

CSE 8B—Intro to CS: Java

Winter, 2006

January 12 (Day 2)
Test-Driven Development

Instructor: Neil Rhodes

Problems with unit tests

Hard to keep them up to date

Hard to be motivated to write them in the first place

2

The basics of test-driven development

1. Decide the next small step to accomplish
 - Incremental development: small steps
2. Write a test that'll test that accomplishment is successful
 - How will you know when it's working if you can't test it?
3. Compile
 - compile will likely fail
4. Write enough code that the test complies
 - test will likely fail
5. Write just enough code to make your test pass
 - Extreme Programming: You aren't gonna Need it (YagNi)
6. If you need more testing of the accomplishment, go back to step 2.
7. Refactor
 - Code to be tested, test code, or both

Lather, rinse, repeat.

3

Refactoring

Definition:

- Modifying code without changing its external behavior

What it is:

- Cleaning up the code
- Removing repetition (capturing commonality)
 - Once and Only Once (or, Don't Repeat Yourself)
- Removing dead code
- Improving the understandability

What isn't:

- fixing bugs
- adding functionality

Examples

- Renaming a variable, method, or class
- Breaking a large method into smaller methods
- Encapsulating an instance variable

4

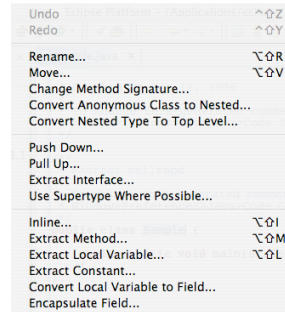
IDE support for refactoring

Eclipse, a Java IDE, has built-in support for refactoring

- Others do as well

Computer can automate

- Changing method name means changing all callers, for example



5

Advantages of test-driven development

You've always got something working

Unit tests are always there

You can refactor with impunity

You don't waste time on unneeded code

6

Test-Driven Development Demo

Application to create

Bowling scores

- American ten-pin bowling
- Consists of ten frames
 - Each frame bowler gets up to two rolls
 - Score for frame is:
 - If not all pins knocked down: total knocked down
 - If all pins knocked down in two rolls (spare): 10 + number pins knocked down on next roll
 - If all pins knocked down in one roll (strike): 10 + number pins knocked down on next two rolls
 - Max score per frame = 30
 - Min score per frame = 0
 - If spare (strike) in last frame, bowler bowls one (two) extra rolls.

We want to compute score for entire game

- Don't care about: validity checking for number of rolls or pins knocked down
- Don't care about scores for each frame, just the entire game

8