SE2AA4 Winter 2007 - Lab 3

TA: Clare So (socm@mcmaster.ca)

March 16, 2007. 10:30–11:30

0 Before you begin

Good morning! You are about to practice what you have learned during the lectures. During this lab, you will

- Implement a binary tree ADT in Java
- Learn to use interfaces from the Java API
- Understand how interfaces help us to develop a general solution
- Practice the **information hiding** technique

You need a few (incomplete) Java classes for this exercise. Please download the following files under "TA-files" of the course website:

- `TreeNode.java`
- `BetterBinaryTree.class`
  (Yes, you only need the **class** file!)
- `TestBinaryTree.java`

1 First Task: Finish `TreeNode.java`

You are to implement a **node** of a binary tree. Remember that a node in a binary tree stores data and its two children. The children of a node are nodes, too.

`TreeNode.java` is an incomplete implementation of a node in our binary tree. You are given the all three attributes of a node: `contents`, `myRight` and `myLeft`. Don't change or add any attributes. I have implemented some of the methods. Please implement the following:

- Constructor: `TreeNode`
- Selectors: `getLeft`, `getRight`
- Mutators: `setLeft`, `setRight`

1.1 Help! What does `Comparable` mean?

As you have noticed, the `TreeNode` class implements `Comparable`. `Comparable` is an interface from the standard Java API. This interface ensures all objects generated can be compared. All classes that implements this interface must have a `compareTo` method. I have implemented `compareTo` of the `TreeNode` class. This method returns a negative integer, zero, or a positive integer as this object is less than, equal to, or greater than the specified object.

Some classes in the Java API such as `Integer` (captial "I"!), `Float`, and `Date` implement `Comparable`. You can also write your own class that implements the `Comparable` interface.

You can read more about `Comparable` interface from the Java API documentation from Sun's Java website (go to `http://java.sun.com` → APIs → J2SE 1.5.0).

2 (Task 2: Finish `BinaryTree.java`)

I have provided you the **class** file of BinaryTree. Sorry, you can't have a look in the implementation. All I can tell you is the interface of this class. This class contains three public methods

- `public BinaryTree()` (Constructor)
- `public int depth()`
- `public boolean insert(Comparable contents)`
- `public String toString()`
  (Visit each node by in-order traversal)

(I will post the source code of this class **after** the lab.)

3 Task 3: Create your own class

Well, this is your turn to write a class that implements `comparable`. All public methods you need are the contructors) and `compareTo` method. Here are some objects you may be interested in creating:

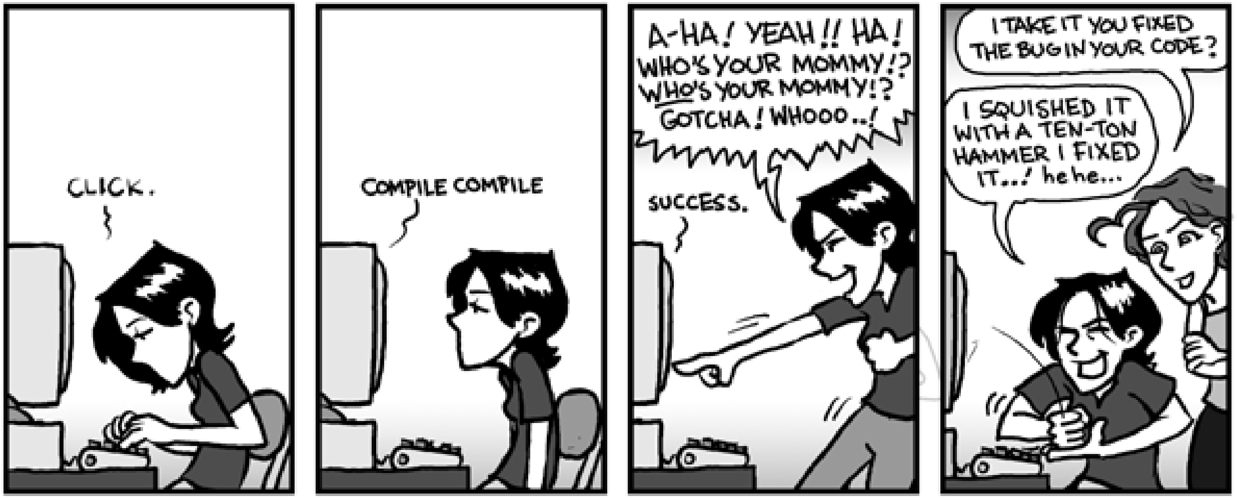
- Vectors
- Sets
- Favourite movies, books, games etc.
- Star wars characters

4 Task 4: Testing

A sample main program is found in `TestBetterBinaryTree.java`. This main program demonstrate how you can insert any `comparable` objects into the tree, get the depth and print out the contents of the tree.

5 Questions

1. Do you really need the source code `BetterBinaryTree.java` to develop your test?
2. Name some interfaces from the Java API. What do they do?
3. What can we put into our tree?


CLICK.
COMPILE COMPILE
A-HA! YEAH!! HA! WHO'S YOUR MOMMY!? WHO'S YOUR MOMMY!? GOTCHA! WHOOO..!
SUCCESS.
I TAKE IT YOU FIXED THE BUG IN YOUR CODE?
I SQUISHED IT WITH A TEN-TON HAMMER I FIXED IT...! hehe...

phd.stanford.edu/

(PhD comics. 4/28/2000)