

Quiz 1

- In class on Tuesday (April 14)
 - Will start promptly at 9:30am
 - Will take about 20 minutes
- Review learning goals (from lecture)
- Review lectures 1-4, PSA 1
- You should be able to:
 - Read and trace code
 - Write code

Learning Goals (1)

- LG1: Describe the effect of Java's "call-by-value" parameter passing mechanism
- LG2: Trace code to recognize that primitive type variables (ints, doubles, etc.) that are passed as parameters to methods and modified locally in those methods DO NOT have their values changed in the calling method scope.
 - Since the value is copied into the local method (but not BACK to the calling method).
- LG3: Trace code to recognize that reference type variables (including objects and arrays of anything) that are passed as parameters to methods and modified locally in those methods DO have their values changed in the calling method scope.
 - Since the REFERENCE to the object/array is copied into the local method and anything that changes the data REFERENCED by that object/array in the local method is modifying the SAME COPY as is in scope of the calling method.

Learning Goals (2)

- LG4: Trace code which news a new object “overwriting” the reference to an object passed as a parameter (with no effect in the calling method scope since the reference is not copied BACK to the calling method).
- LG5: Use common data protection conventions when constructing a new object that has an array instance variable (create a copy of the array for the object, not just a reference to it).
- LG6: Explain why we’d create and use a static class variable.
- LG7: Be able to find information (including Javadocs for Java 6) on the Wrapper classes so you can use them in your PSA.
- LG8: Draw a unix directory structure diagram employing the terms: root directory, file suffix, root, leaf, parent, child.

Learning Goals (3)

- LG9: Read and write code that uses `BufferedReader` to read text from a file and catch exceptions appropriately with a simple try and catch block.
- LG10: Write code to iteratively read through a file until the end of file is reached (or some other sentinel value).
 - Sentinel value (not defined in book) is a value that means “STOP HERE”
- LG11: Read and write code using `BufferedWriter` to write to a file and catch exceptions appropriately.
- LG12: Be able to use Javadocs to determine whether a specific method call needs to be included in a try block, because it throws an exception.

Parameter Passing

- Review examples from lecture
- CALL BY VALUE (IN)
 - Value is copied in to a method, but is not copied back out
 - What that value is depends on variable type (primitive / reference)
- Passing primitives
 - What gets copied into called method?
 - If modified in called method, does value of primitive get changed in calling method scope?
- Passing reference types (objects, arrays)
 - What gets copied into called method?
 - If modified in called method, does value of referenced object/array get changed in calling method scope?

File Reading

- BufferedReader API
 - <http://java.sun.com/j2se/1.5.0/docs/api/java/io/BufferedReader.html>
- String readLine()
 - Reads a line of text.
 - Returns the string that was read.
- Example

```
BufferedReader reader =  
    new BufferedReader(new FileReader("input.txt"));  
String line = reader.readLine();
```

String Manipulation (1)

- String API
 - <http://java.sun.com/j2se/1.5.0/docs/api/java/lang/String.html>
- Some useful String methods
 - equals
 - substring
 - indexOf
 - replace
 - toUpperCase / toLowerCase
 - split
- Methods may be overloaded
 - Multiple versions of method with same name, different signature

String Manipulation (2)

- `boolean equals(Object anObject)`
 - Compares this string to the specified object.
 - Returns true if equal, false if not equal.
- Example
 - `String string1 = "string";`
 - `String string2 = "string";`
 - `String string3 = "differentString";`
 - What is value of `string1.equals(string2)` ?
 - What is value of `string1.equals(string3)` ?

String Manipulation (3)

- `int indexOf(String str)`
 - Returns the index within this string of the first occurrence of the specified substring. If substring does not occur within this string, returns -1.
- Example
 - `String string1 = "This is a string";`
 - `String string2 = "This";`
 - `String string3 = "Not in string1";`
 - What is value of `string1.indexOf(string2)` ?
 - What is value of `string1.indexOf(string3)` ?

File Reading and String Manipulation

- Examples

- Read until you see a line containing JUST “Bob”

```
BufferedReader reader =
```

```
    new BufferedReader (new FileReader("input.txt"));
```

```
String line;
```

```
while (_____) {} // line doesn't contain JUST “Bob”
```

- Read until you see a line containing “Bob” ANYWHERE in it

```
BufferedReader reader =
```

```
    new BufferedReader (new FileReader("input.txt"));
```

```
String line;
```

```
while (_____) {} // line doesn't contain “Bob” ANYWHERE
```

PSA 2

- PSA 2 is on the course website
- Due Thursday, April 23 at 11:59pm
- Read and process sensor data about waves.
 - Data collected at SIO buoy stations in Oceanside and Torrey Pines
- Classes
 - SensorData – corresponds to a line of sensor data
 - SensorReader – read data from webpage and parse lines into SensorData objects
 - DataProcessor – process lists of SensorData objects