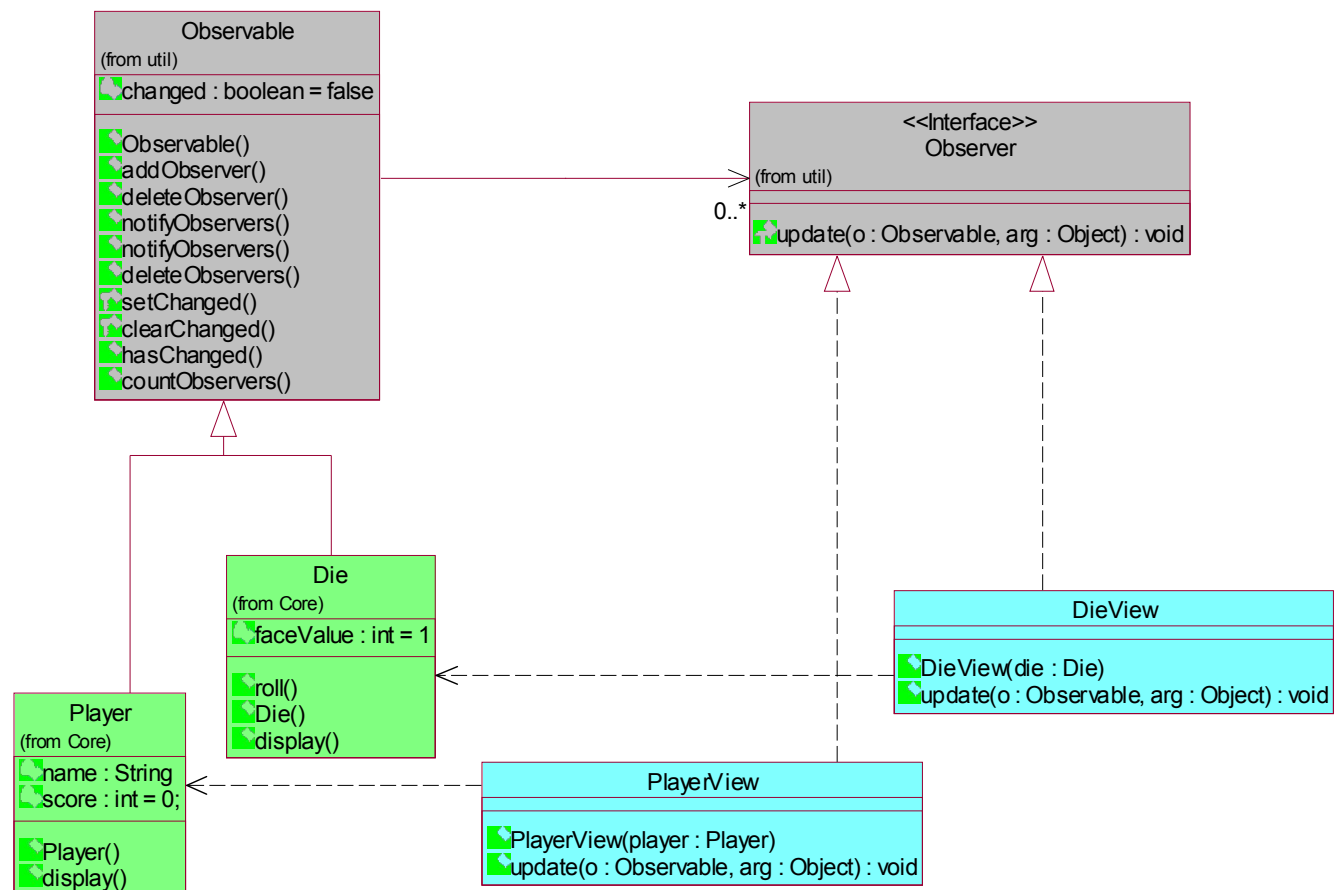


# Observer: Struktura

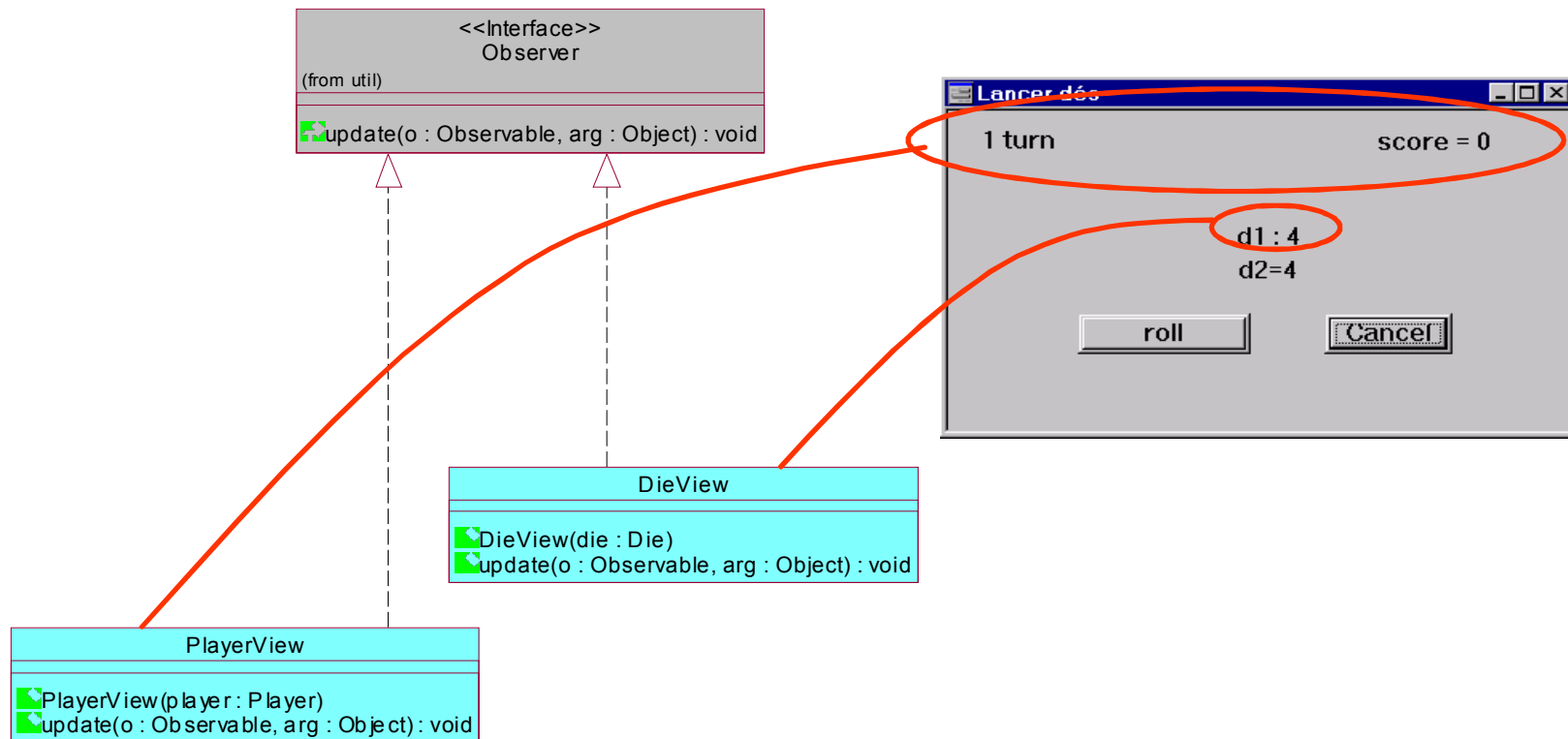




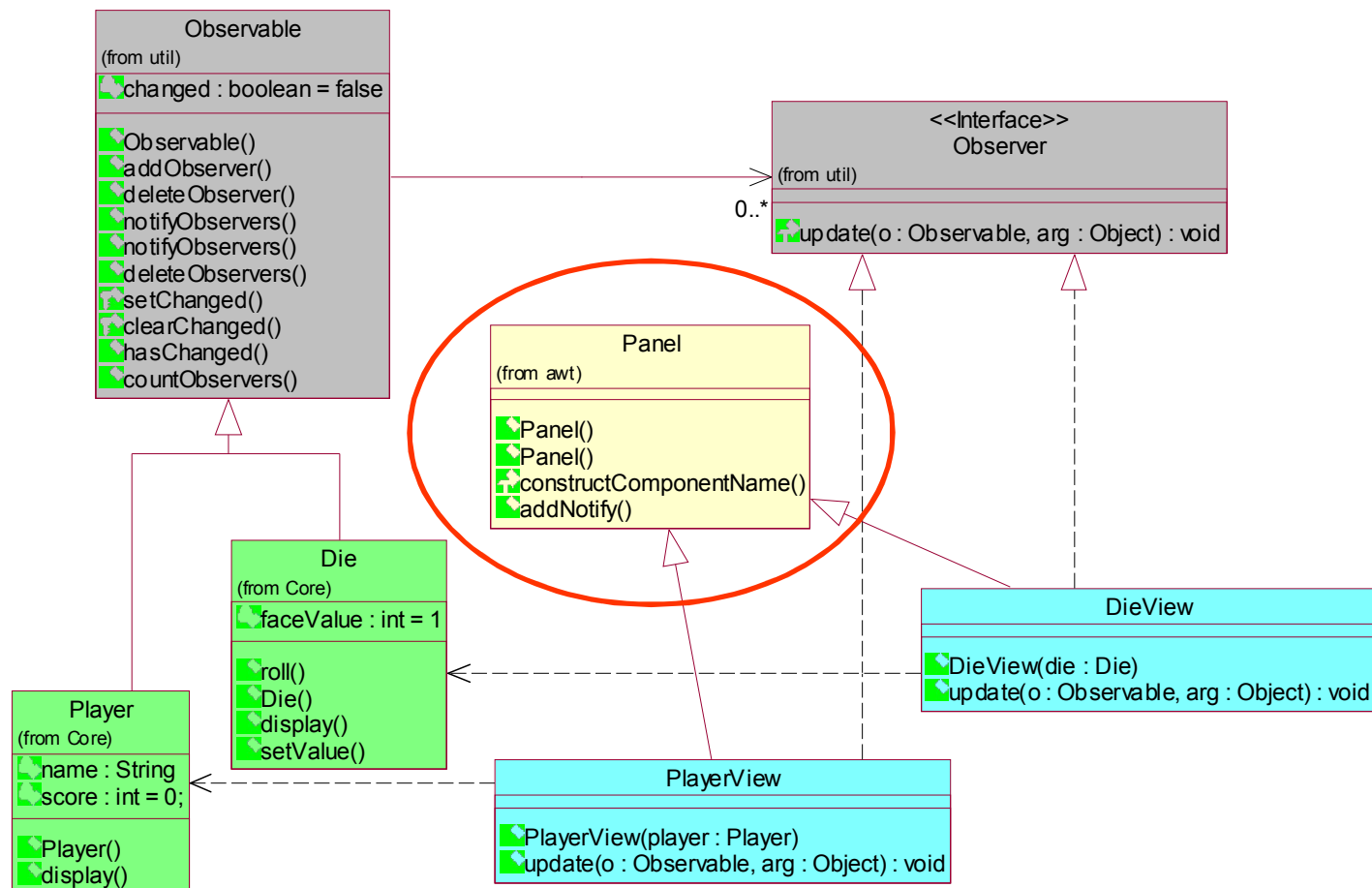
# Primena Observer Pattern-a



# Observer View

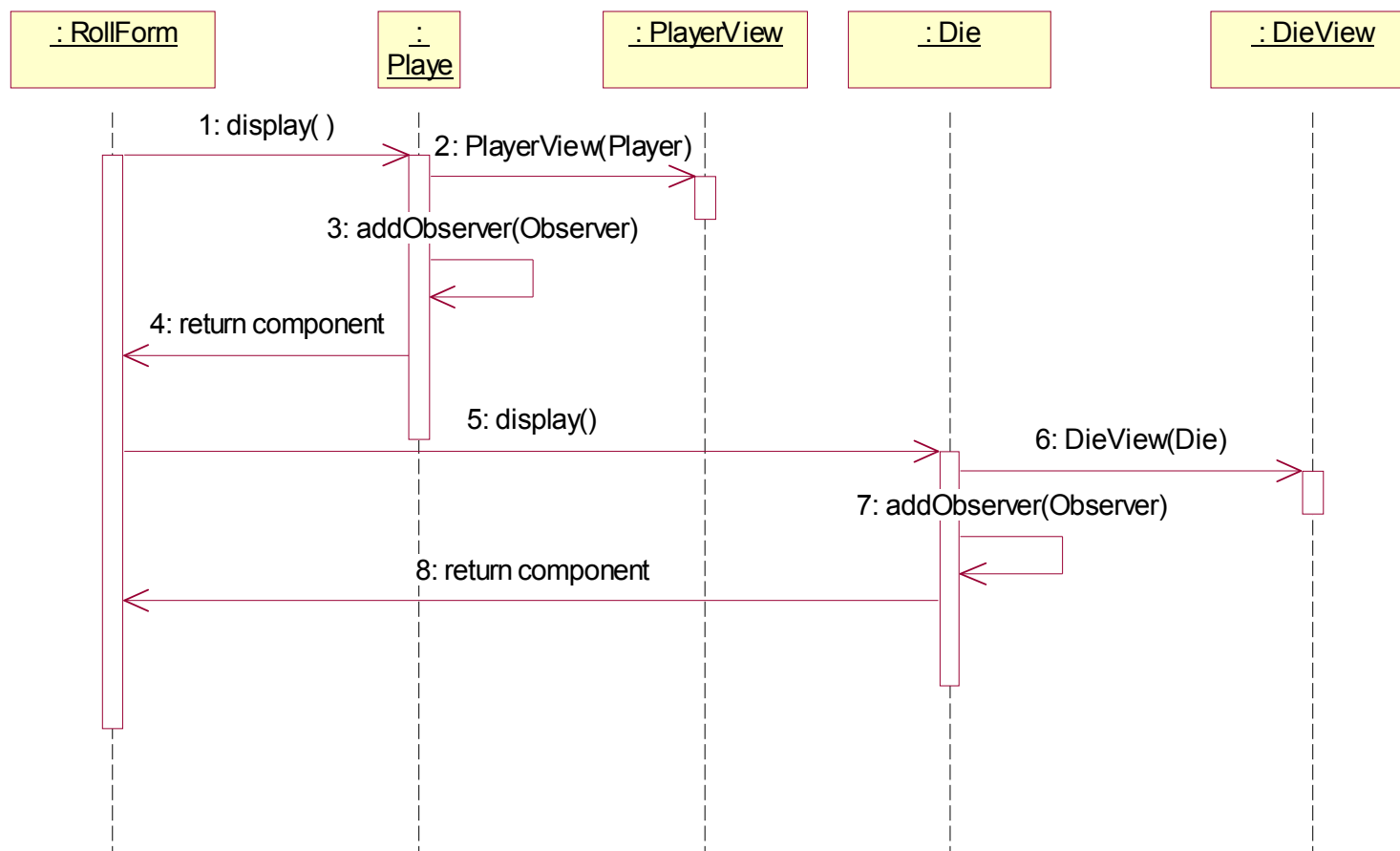


# Primena Observer Pattern-a

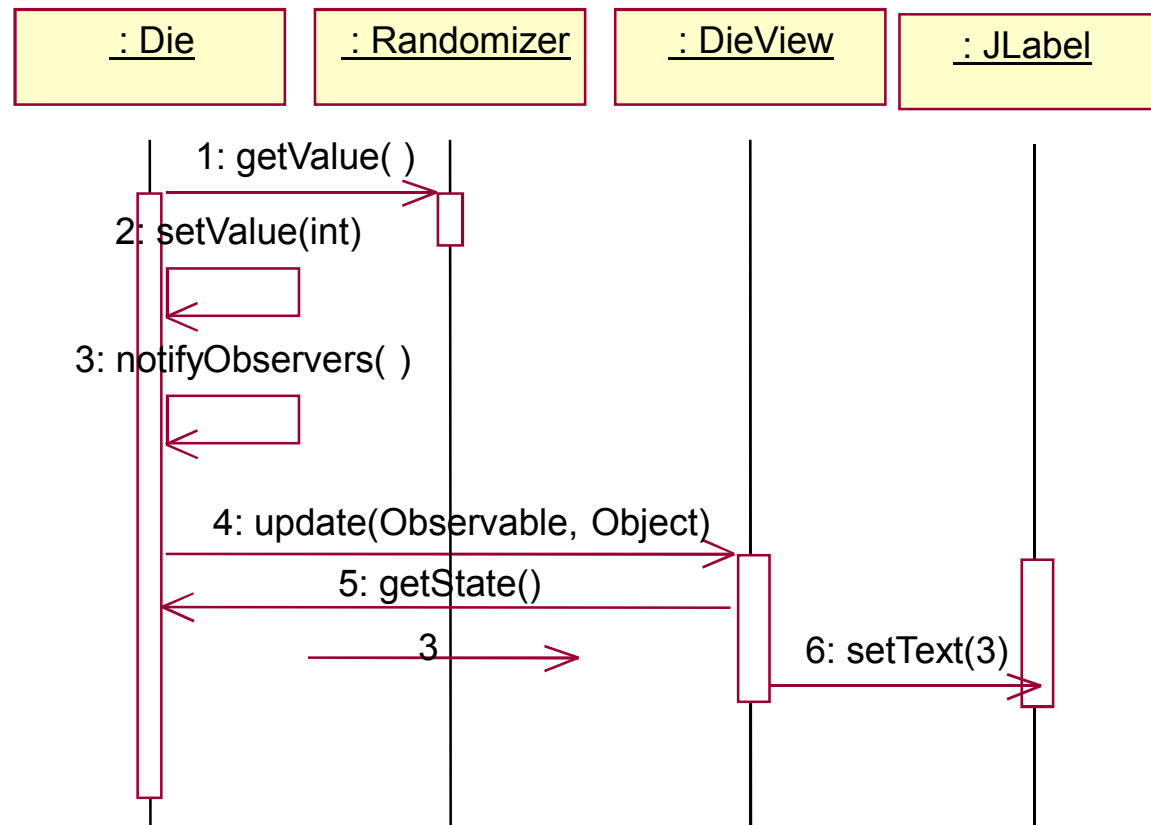




# Dijagram sekvence za postavljanje Observera



# Observer : obaveštavanje o promeni

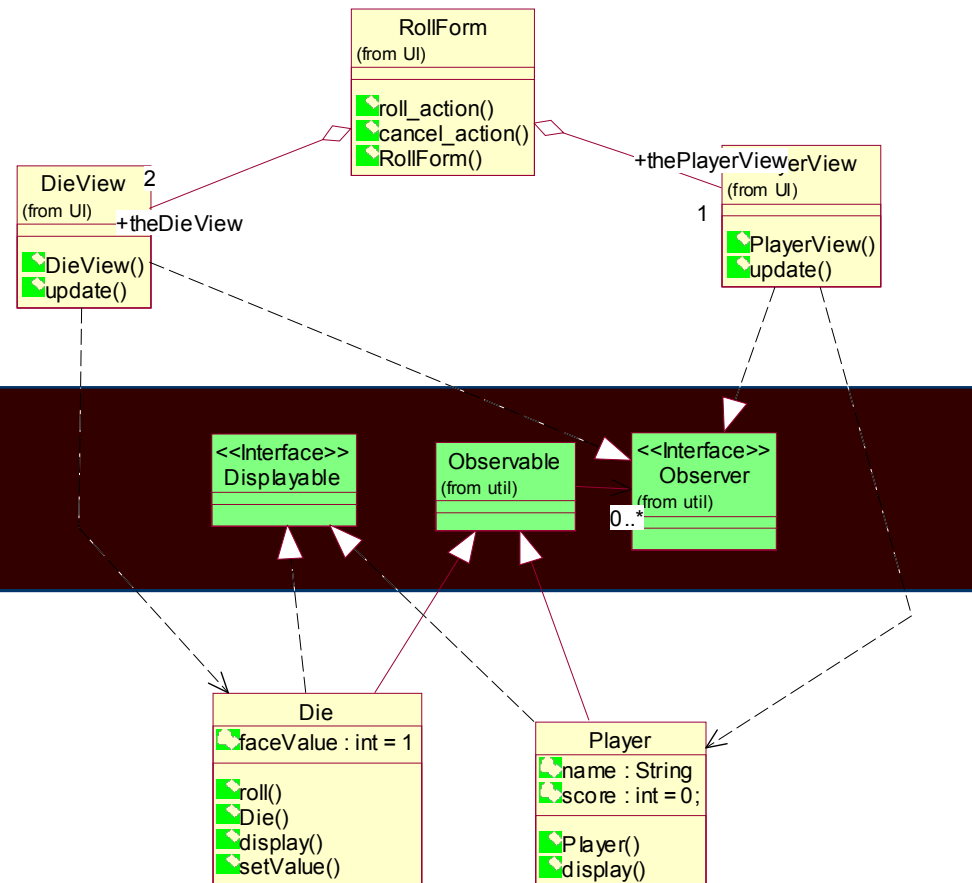


# Nivoi...

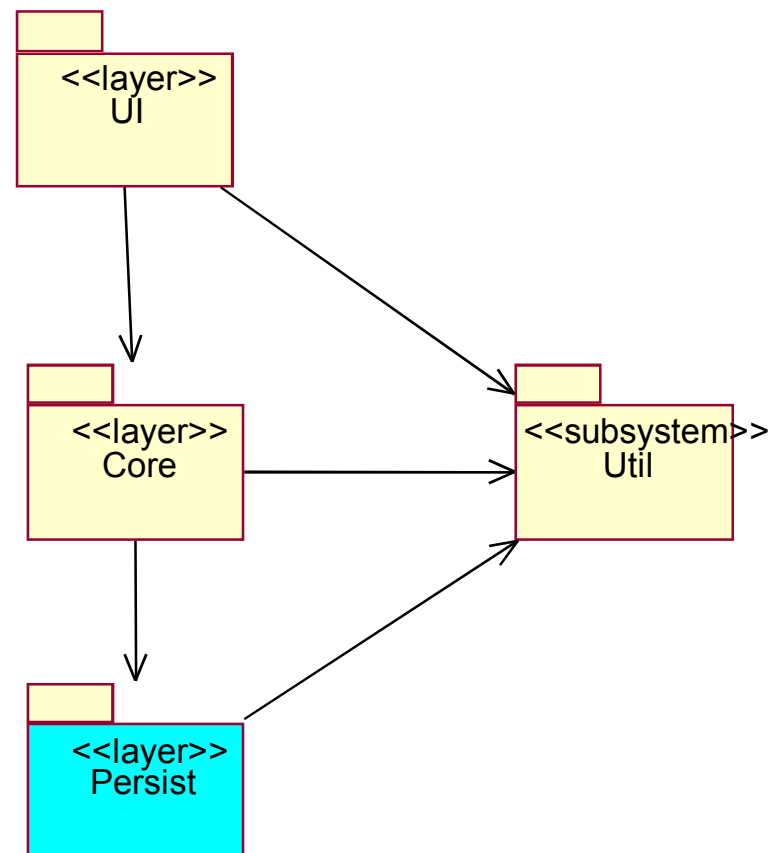
UI

Razdvajanje klasa i  
interfejsa

Core



# Smeštaj podataka...





# Obrazac Factory Method

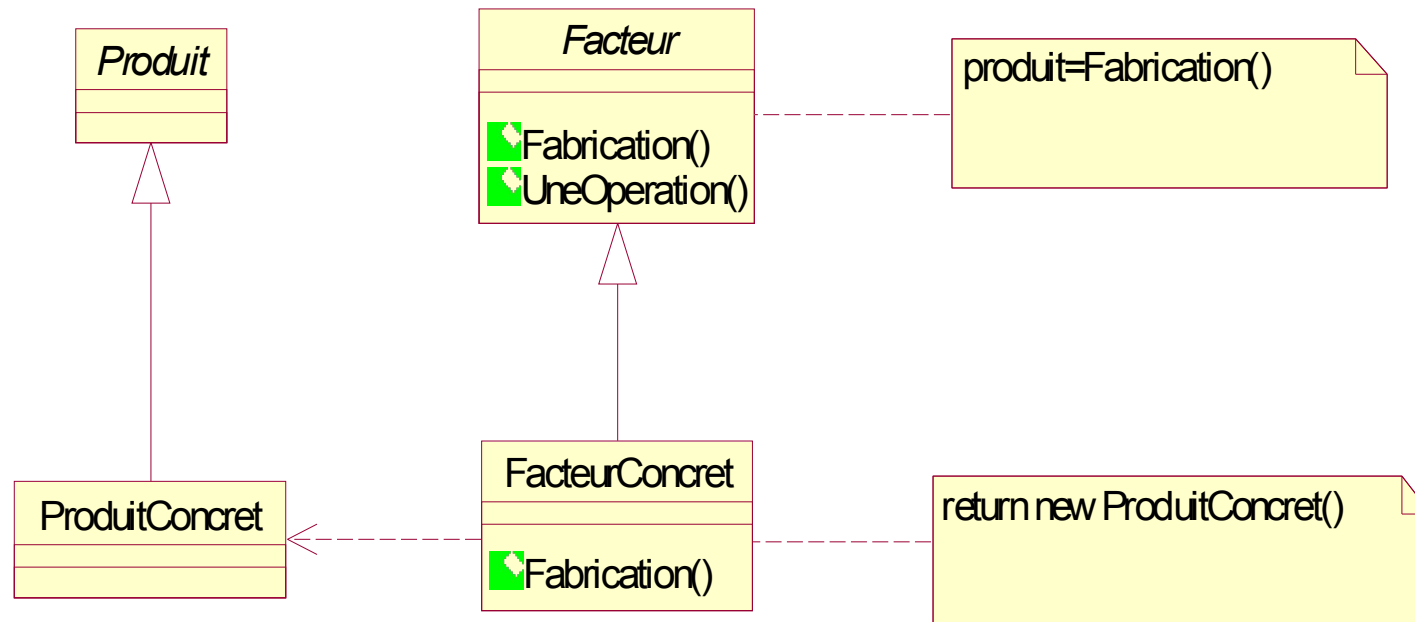
## ■ Namena

- Definiše interfejs za kreiranje objekta, ali prepušta potklasama da odluče koju će klasu instancirati.
- Omogućava klasi da odloži instanciranje i taj posao prepusti potklasama
- Poznat je i kao **Virtual Constructor**





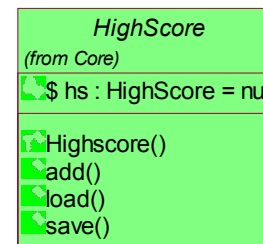
# Struktura



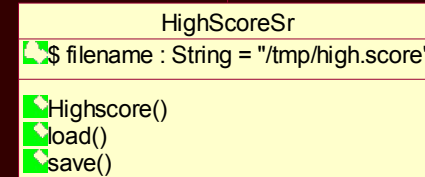
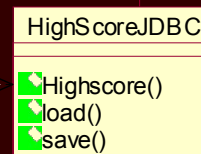


# Primena Factory Method obrasca

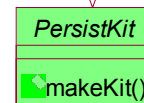
Apstraktni proizvod →



Konkretni proizvod



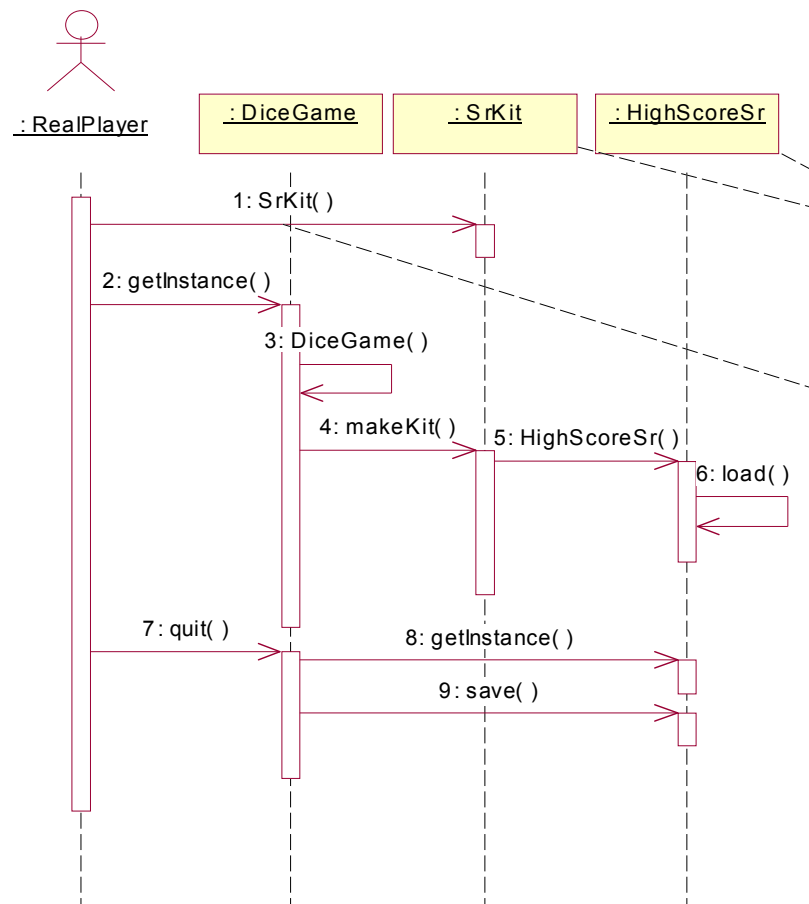
Konkretna fabrika



Apstraktna fabrika →



# Primena Factory Method obrasca





# Zaključak

- 1 Arhitekturni obrazac : **Layer**
- 3 Projektna obrasca : **Singleton, Observer, Factory Method**