

# **OOP paradigm, the basics of good design and architecture of Java applications**

Mikalai Alimenkou

<http://xpinjection.com>

@xpinjection



# Mikalai Alimenkou

@xpinjection

*Happy father, Java Tech Lead, ScrumMaster, XP Injection founder, speaker, Agile/XP Coach, conference organizer:*

*@jeeconf, @seleniumcamp, @xpdays\_ua, @itbrunch.*

Ukraine, Kiev · <http://xpinjection.com>

13,550

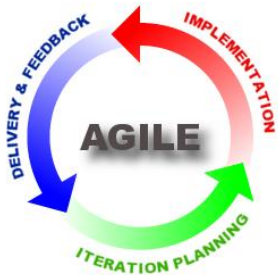
TWEETS

51

FOLLOWING

1,219

FOLLOWERS



**7 YEARS,  
EXPERT**



**9 YEARS,  
TECH LEAD**

**FOUNDER AND COACH:**



**ORGANIZER AND SPEAKER:**



# Disclaimer



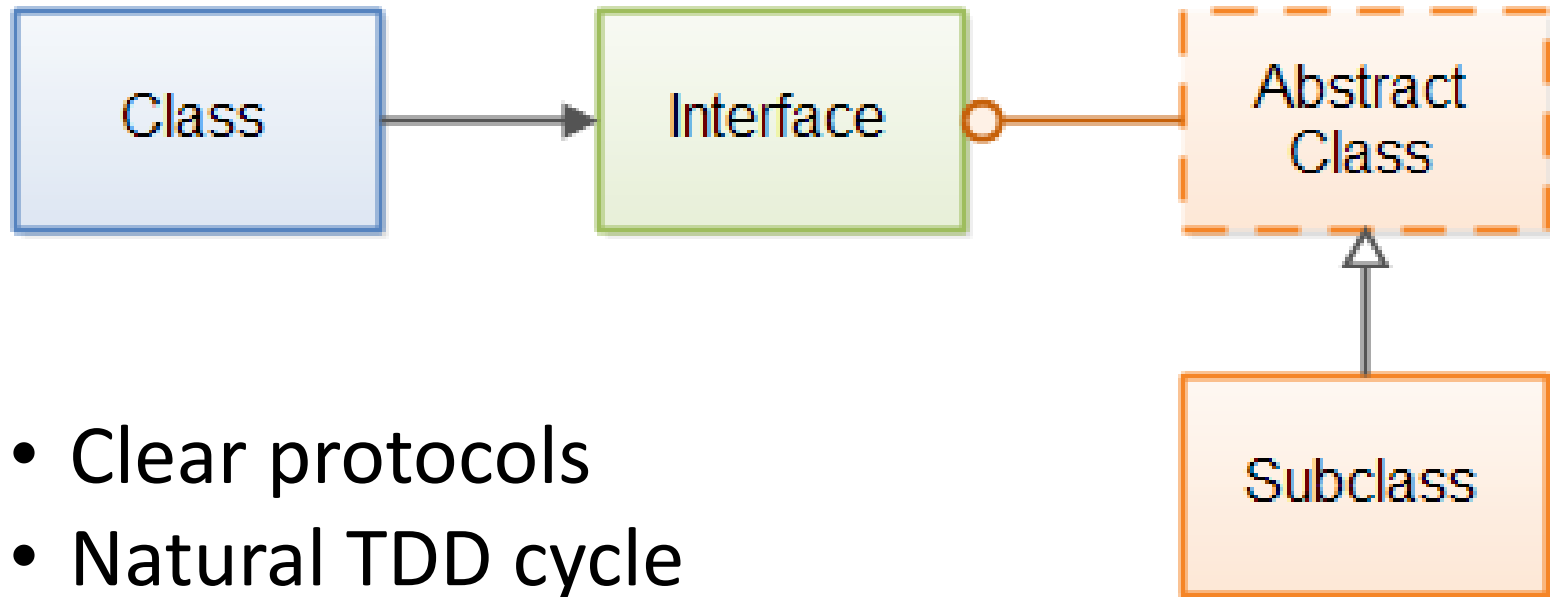
This is all my  
personal experience

TAP

TAP

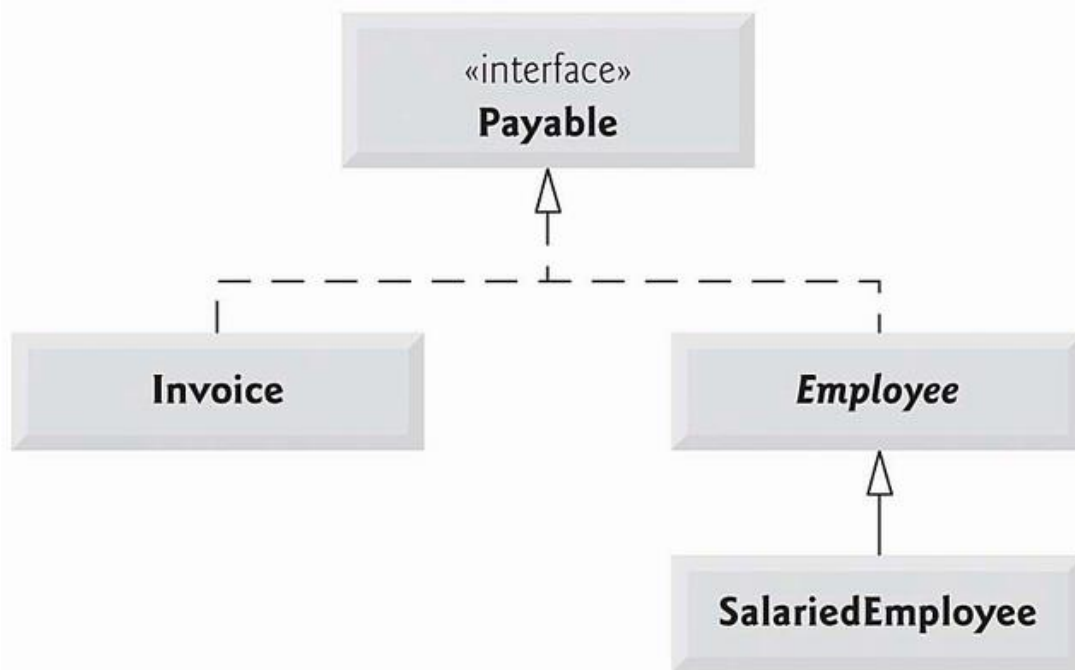
TAP

# #1. Use interfaces everywhere



- Clear protocols
- Natural TDD cycle
- Delayed transport decisions
- Easy logic extension
- Simple scalability

# #2. Inheritance is evil



- Current and future behavior inherited
- ‘Protected’ breaks encapsulation
- Long inheritance chains is hell
- Dependencies on frameworks inherited

# #3. 'Natural' encapsulation

```
1 package uk.co.planetjava.hibernate.bookstore;
2
3 public class Book {
4
5     private Integer id;
6     private String title;
7     private String author;
8     private String price;
9
10    public Book() {
11    }
12
13    public Integer getId() { return id; }
14    private void setId(Integer id) { this.id = id; }
15
16    public String getTitle() { return title; }
17    public void setTitle(String title) { this.title = title; }
18
19    public String getAuthor() { return author; }
20    public void setAuthor(String author) { this.author = author; }
21
22    public String getPrice() { return price; }
23    public void setPrice(String price) { this.price = price; }
24 }
25
```

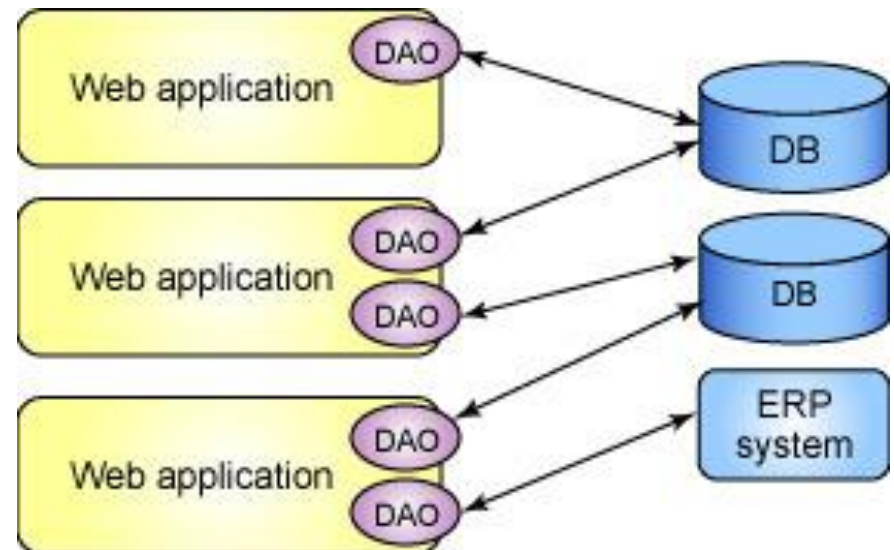
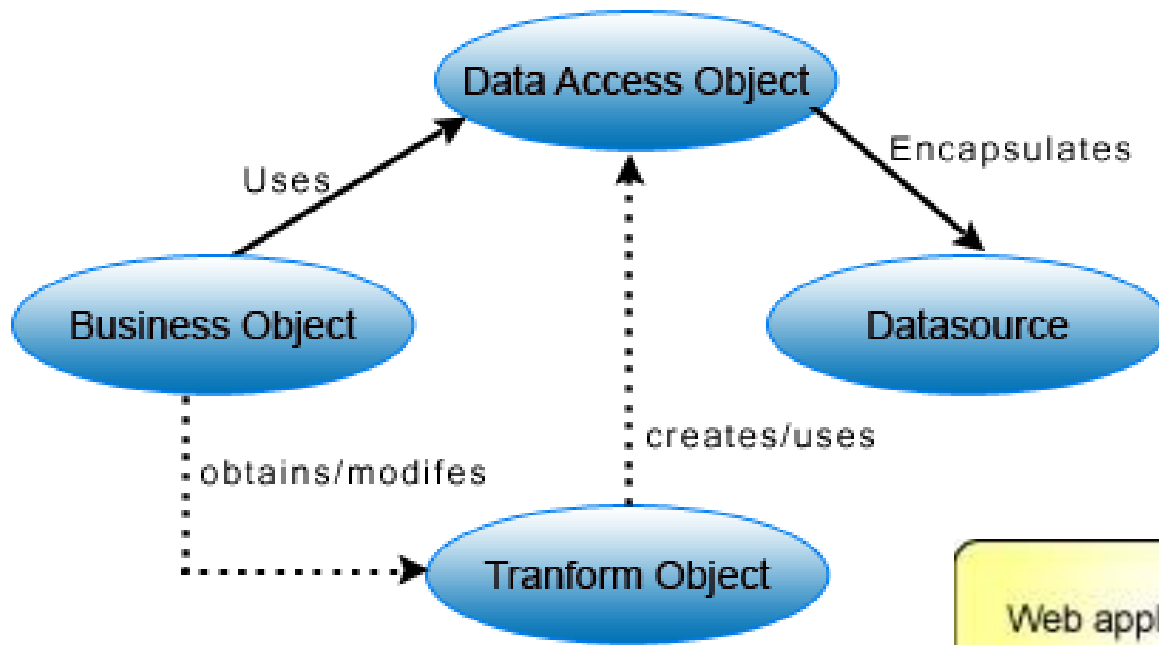
*POJO bullshit  
game*

# #4. SOLID design principles

- **SRP** – Single Responsibility Principle
- **OCP** – Open/Closed Principle
- **LSP** – Liskov Substitution Principle
- **ISP** – Interface Segregation Principle
- **DIP** – Dependency Inversion Principle

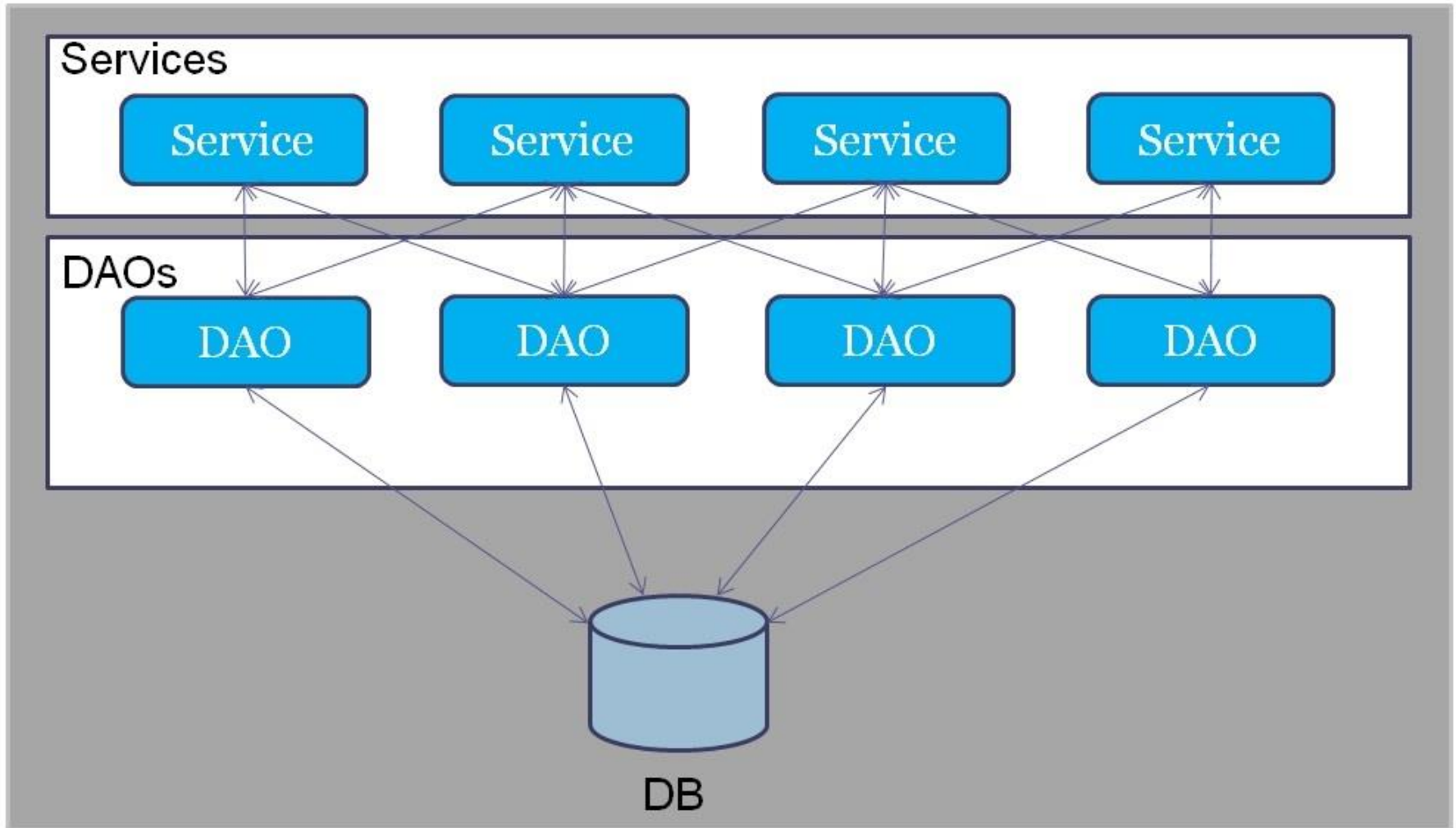
Don't forget about: **DRY + KISS + YAGNI**

# #5. DAO layer for data access





# #6. All logic is in services



# #7. Don't rely on frameworks



And remember...



**НЕЛЬЗЯ ПРОСТО ТАК ВЗЯТЬ**

**И НАПИСАТЬ КРУТОЙ JAVA КОД**

risc.ru-h.ru

Thank you!

*@xpinjection*

*<http://xpinjection.com>*

*[mikalai.alimenkou@xpinjection.com](mailto:mikalai.alimenkou@xpinjection.com)*