



# Testing Spring

---

Beginner to Guru

TDD 14 - Change



## TDD To-Do List

- $\$5 + 10 \text{ CHF} = \$10$  (with rate of 2:1)
- $\$5 + \$5 = \$10$
- Return Money from  $\$5 + \$5$
- **Reduce Money with Conversion**
- ~~Bank.reduce(Money)~~
- ~~$\$5 * 2 = \$10$~~
- Make "amount" private
- ~~Dollar side effects?~~
- Money Rounding?
- equals()
- hashCode()
- Equal Null Equal Object
- ~~$5 \text{ CHF} * 2 = 10 \text{ CHF}$~~
- ~~Dollar/Franc Duplication~~
- ~~Common equals()~~
- ~~Common times()~~
- ~~Compare Francs With Dollars~~
- ~~Currency~~
- ~~Delete testFrancMultiplication?~~





## Tasks

- Create object 'Pair' to represent currency pairs
  - Will need Equals and Hashcode since object will be used as key
- Add rate to Bank - From Currency, To Currency, conversion rate
  - Examples in book use HashTable - circa Java 1.2
  - We will implement using HashMap (ThreadSafety not needed in our example)
    - If ThreadSafety was needed, we could use ConcurrentHashMap
  - Need method to return rate for given from / to currency codes
  - Need Add rate method

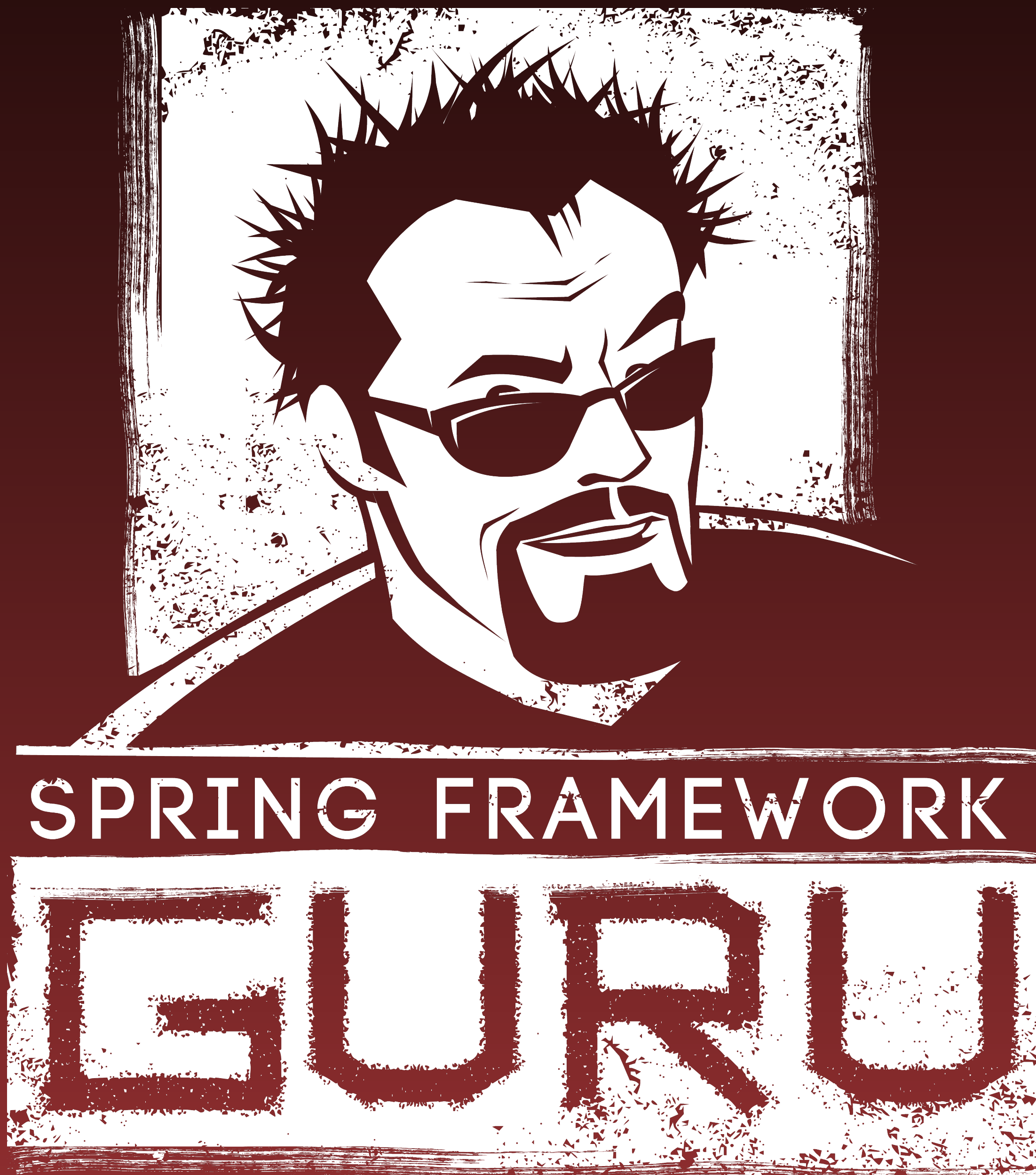




## Tasks

- Change Expression.reduce method to accept bank
  - Change Money / Sum reduce methods to accept Bank parameter in reduce method
  - Change Money to use rate in reduce method









## TDD To-Do List

- $\$5 + 10 \text{ CHF} = \$10$  (with rate of 2:1)
- ~~$\$5 + \$5 = \$10$~~
- Return Money from  $\$5 + \$5$
- ~~Reduce Money with Conversion~~
- ~~Bank.reduce(Money)~~
- ~~$\$5 * 2 = \$10$~~
- Make "amount" private
- ~~Dollar side effects?~~
- Money Rounding?
- equals()
- hashCode()
- Equal Null Equal Object
- ~~$5 \text{ CHF} * 2 = 10 \text{ CHF}$~~
- ~~Dollar/Franc Duplication~~
- ~~Common equals()~~
- ~~Common times()~~
- ~~Compare Francs With Dollars~~
- ~~Currency~~
- ~~Delete testFrancMultiplication?~~



