Software Testing and Debugging

Testing

- As you program more, your functions will get more complicated
- How can we ensure they work?

Testing

- Run your code a bunch
  - Is this enough?

- Would like a more rigorous system

Testing

- Want something that:
  - Will tell us when our code breaks
  - Tells us early if something goes wrong
  - Should test our code at an *atomic* level
- How would we accomplish this with our current tools?

Testing

- Two Programs?

- One to run all our of tests
  - Print the output if something isn’t right
- One to actually run our program
  - Shouldn’t interact with our test code

Testing

- This would get tedious very quickly
  - Does not scale as your team grows

- We should bring in a more rigorous system to solve this problem at scale

Aside: External Libraries

- Compile to a “.jar” file
- Let someone else use your code
- Your code provides a set of public:
  - Interfaces
  - Classes
  - Functions
  - Annotations*

Aside: External Libraries

- Compile with added jar files
- Must match directory structure

- VSCode → lib folder
- IntelliJ → File > Project Structure > Modules > [Your Module] > Dependencies

JUnit Library

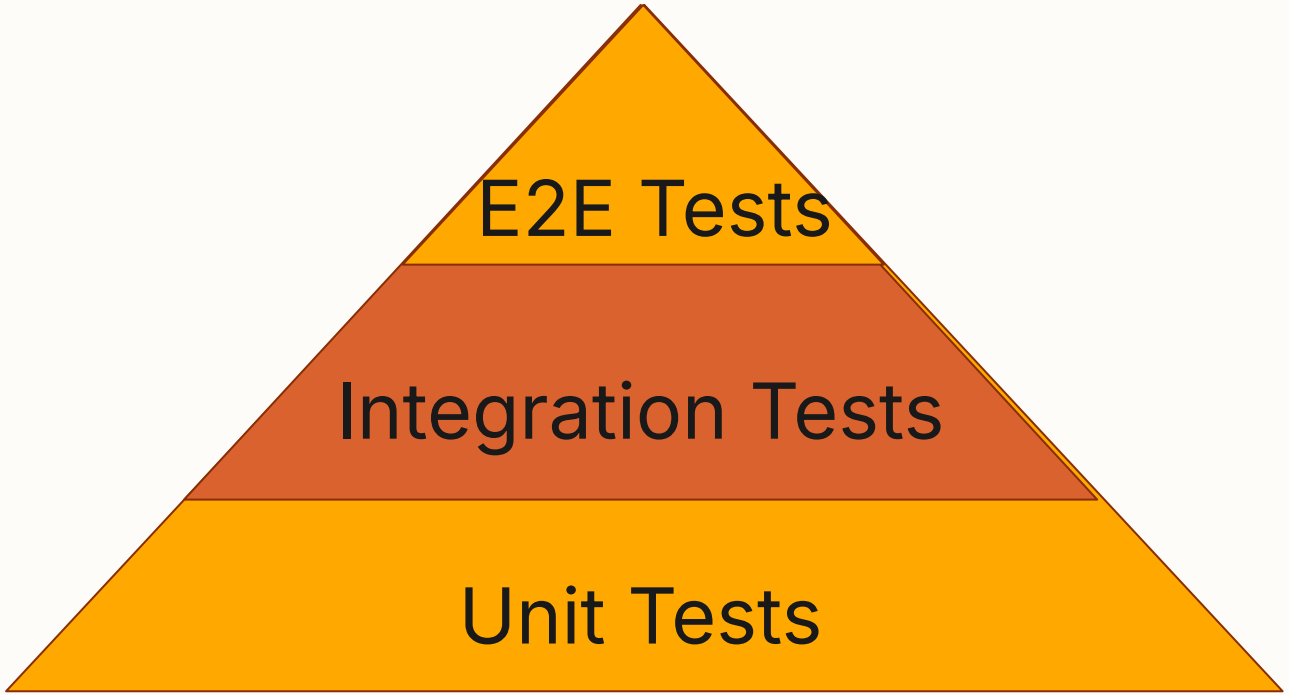
- JUnit is the de facto standard for testing
- File Conventions
  - Dog.java
  - DogTest.java

What does it mean to “test” a function

- Atomic level testing
- Smallest “unit” of code

- “Unit Testing” is the SE buzzword
- There are other types of testing

Aside: Types of Tests



How does JUnit work?

- Provides an @Test annotation
- Your editor will let you run just the test portion of the code

- Can run:
  - Individual Test
  - All tests in the class
  - All tests in the project

Actually writing test

- @Test
- public void [description]() { ... }

- Overly descriptive name
- Separate file just for test

Actually writing test

- Let’s write some test for some basic code
- Calculator.java

- We should put out code into CalculatorTests.java
- Can test for Exceptions too!

Test Driven Development

- Why am I doing this?
- How do problems look when they are given?

Test Driven Development

- Why am I doing this?
- How do problems look when they are given?

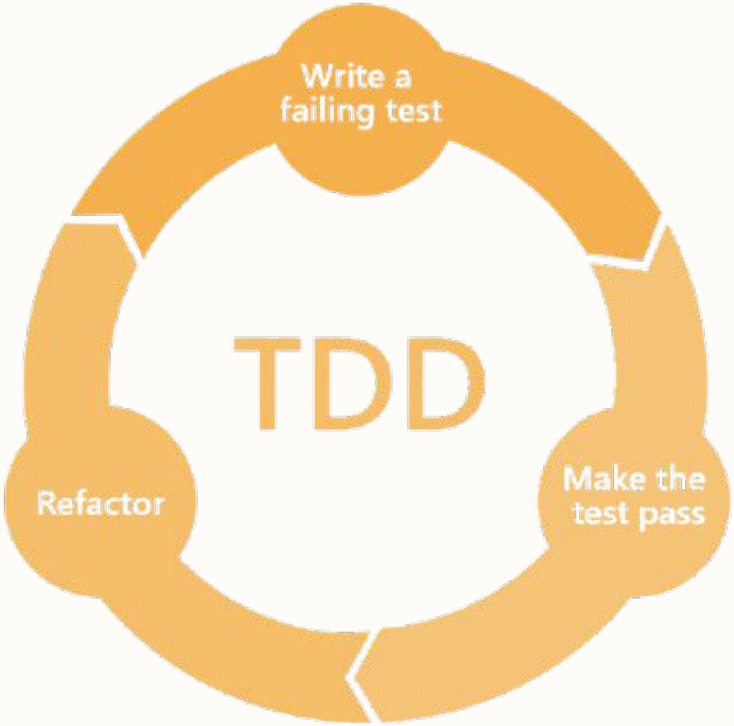
Test Driven Development

- This looks a lot like how tests work
- Why did we do that order of things?

Test Driven Development

- Code → Test?
- Test → Code?

- Iterate on this idea continuously



Test Driven Development

- Commonly seen in new projects
- Your boss will love this
- Not always the answer

Using TDD

- We have a function that does *a lot* of stuff
- processString(s) should:
  - Convert to lowercase
  - Remove all letter “z”s

Using TDD

- We know how processString() should work
- Write the tests first
- Then code it after
- Let’s do it

Using TDD

- New requirement just dropped
- Must pad string with “_ _ … _ _”
  - Fix code?
  - Fix tests?

Aside: Stateless tests

- Notice we only tested static methods
- How would we test non-static methods?
- We would need instances of everything our code relies on
- “Mocking” is how this is done IRL