Chapter 8. Makin' Objects

Money Operations

- Addition and Multiplication Examples
  - $5 + 10 CHF = $10 if rate is 2:1
  - $5 * 2 = $10
  - 5 CHF * 2 = 10 CHF
- Issues and Considerations
  - Make "amount" private
  - Dollar side effects?
  - Money rounding?
  - equals() method
  - hashCode() method
  - Equal null check
  - Equal object check
  - Dollar/Franc duplication

times() Method

- Two Implementations
  - Franc times(int multiplier) returns new Franc(amount * multiplier)
  - Dollar times(int multiplier) returns new Dollar(amount * multiplier)
- Step Toward Reconciliation
  - Both times() methods return Money type instead of subclass
  - Franc Money times(int multiplier) returns new Franc(...)
  - Dollar Money times(int multiplier) returns new Dollar(...)
- Elimination of Subclasses
  - Subclasses aren't doing enough to justify existence
  - Cannot eliminate in one big step for effective TDD demonstration
  - Aim to reduce direct references to subclasses

Factory Method Introduction

- Purpose
  - Reduce direct references to Dollar and Franc subclasses
  - Create Dollar instance via factory method: Money.dollar(int amount)
- Test Example
  - Money five = Money.dollar(5)
  - assertEquals(new Dollar(10), five.times(2))
- Implementation Details
  - static Money dollar(int amount) returns new Dollar(amount)
  - Change factory method declaration to return Money
  - Declare Money as abstract class with abstract Money times(int multiplier)
- Benefits
  - Tests depend on Money, not on Dollar subclass directly
  - Allows changes in inheritance without affecting model code

Test Code Adjustments

- Multiplication Tests
  - Use factory methods in tests
  - assertEquals(Money.dollar(10), five.times(2))
  - assertEquals(Money.dollar(15), five.times(3))
- Equality Tests
  - assertTrue(Money.dollar(5).equals(Money.dollar(5)))
  - assertFalse(Money.dollar(5).equals(Money.dollar(6)))
  - assertTrue(Money.franc(5).equals(Money.franc(5)))
  - assertFalse(Money.franc(5).equals(Money.franc(6)))
  - assertFalse(Money.franc(5).equals(Money.dollar(5)))
- Franc Factory Method
  - static Money franc(int amount) returns new Franc(amount)
  - Test Franc multiplication with Money.franc(5)

Refactoring Observations

- Test Redundancy
  - testFrancMultiplication() tests logic identical to Dollar multiplication test
  - Deleting testFrancMultiplication() may reduce confidence
  - Suspicious redundancy but test retained for now
- Achievements So Far
  - Reconciled signatures of times() in subclasses
  - Moved times() declaration to common superclass
  - Decoupled tests from concrete subclasses via factory methods
  - Recognized some tests will become redundant as subclasses disappear
  - No action taken yet on redundant tests