

TDD by Example

Chapter 10: Interesting Times

Arithmetic and Currency Conversion

- \$5 + 10 CHF = \$10 if rate is 2:1
- \$5 * 2 = \$10
- 5 CHF * 2 = 10 CHF

Code Refactoring and Implementation

- Making "amount" Private
- Handling Dollar Side Effects and Money Rounding
- equals() and hashCode() Methods
 - Common equals implementation
 - Comparing Francs to Dollars
 - Equal null and Equal object considerations
- Dollar/Franc Duplication
 - Two implementations of times() method in Franc and Dollar classes are close but not identical
- Factory Methods and Inlining
 - Initially use factory methods Money.franc() and Money.dollar()
 - Inline factory methods to create new Franc and Dollar instances with currency codes
- Currency Instance Variable Optimization
 - Use currency instance variable directly in Franc and Dollar times() methods
- Experimenting with Returning Money Instead of Subclasses
 - Change Franc.times() to return Money instead of Franc
 - Compiler requires Money to be concrete class
 - Initial tests fail due to equals() method comparing classes instead of currencies
- Improving Debugging with toString()
 - Implement toString() to show amount and currency for better error messages
 - toString() written without a test due to low risk and debug purpose
- Fixing equals() Method
 - Original equals() uses getClass() comparison which fails for Money and Franc equality
 - Change equals() to compare amount and currency instead of class type
 - Write test testDifferentClassEquality() that checks Money equals Franc with same amount and currency
- Unifying times() Method
 - After fixing equals(), Franc.times() and Dollar.times() return Money instances and pass tests
 - Push common times() implementation up to Money class
- Eliminating Subclasses
 - With common times() method in Money, ready to remove Dollar and Franc subclasses

Summary of Refactoring Steps

- Reconciled times() methods by inlining and variable replacement
- Wrote toString() without a test to aid debugging
- Performed experiment returning Money instead of subclass and used tests to validate
- Backed out failing changes and added tests to fix equals() method
- Made tests pass and unified times() method in Money class