Fundamentals:
The Don't Repeat Yourself Principle
Part 2

Steve Smith

http://pluralsight.com/

pluralsight
see what you can learn

Outline

Part 1

- **DRY Defined**
- **Demo: Repetition in Code**
- **Analysis**
- **Demos: Refactoring to apply DRY**

Part 2

- Refactoring for greater testability
- Adding Tests

- **Code Generation**
- **Repetition in Process**
- **Demo: Automation to apply DRY**
- **Summary and Related Fundamentals**

pluralsight
see what you can learn

Analysis

Part 1

- ~~Magic Strings/Values~~
- ~~Duplicate logic in several methods~~
- ~~Repeated if-then logic~~
- ~~Conditionals instead of polymorphism~~
- Repeated Execution Patterns (part 3)
- Lots of duplicate, probably copy-pasted, code (part 3)
- Only manual tests
- Static methods everywhere

No Tests

- **Only manual tests**

- **More and more expensive to verify all results are correct as new operations are added**

- **More and time wasted as long-running queries are added**

Static Methods

- **Tightly coupled**

- **Difficult to test**

- **Difficult to change behavior (violates OCP)**

- **Cannot use object oriented design techniques**
  - Inheritance
  - Polymorphism

Demo

Refactoring to Eliminate Static Cling and Add Tests

Summary

- **Repetition breeds errors and waste**

- **Abstract repetitive logic in code**

- **Related Fundamentals:**
  - Template Method Pattern
  - Command Pattern
  - Dependency Inversion Principle

- **Recommended Reading:**
  - The Pragmatic Programmer: From Journeyman to Master
    http://amzn.to/b2gJdK
  - 97 Things Every Programmer Should Know http://amzn.to/cAse1Y