



Lekcija 10 - Dizajn paterni

Šta su dizajn paterni i koje su im ključne osobine?

- Ovde je dat citat iz knjige u originalu:

"...names, abstracts, and identifies the key aspects of a common design structure that makes it useful for creating a reusable object-oriented design."

- Ključne osobine dizajn paterni:
 - ime
 - opis problema i konteksta
 - opis rešenja
 - konsekvence primene paterni – opisuje razna ograničenja, prednosti i mane korišćenja pojedine varijante rešenja (paterni uglavnom daju i alternativne implementacije za isti problem)
- Paterni se klasifikuju na osnovu dva kriterijuma: opsega (*scope*) i svrhe (*purpose*)

Elektroenergetski softverski inženjering – Razvoj EE softvera - 2016

2

Razlika između paterni i idioma je u nivou apstrakcije. Uglavnom se rutinski šabloni korišćeni tokom zapisa programa tretiraju kao programski idiomi. Dizajn paterni su slični šabloni ali za dizajn.

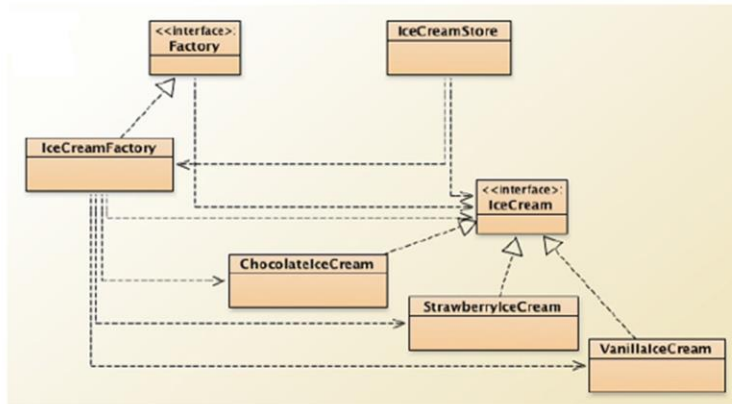
Opseg može podeliti paterne na osnovu relacija između klasa/objekata: na statičke veze između klasa/objekata, i dinamičke. Svrha, s druge strane, definiše da li se patern pretežno bavi načinom kreiranja objekata, kompozicijom klasa/objekata ili distribucijom funkcionalnosti sistema u vidu interakcije objekata.

Na osnovu pomenuta dva kriterijuma, dizajn paterni se dele na sledeće 3 velike klase: kreacioni, strukturalni i funkcionalni.

Dizajn paterni se mogu smatrati i nekom vrstom *quality assurance*-e, jer smanjuju šansu pravljenja grešaka u programu (oni daju proverena rešenja za česte probleme u praksi).

Primeri kreacionih dizajn paterna

- *Singleton* – videti listing iz knjige
- *Factory method* – videti listing iz knjige



Elektroenergetski softverski inženjering – Razvoj EE softvera - 2016

3

23 klasična paterna, koje spominje knjiga su (one u kurzivu su detaljno obrađene):

Kreacioni

Abstract factory

Builder

Factory Method

Prototype

Singleton

Strukturni

Adapter

Bridge

Composite

Decorator

Façade

Flyweight

Proxy

Funkcionalni

Chain of responsibility

Command

Interpreter

Iterator

Mediator

Memento

Observer

State

Strategy

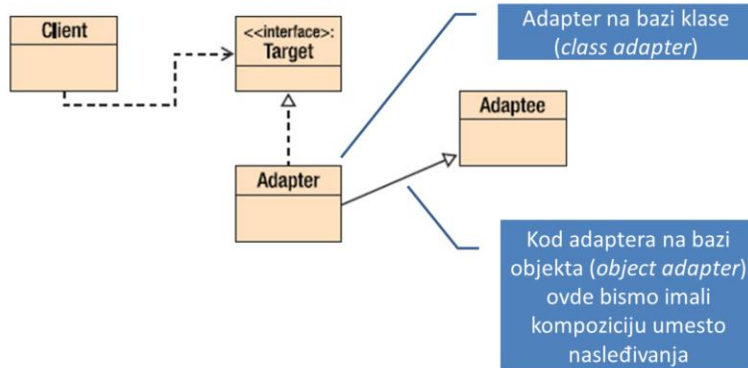
Template method

Visitor

Arhitekturni paterni (pipe-and-filter i model-view-controller) se takođe mogu smatrati i dizajn paternima.

Primeri strukturalnih dizajn paterna

- Adapter – videti listing iz knjige



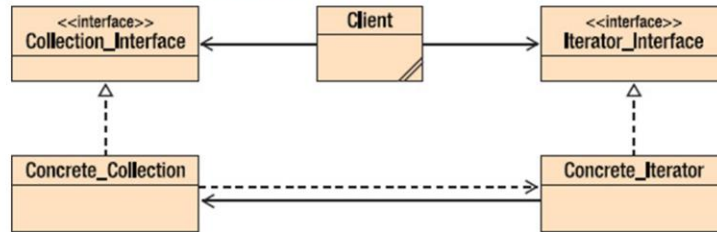
Elektroenergetski softverski inženjering – Razvoj EE softvera - 2016

4

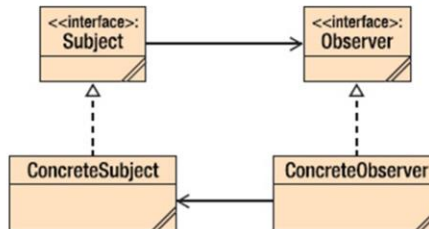
Najbolja analogija za adapter dizajn patern jesu razni adapteri za strujne priključke. Znamo da postoje različiti standardi za utičnice, i ako želimo naš uređaj priključiti u SAD onda moramo imati adapter, koji će prilagoditi tamošnju utičnicu našem uređaju.

Primeri funkcionalnih dizajn paterna – deo 1

- Iterator – videti listing iz knjige



- Observer – videti listing iz knjige



Elektroenergetski softverski inženjering – Razvoj EE softvera - 2016

5

U Java SE 5 je prvi put uveden interfejs `Iterable`. Svaka kolekcija koja implementira ovaj interfejs može da se iterira korišćenjem `for-each` konstrukcije. Ovakav vid iteracije je često zgodniji zbog kompaktnosti zapisa.

U Java biblioteci postoje dva entiteta za Observer patern: `Observable` (igra ulogu subjekta) i `Observer` (ista uloga kao u primeru). Kao što vidimo, neki dizajn paterni su sastavni delovi biblioteka i mogu se smatrati već i idiomima na nivou jezika.

NAPOMENA:

Primer iz knjige nije implementiran do kraja, tj. ne postoji mogućnost da klijent kreira iterator vezan za kolekciju.

Primeri funkcionalnih dizajn paterna – deo 2

- Strategy – videti listing iz knjige

