APS101 lab 6 – week 8

This document contains the instructions for the week 8 APS101 lab.

1 Objectives

1. Practice loops in the context of reading and writing files.
2. Learn one way to handle `IOException`s.
3. Learn how to use a `JTextArea`.
4. Learn how to prompt a user for input using a GUI.

2 Starting up

Sit down with your partner. The rest of these instructions call you two `s1` and `s2`. Pick which one is which.

`s1` should log in and start up DrJava, and be the first driver.

3 File copying

This opens a file for reading:

```
BufferedReader br = new BufferedReader(new FileReader("file.txt"));
```

To read a line of the file, use `br.readLine()`. When there is no more input, `br.readLine()` returns `null`.

The following code opens a file for writing; `PrintStream` has methods `print` and `println`, just like `System.out`:

```
PrintStream p = new PrintStream(new FileOutputStream("file.txt"));
```

Most of these I/O (input/output) operations might throw an `IOException`. You must add `throws IOException` to any method header that uses any of this code.

Write a class `FileCopier` with one `static` method `copy` that takes the name of the input file and the name of the output file, and copies the contents of the input file to the output file. As a warning, do not use an important file as the output file; you will lose the contents!

Save `FileCopier.java` directly in your home directory. This will be important in the last section.

Compile it, test it, and fix any errors. You can, of course, open the output file with DrJava.

4 Showing a file in a window

Switch roles: `s2` drives and `s1` navigates.

A `javax.swing.JTextArea` is a typing area, which is a Graphical User Interface (GUI) *component.* Most GUI components go in the *content pane* of a `JFrame`. The content pane is a *container*, because it can contain components.

Download TextJFrame.java from the Labs page on the course website. Open it in DrJava, compile it, and in the Interactions Pane make a new TextJFrame and display it:

```
new TextJFrame().setVisible(true);
```

Using a method in the JTextArea class, set the text in the text area to "This is Lab 6.". Find out which method to use by looking up JTextArea in the Java API (google jtextarea or go directly to the main API page):

```
http://java.sun.com/javase/6/docs/api/
```

Do the following:

1. **Switch roles: s1 drives and s2 navigates.**
2. In the constructor, use the method showInputDialog(String) from the class JOptionPane to prompt the user for a filename.

   To find out more about the method showInputDialog, look up JOptionPane in the Java API.
3. **Switch roles: s2 drives and s1 navigates.**
4. Write an instance method readAndDisplay(String) that reads the contents of a file named by the parameter and shows it in the JTextArea.

   The code will be similar to your FileCopier.copy method.

   You will need to import the java.io package.

   Remember that method readLine might throw an IOException. You must add throws IOException to your method header.

   Compile it, test it, and fix any errors.
5. Change the constructor so that it reads the contents of that file (given by the user) and displays it in the text area.

   Demonstrate to your TA that it works, if they're not totally swamped. :-)