APS101 lab 7 – week 9

This document contains the instructions for the week 9 APS101S lab.

1   Objectives

1. Practice with arrays.

2   Icebreaker

Pick a new partner!

3   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.

4   Arrays of `chars`

Download `OurString.java` from the Labs page on the course website.

Complete the methods in `OurString`, alternating driver and navigator for each one. Do NOT use any `String` variables or methods!

5   Testing an array of `chars`

Write a JUnit tester class called `OurStringTester.java` for the program that you wrote in the previous section. If you need a reminder of how to write a tester class, take a look at `BookTester.java` from Lab 4.

Make sure to test each method in `OurString.java` *thoroughly*. This means that some methods require you to test more than one scenario (i.e., set of inputs and expected outputs).

Show the TA your test cases in order to demonstrate that your program works.

6   Arrays of `JFrames`

Write a class called `JFrameList` that has a single `JFrame[]` instance variable. The constructor should take the size of the array as a parameter, and should also fill the array with `JFrame`s. Make each `JFrame` 150x150, and set their title to `"Window i"`, where i is their index in the array (e.g. the first `JFrame` will have the title `"Window 0"`).

Write a `setVisibleAll` method that displays/hides all the `JFrame`s.

Compile it and test it from the Interactions Pane.

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

As you saw in week 3, the following code retrieves the screen width and height:

```
Dimension d = Toolkit.getDefaultToolkit().getScreenSize();
int screenWidth = (int) d.getWidth();
int screenHeight = (int) d.getHeight();
```

Write a `displayDownRight` method that sets the location of all the `JFrames` so that they are spread evenly on the screen diagonal from top left to bottom right. Put the first `JFrame` at (0, 0) and the last `JFrame` so that its bottom right corner is touching the bottom right corner of the screen.

Compile it and test it from the Interactions Pane, and also look at the window titles to see if you have arranged them properly.

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

Write a `displayDownLeft` method that sets the location of all the `JFrames` so that they are spread evenly on the screen diagonal from top right to bottom left.

Compile it and test it from the Interactions Pane.