

# Test-Driven Development with JAVA





Somkiat Puisungnoen

Somkiat Puisungnoen

Update Info 1 View Activity Log 10+ ...

Timeline About Friends 3,138 Photos More

When did you work at Opendream? X

... 22 Pending Items

Intro

Software Craftsmanship

Software Practitioner at สยามชัมนาภิกิจ พ.ศ. 2556

Agile Practitioner and Technical at SPRINT3r

Post Photo/Video Live Video Life Event

What's on your mind?

Public Post

Somkiat Puisungnoen 15 mins · Bangkok · ⚙️

Java and Bigdata



somkiat.cc

Page Messages Notifications 3 Insights Publishing Tools Settings Help ▾

somkiat.cc  
@somkiat.cc

Home Posts Videos Photos

Liked Following Share ... + Add a Button





# Problems

Mixed responsibilities  
Missing responsibilities  
Limited reuse potential  
Poor design



# Better design



# S.O.L.I.D principles



# S.O.L.I.D



**SOLID**

Software development is not a Jenga game.



# Single Responsibility Principle



## Single Responsibility Principle

Just because you *can* doesn't mean you *should*.



# Open-Closed Principle



## Open-Closed Principle

Open-chest surgery isn't needed when putting on a coat.



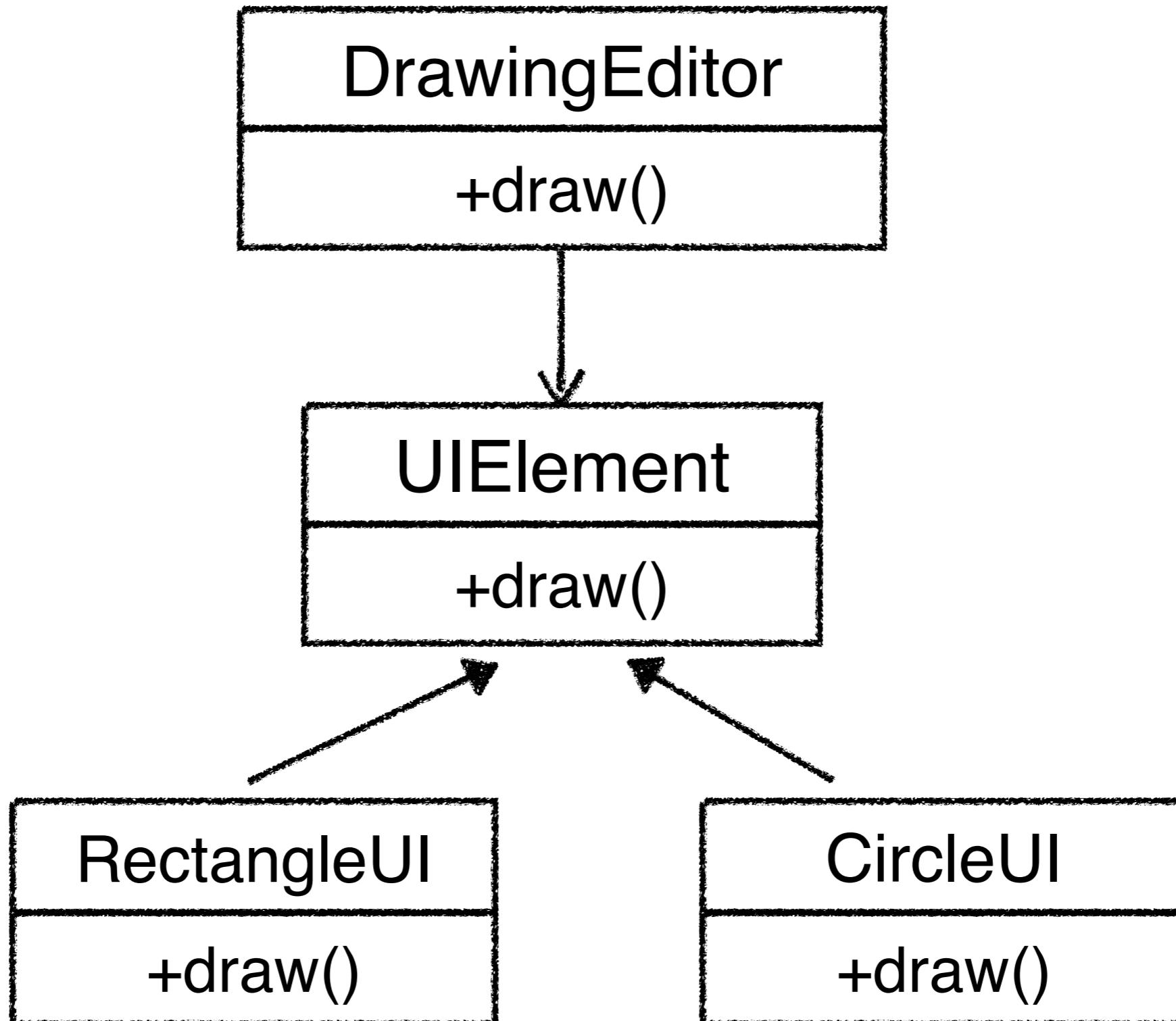
# Example

## DrawingEditor

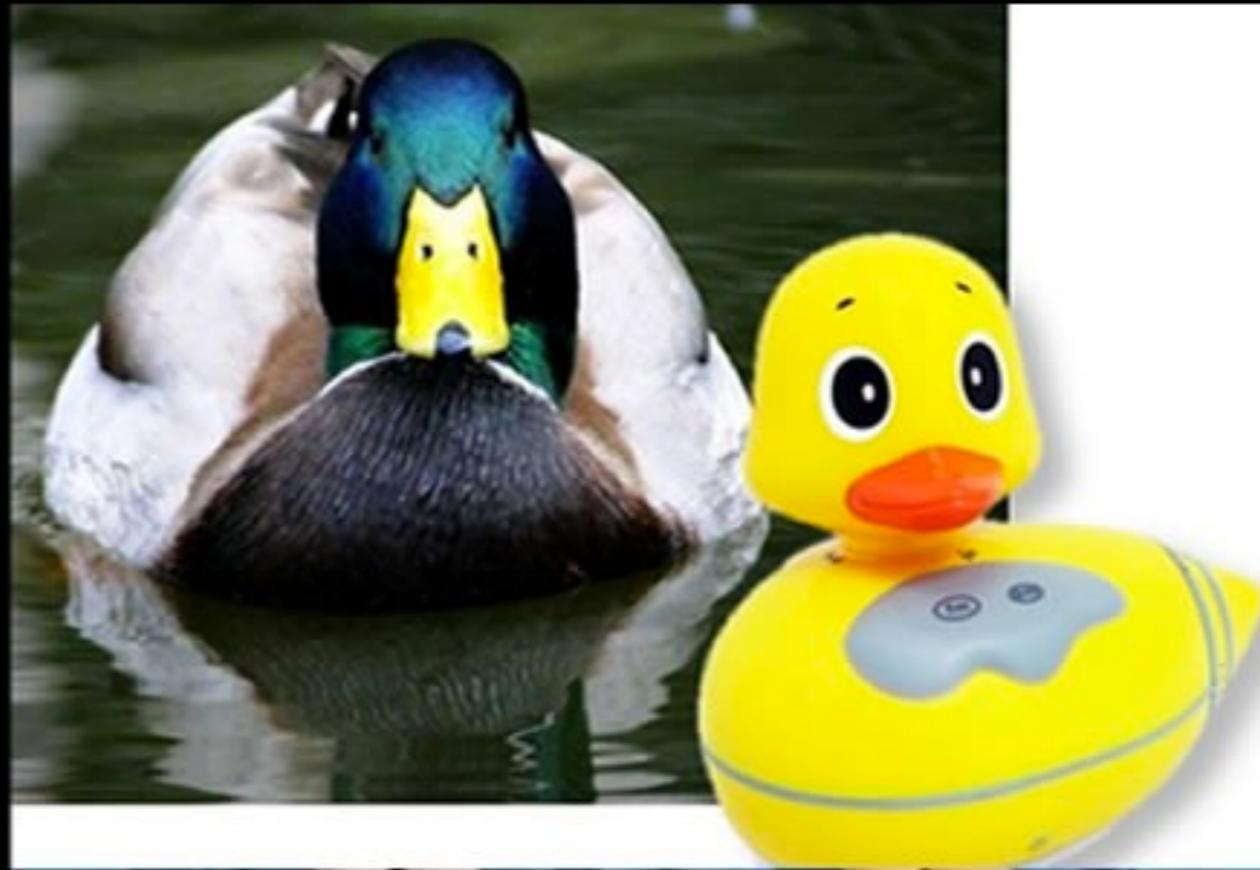
```
+draw()  
-drawCircle()  
-drawRectangle()
```



# Example



# Liskov Substitution Principle

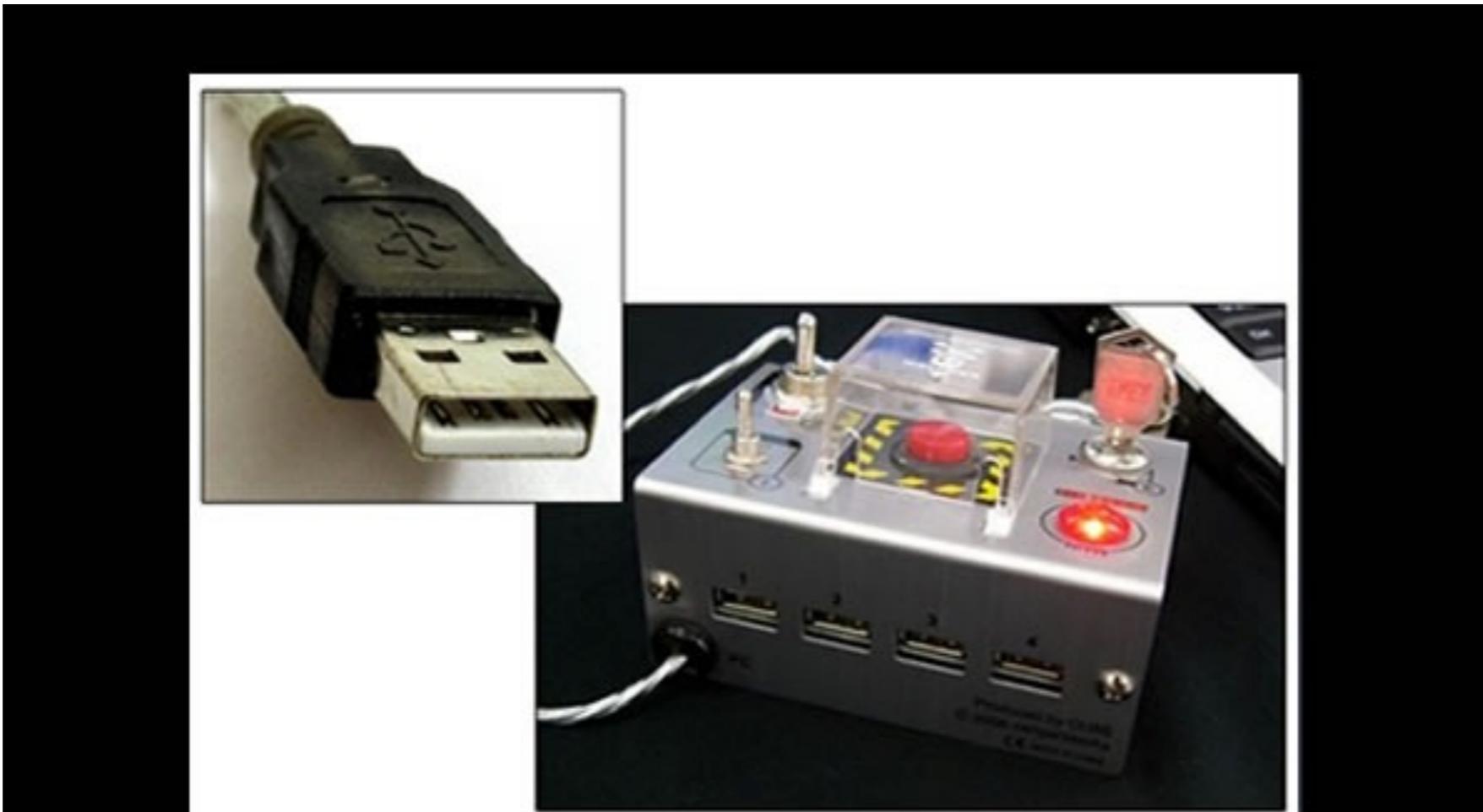


## Liskov Substitution Principle

If it looks like a duck and quacks like a duck but needs batteries,  
you probably have the wrong abstraction.



# Interface Segregation Principle

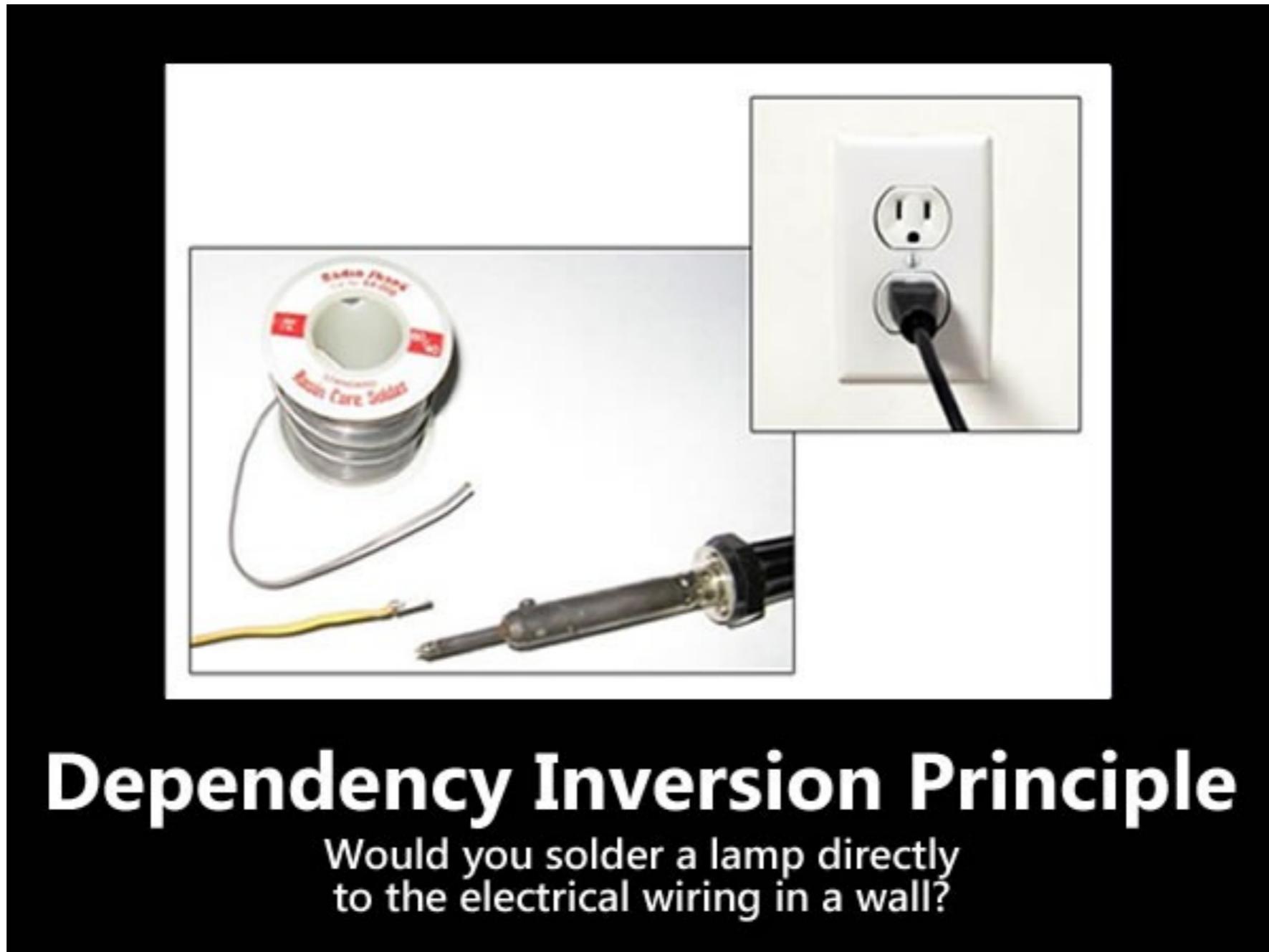


## Interface Segregation Principle

You want me to plug this in *where?*



# Dependency Inversion Principle



## Dependency Inversion Principle

Would you solder a lamp directly  
to the electrical wiring in a wall?



# Software Engineering Issues

Resource use  
Performance  
Security  
Deployment  
Support/Maintain



# Design process

CRC (Class Responsibility Collaborators)

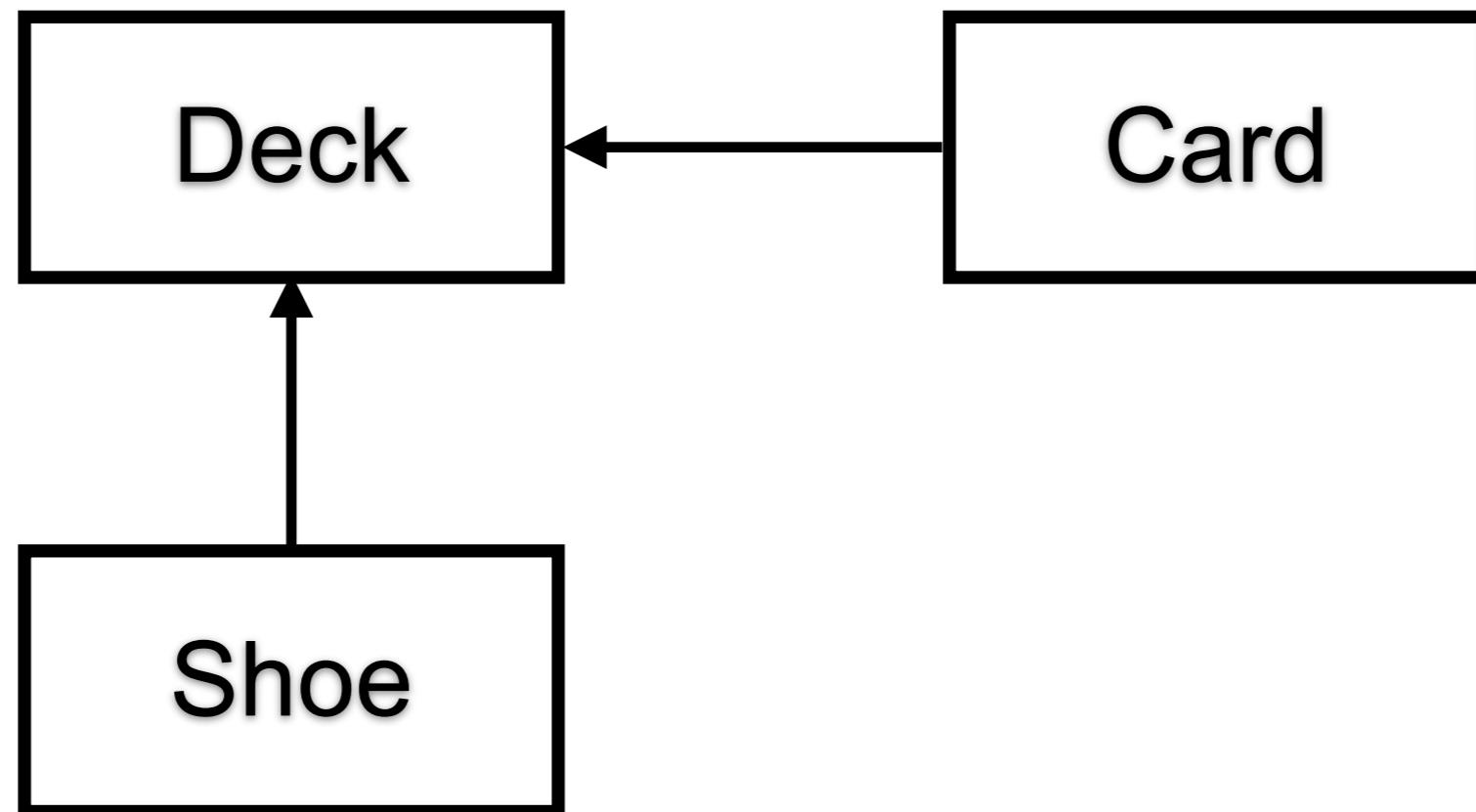


# Start with Nouns

Tangible things in the problem domain

Draw a picture

Define responsibility



# Scenario Testing

Play out a scenario or story

Examine card with the responsible class

Determine broken processing logic

Suggest changes



# Test as Design



# Workshop



<http://codingdojo.org/kata/Potter/>



TDD with Java

© 2017 - 2018 Siam Chamnankit Company Limited. All rights reserved.

# Definition of Done

Software is finished when it passes the acceptance test

Features are defined as test results



# Design is not easy



# Good code doesn't happen by accident



# GRASP

## General Responsibility Assignment Software Patterns

