**Money Example**

*Make to – do list first.* Development Cycle:

* Write the test.
* Test fails (getting red).
* Change code a little to pass the test (gettin green).
* Refactor the code.

**Degenerate Objects**

Don’t want to test object is changed. (For avoiding side effects)

Two strategies for getting green:

* Fake it
* Real implementation

**Equality For All**

“Value Objects Pattern” (avoid aliasing)

Equals() method

**Privacy**

Making “amount” private

**Franc-ly Speaking**

-Couldn’t tackle big test, so we invented a small test that represented progress.

To do list

* $5 + 10 CHF = if CHF:USD is 2:1
* ~~$5\*2 = 10$~~
* ~~Make “amount” private (due to privacy)~~
* ~~Dollar side effects? (value object pattern)~~
* Money rounding?
* ~~Equals()~~
* HashCode()
* Equal null
* Equal Object
* ~~5 CHF \* 2 = 10 CHF~~
* Dollar/Franc duplication
* Common Equals