

Programming By Intention/ Intro to JUnit



Admin

- ▶ Astels, p. 50 – “The test in the section titled *Programming by Intention...*” should read “The test in the section titled *Do the Simplest Thing...*”
- ▶ Quizzes
 - Read the question
 - Handling quiz opt-outs
 - Two items != one item worded two ways
 - Refactoring key – Not changing functionality
 - Shotgun Surgery = having to make changes in

Programming By Intention

- ▶ Methods for keeping code as good as possible
- ▶ Guidelines we'll follow for code we write in this class

Naming

- ▶ Nouns for class names/variables (Person, Order, Item)
- ▶ Adjectives/generic nouns for interfaces (Testable, Database, Iterator)
- ▶ Verbs for method names (Save, calculateTotalInterest)
- ▶ getX(), setX(), isX(), hasX()
- ▶ Leave out redundant info

Simplicity

- ▶ Make code as simple as possible to do what's needed
- ▶ Use refactorings to keep things simple
- ▶ Use test-first programming

Assumptions

- ▶ Make assumptions while you write your test
- ▶ Makes you think about your code from the standpoint of using the code, which means your code plays better with others
- ▶ Don't worry about tracking the assumptions you make, let the compiler tell you

Comments

- ▶ Not all comments are bad
- ▶ Examples of good comments
 - Todo items – incomplete code
 - Future refactorings (either by you or someone else)
 - Use of uncommon acronym
 - Attribution
 - Changes made for specific reason (e.g., performance tuning)
 - Brief class overview

JUnit

- ▶ Developed by Kent Beck and Erich Gamma
- ▶ Variant of sUnit (Smalltalk)
- ▶ Current version = 3.8.1
- ▶ Included in Eclipse and many other IDEs

Structure

▶ Test Cases

- Extend JUnit TestCase class
- Contain methods which perform actual tests
- Main piece for TDD

▶ Test Suites

- Contain Test Cases and/or other Test Suites
- Default Test Suite contains all tests
- Generally explicitly defined to allow running subsets of tests

Asserts

- ▶ Actual checks performed by test methods
- ▶ Two forms – use one with extra string param
- ▶ Fail()
- ▶ assertTrue/assertFalse
- ▶ assertNull/assertNotNull
- ▶ assertEquals/assertNotSame
- ▶ assertEquals

Eclipse Demo

