



# Code Renovation

..what is it

# Precondition To Renovate

- Get yourself an armor -> TESTS
- Get yourself some tools -> IDE
- Sharpen your skills -> PRACTICE

## No Tests 🤪

- Changes should be minimal
- IDE driven when possible
- Only goal of ***Putting Tests In Place***

# The SEAM Model

*“A seam is a place where you can alter behaviour in your program without editing in that place.”*

- Enabling Point  
you can make the decision to use one (e.g. prod) or another (e.g. mock)
- Types of SEAMs
  - Pre-processing seams
  - Link seams
  - Object seams (the most common ones used for OO programming languages)

# For the sake of completion...

## Tests

- Approval Test
  - for human easy to evaluate test results
- Characterization Tests
  - Tests to enable refactoring
  - Ensure the system works the same
  - Quickly results in high test coverage
  - -> Golden Master
- Mutation Tests
  - mutations seeded into code

# Reconsidering Refactorings

- Nice and handy cheat sheet and refactoring methods
- What about the refactorings?
  - Some are easy understandable
  - Some seem easy understandable
  - Some are breaking the public interface
- Trying to categorize and reconcile the refactorings for myself

# Breaking Public Interface vs Non Breaking

- Adapt Parameter
- Break Out Method Object
- Encapsulate Global References
- Expose Static Method
- Primitivize Parameter
- Pull Up Feature
- Replace Global Reference With Getter
- Supersede Instance Variable
- Subclass And Override Method
- Extract Implementer
- Extract Interface
- Introduce Static Setter
- Push Down Dependency
- Extract And Override Call
- Extract And Override Factory Method
- Extract And Override Getter
- Introduce Instance Delegator
- Parametrize Constructor
- Parametrize Method

# Breaking Public Interface vs Non Breaking

- Adapt Parameter 🧨
- Break Out Method Object ✓
- Encapsulate Global References 🧨
- Expose Static Method ✓
- Primitivize Parameter ?
- Pull Up Feature ✓
- Replace Global Reference With Getter ✓
- Supersede Instance Variable ✓
- Subclass And Override Method ✓ 🏆
- Extract Implementer ✓ ?
- Extract Interface 🧨 ?

- Introduce Static Setter ✓
- Push Down Dependency ✓ ?
- Extract And Override Call ✓
- Extract And Override Factory Method ✓
- Extract And Override Getter ✓
- Introduce Instance Delegator ✓
- Parametrize Constructor ✓
- Parametrize Method ✓ ?

Breaking 🧨

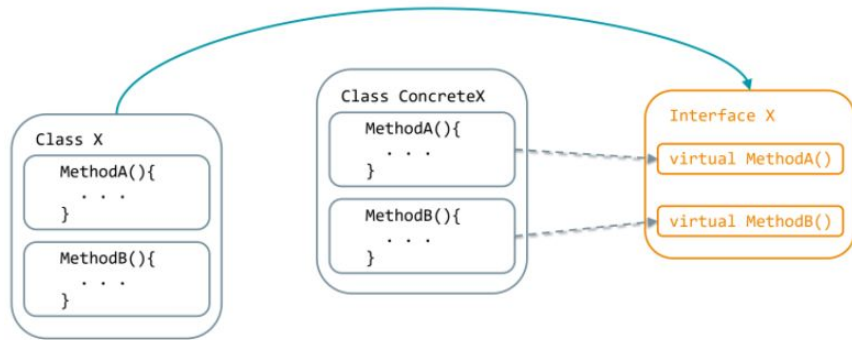
NonBreaking ✓

The Mother Of All Refactoring's

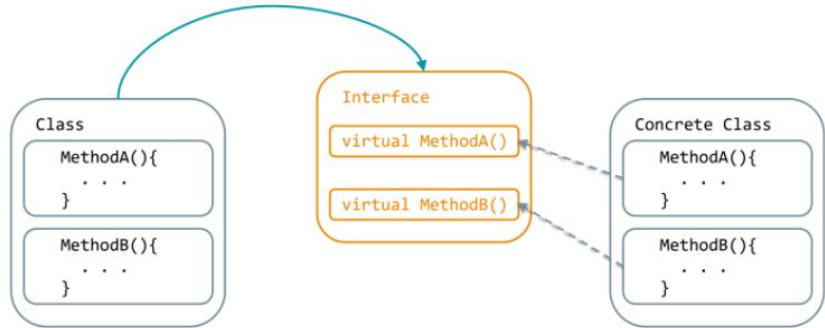


# Extract Implementer vs Extract Interface ?

## Extract Implementer



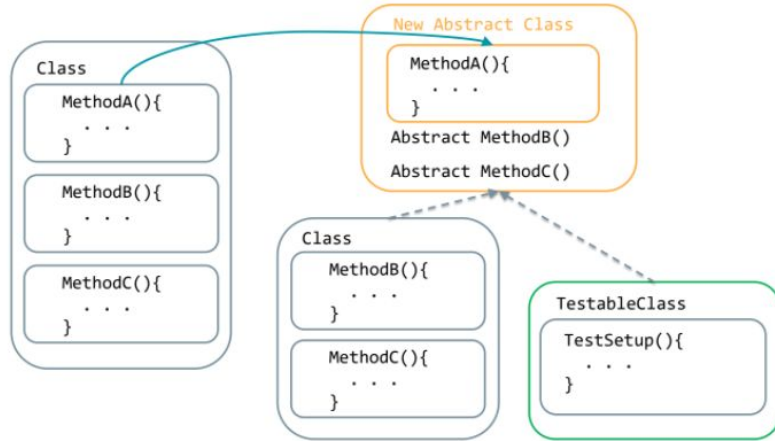
## Extract Interface



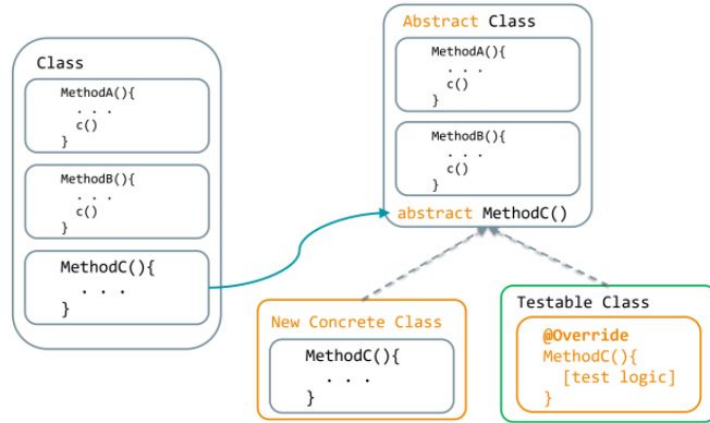


# Pull Up Feature vs Push Down Dependency?

## Pull Up Feature

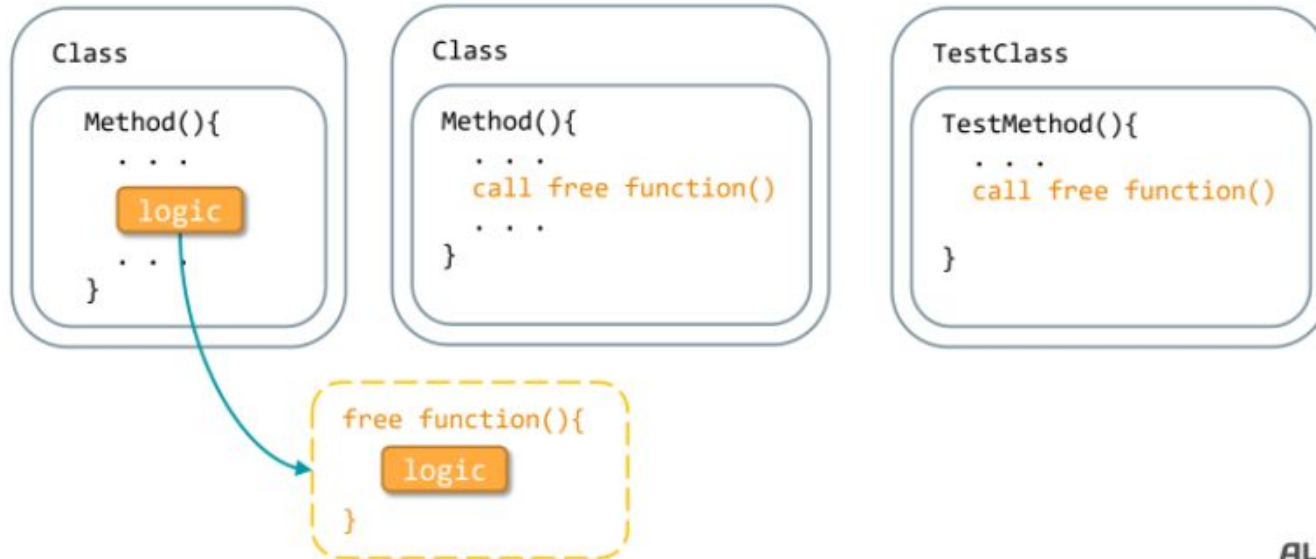


## Push Down Dependency



# What is “Primitivize Parameter” doing?

## Primitivize Parameter



# Thanks for your attention

.. Thx for the nice mob sessions to the fellow attendees

.. Thx for creating such a nice atmosphere @Alex and @Marco

**Contact:** [kliment.stojanov@css.ch](mailto:kliment.stojanov@css.ch)