



JAVA **FAKTURA**

S01: Java is the new Black

E04: Design Patterns

Tobiasz Kowalski

07/05/2019

About me



- Java Dev in IDEMIA
 - co-organizer Javafaktura
 - SDA bootcamp trainer
 - Lecturer at Technical University of Lodz
-
- [@TobKowalski](https://www.linkedin.com/in/tobiasz-kowalski/)
 - <https://www.linkedin.com/in/tobiasz-kowalski/>



My approach

Application description



Business Technical

Implement pizza creation function

Each pizza has:

- 3 layers
 - dough
 - sauce
 - toppings
- properties
 - name
 - size

Each pizza must contain at least one topping.

Default values

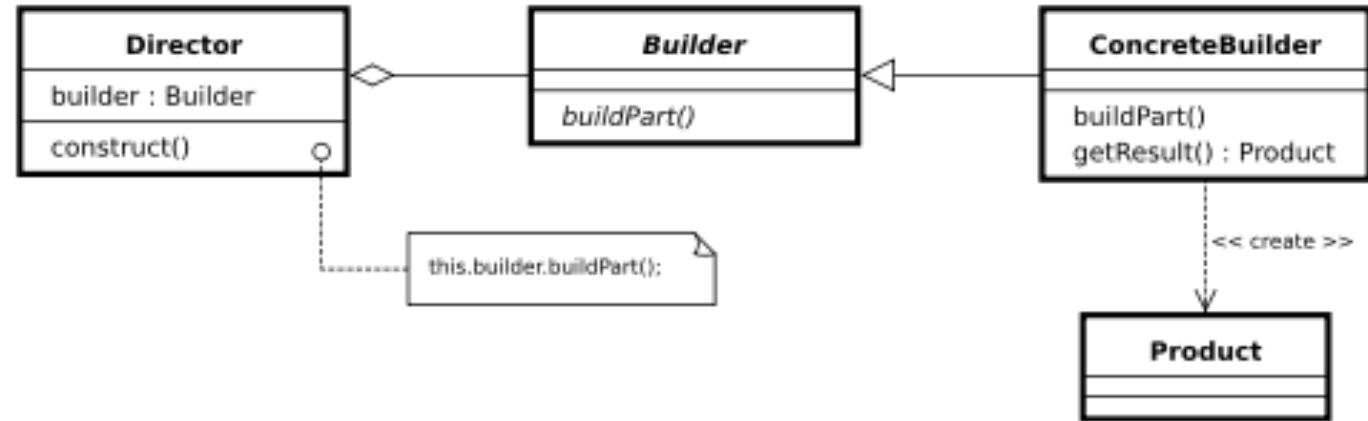
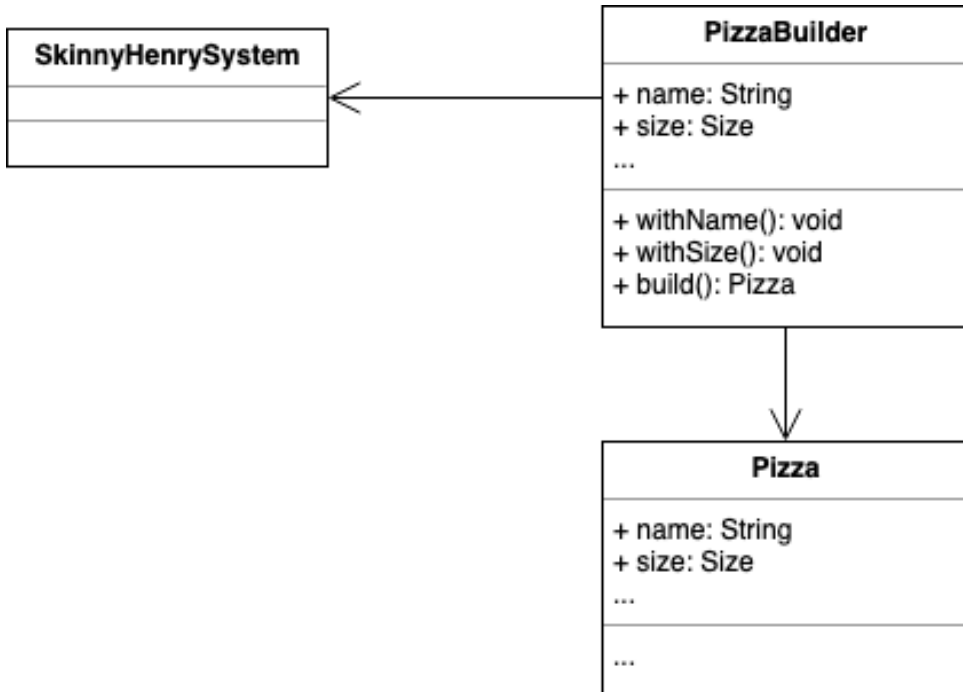
name - John

dough - italian

sauce – tomato&cheese

size - medium

Builders comparison



Builder tools

<https://plugins.jetbrains.com/plugin/6585-builder-generator>

<https://plugins.jetbrains.com/plugin/7354-innerbuilder>

<https://immutables.github.io/>

Calculate the price of pizza

Small:

- up to 2 toppings = 18 PLN
- 3 – 5 toppings = 22 PLN

Medium:

- up to 3 toppings and meat < 50% = 25 PLN
- up to 3 toppings and meat > 50% = 27 PLN
- 4 - 5 toppings and meat < 40% = 27 PLN
- 4 - 5 toppings and meat > 40% = 29 PLN

Extra large:

- up to 2 toppings = 50 PLN
- 3 – 5 toppings = 50 PLN

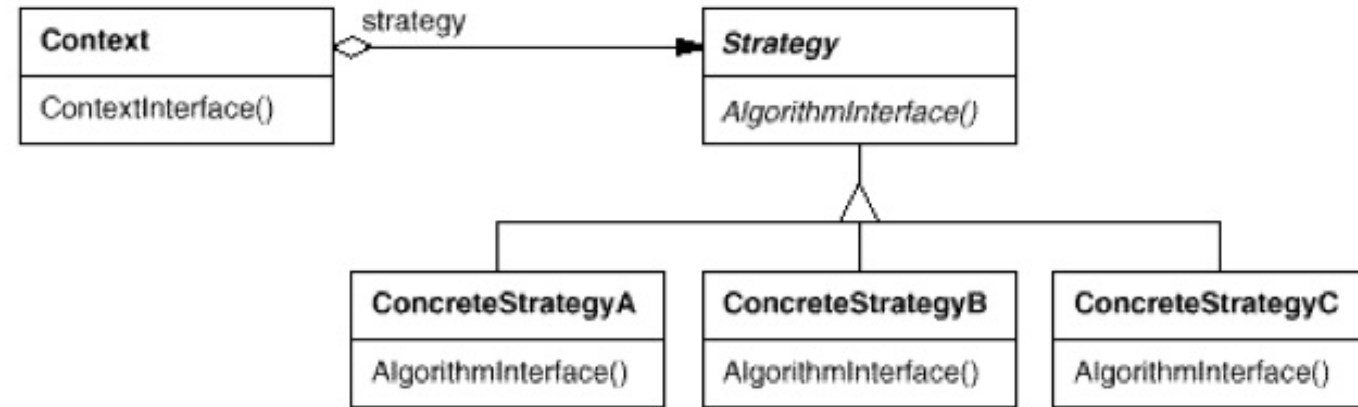
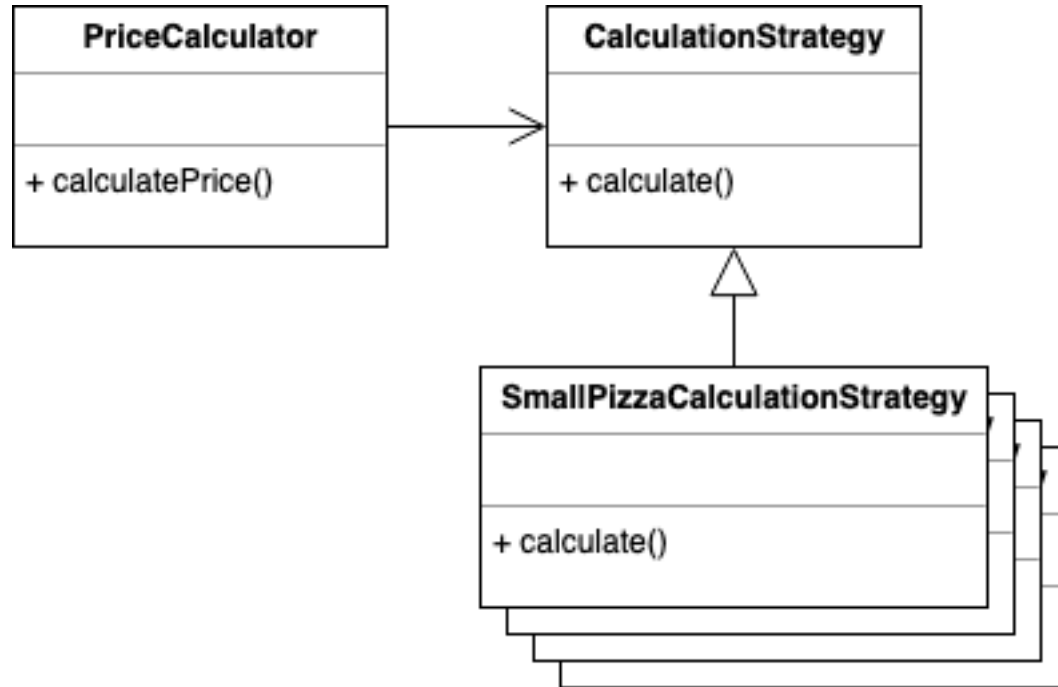
10 PLN extra for plumb dough

Large:

- up to 3 toppings and meat < 40% = 30 PLN
- up to 3 toppings and meat > 40% = 35 PLN
- 3 - 5 toppings without ananas = 37 PLN
- 3 – 5 toppings with ananas = 40 PLN

5 PLN extra for plumb dough

Strategy implementation comparison

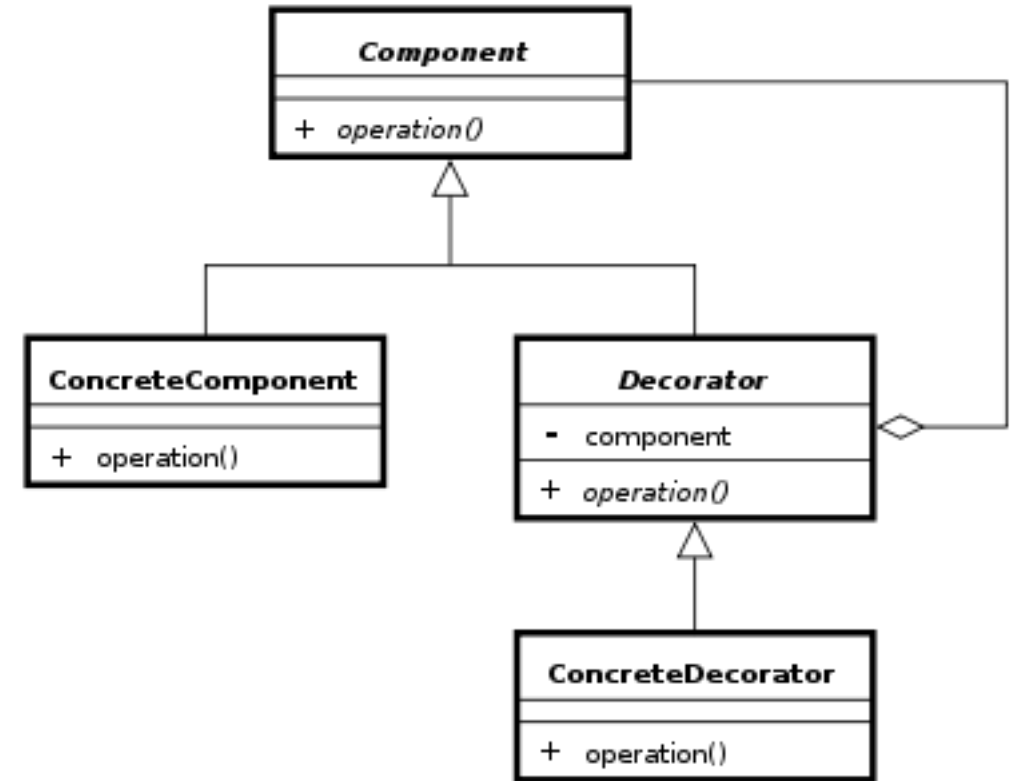
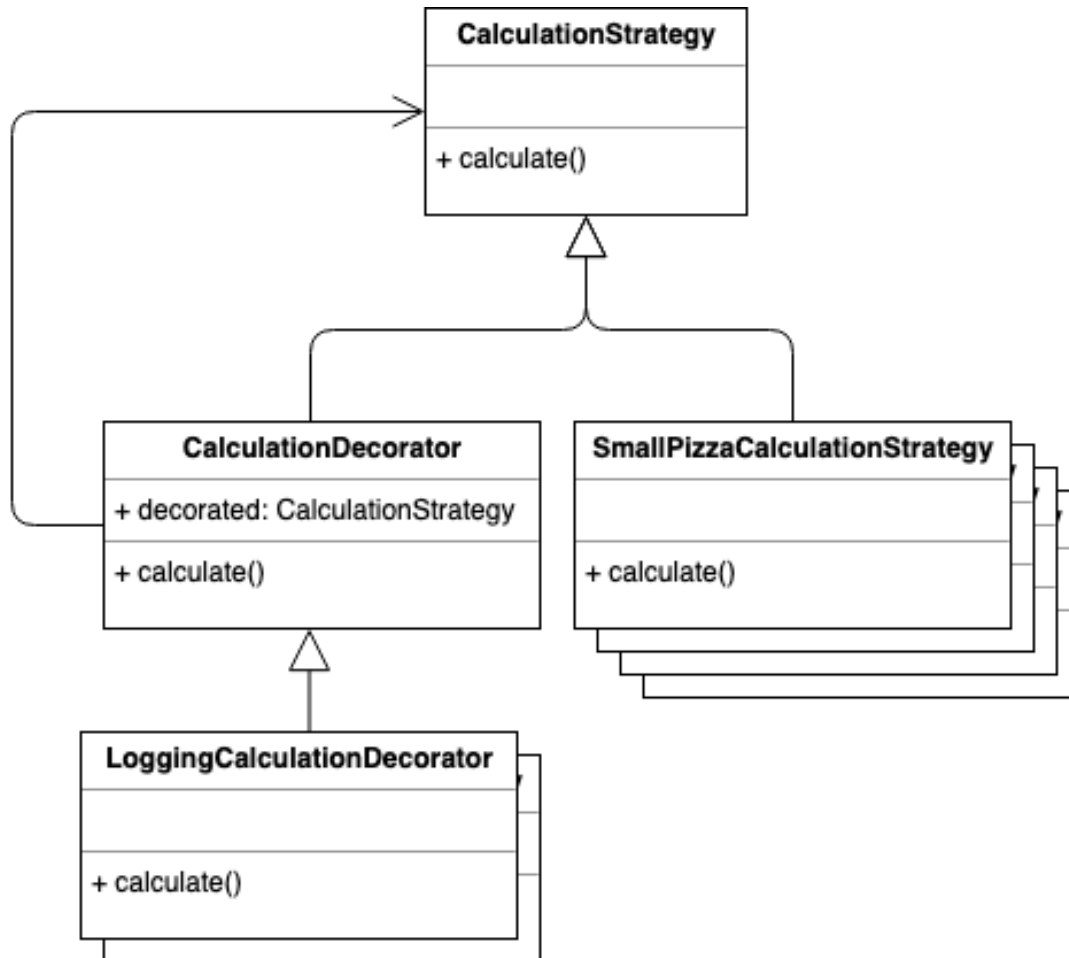


What is the name of this pattern?



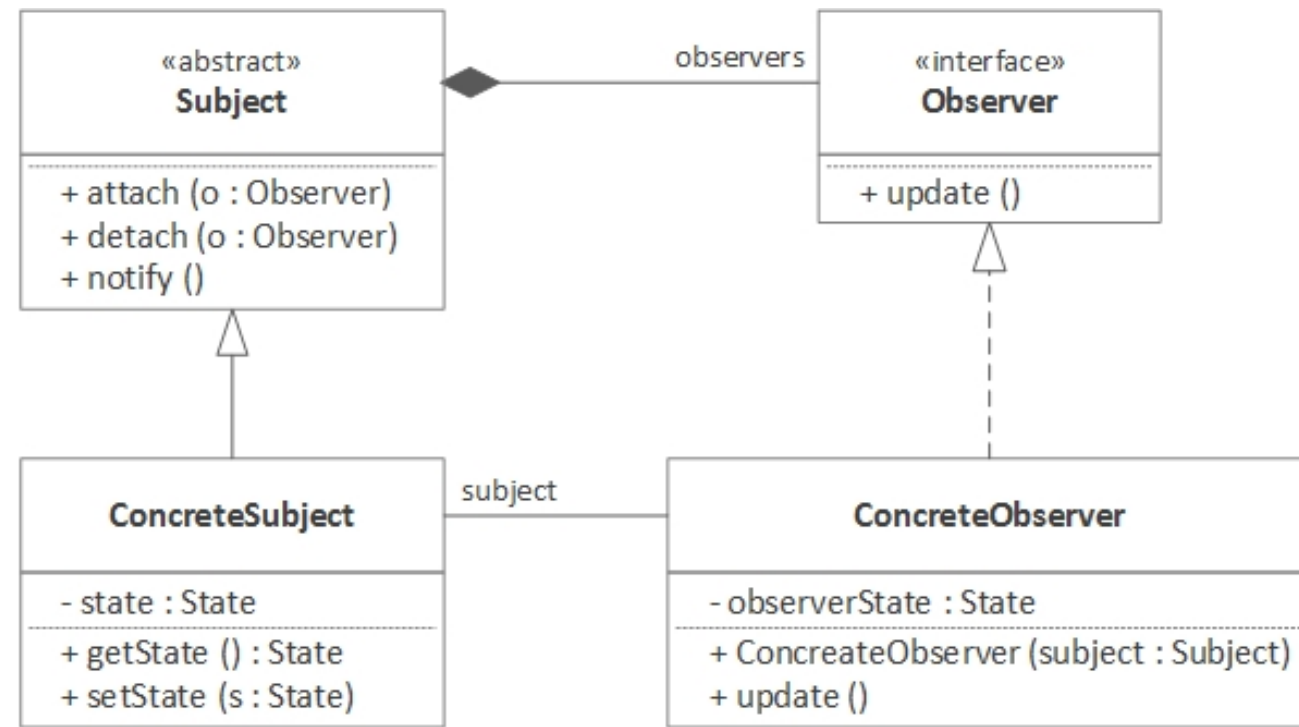
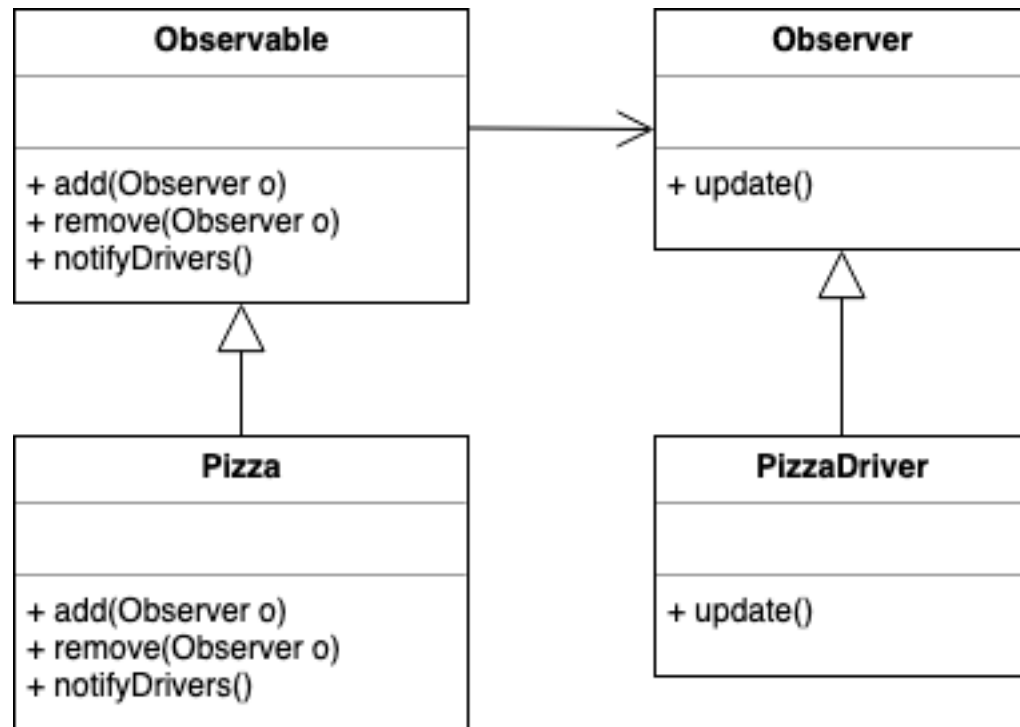
Add logging and execution time for pizza price calculation.

Decorator Implementation comparison



Each supplier should get notification when pizza is ready.
(UberEats, PizzaPortal, PysznePL)

Observer Implementation comparison



Creational

Structural

Behavioral

Levels of initiation

fabulist

scout/orc

wizard/thinker

Confetteria

Raffaello



method extraction

good naming

SOLID

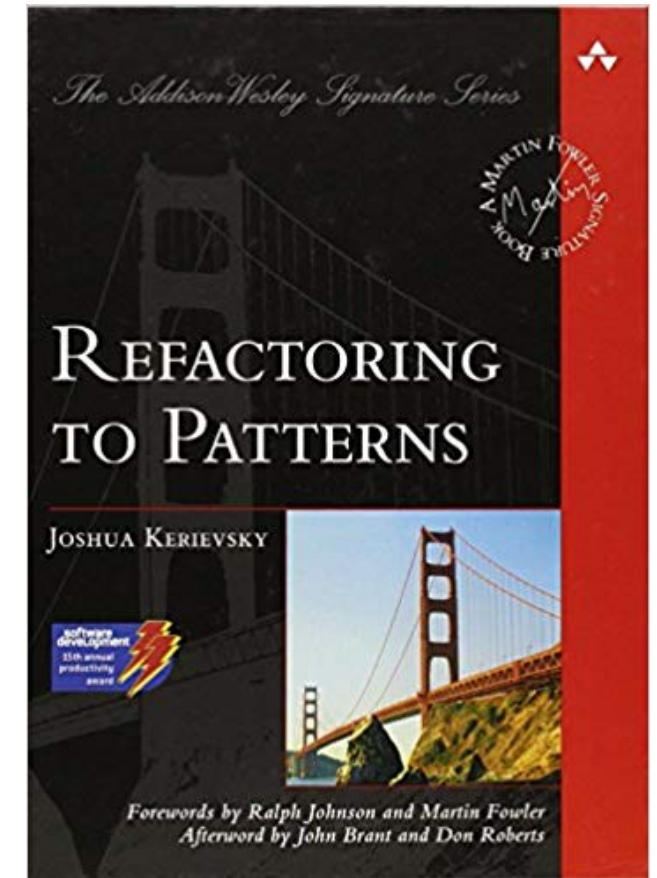
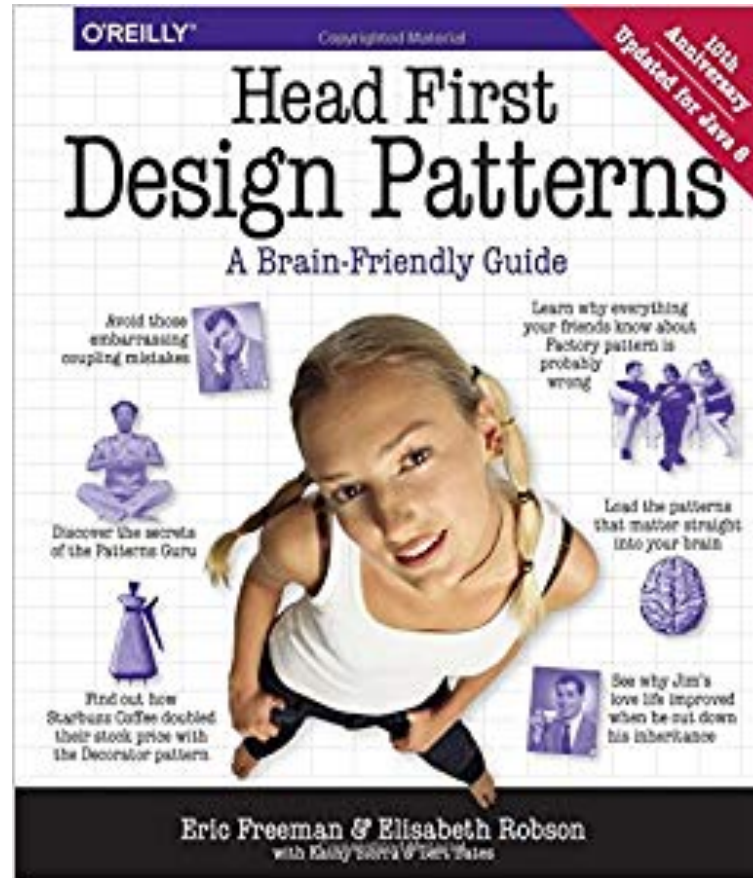
Cyclomatic complexity

testability

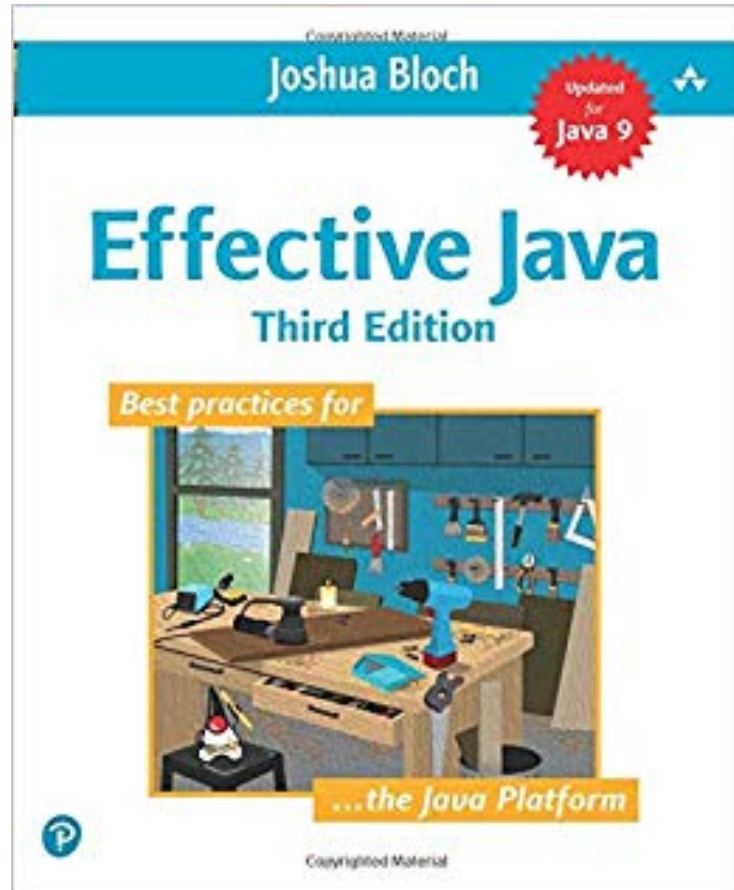
immutables

PROGRAMMING!!!

Books



Books



Useful links



<https://www.youtube.com/playlist?list=PLrhzvlcii6GNjpARdnO4ueTUAVR9eMBpc>

<https://github.com/iluwatar/java-design-patterns>

<https://stackoverflow.com/questions/1673841/examples-of-gof-design-patterns-in-javas-core-libraries>



Thanks



Survey

