CSSE 220 Day 3

Unit Tests, API Documentation, and Object References

Check out *JavadocsAndUnitTesting* from SVN

Questions?

Java Documentation

- API Documentation, Docs in Eclipse, Writing your own Docs

Java API Documentation

- What's an API?
  - Application Programming Interface
- The Java API on-line
  - Google for: java api documentation 6
  - Or go to: http://java.sun.com/javase/6/docs/api/
- Find the String class documentation:
  - Click java.lang in the top-left pane
  - Then click String in the bottom-left pane

Q1,2

Java Documentation in Eclipse

- Setting up Java API documentation in Eclipse
  - Should be done already, but if the next steps don't work for you, we'll fix that
- Using the API documentation in Eclipse
  - Hover text
  - Open external documentation (Shift-F2)

Writing Javadocs

- Written in special comments: /** … */
- Can come before:
  - Class declarations
  - Field declarations
  - Method declarations
- Eclipse is your friend!
  - It will generate javadoc comments automatically
  - It will notice when you start typing a javadoc comment

Example Javadoc for a Class

Description of class

```
/**
 * This class demonstrates unit testing
 * and asks you to use the Java API
 * documentation to find methods to solve
 * problems using Strings.
 *
 * @author Curt Clifton.
 * Created Sep 9, 2008.
 */
public class MoreWordGames { … }
```

@author *Tag* followed by author name and date

Example Javadoc for a Method

```
/**
 * Converts the original string to a
 * string representing shouting.
 *
 * @param input the original string
 * @return input in ALL UPPER CASE
 */
static String shout(String input) {
    return input.toUpperCase();
}
```

Description of method, usually starts with a verb.

@param tag followed by parameter name and (optional) description. Repeat for each parameter.

@result tag followed by description of result. Omit for void methods.

Exercise

- Add javadoc comments to MoreWordGames

Javadocs: Key Points

- Don't try to memorize the Java libraries
  - Nearly 9000 classes and packages!
  - You'll learn them over time

- Get in the habit of writing the javadocs **before** implementing the methods
  - It will help you think before doing, a vital software development skill
  - This is called programming with *documented stubs*
  - I'll try to model this. If I don't, call me on it!

Writing Code to Test Your Code

- Test-driven Development, unit testing and JUnit

Unit Testing

- Writing code to test other code
- Focused on testing individual pieces of code (units) in isolation
  - Individual methods
  - Individual objects

- Why would software engineers do unit testing?

Q3,4

Unit Testing With JUnit

- JUnit is a unit testing *framework*
  - A framework is a collection of classes to be used in another program
  - Does much of the work for us!
- JUnit was written by
  - Erich Gamma
  - Kent Beck
- Open-source software
- Now used by **millions** of Java developers

Q5

JUnit Example

- `MoveTester` in Big Java shows how to write tests in plain Java
- Look at `JUnitMoveTester` in today's repository
  - Shows the same test in JUnit
  - Let's look at the comments and code together…

Interesting Tests

Important Slide: Use this as a reference!

- Test “boundary conditions”
  - Intersection points: −40℃ == −40℉
  - Zero values: 0℃ == 32℉
  - Empty strings
- Test known values: 100℃ == 212℉
  - But not too many
- Tests things that might go wrong
  - Unexpected user input: “zero” when 0 is expected
- Vary things that are “important” to the code
  - String length if method depends on it
  - String case if method manipulates that

Exercise

- Walk through creating unit tests for shout in MoreWordGames

  Test whisper and holleWerld

Object References

- Differences between primitive types and object types in Java

What Do Variables Really Store?

- Variables of number type store *values*
- Variables of class type store *references*
  - A reference is like a pointer in C, except
    - Java keeps us from screwing up
    - No & and * to worry about
      (and the people say, "Amen")
- Consider:

```
1. int x = 10;
2. int y = 20;
3. Rectangle box = new Rectangle(x,y,5,5);
```

Q6

Assignment Copies Values

- Actual value for number types
- **Reference** value for object types
  - The actual **object is not copied**
  - The **reference value** ("the pointer") **is copied**
- Consider:

```
1. int x = 10;
2. int y = x;
3. y = 20;

4. Rectangle box = new Rectangle(5,6,7,8);
5. Rectangle box2 = box;
6. box2.translate(4,4);
```

Q7,8

Exercise

- Begin the Written Exercise from Homework 3

Q9,10