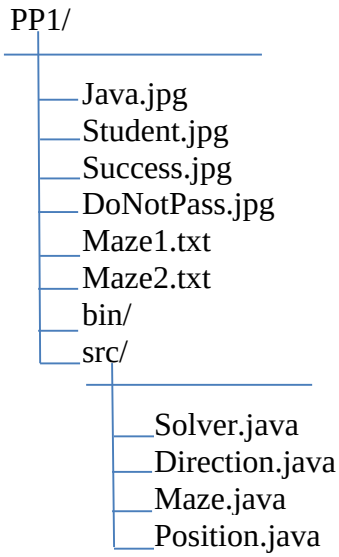
Part 1: Project Setup

Create a new java project and name it Project PP1. This project has Java files for the graphical user interface and several images. Use Eclipse IDE to create a PP1 project. Open the MazeSolver folder and copy the image and maze files from provided Files folder into../**PP1**, and copy all the ***.java** source files form the provided Source folder to **../PP1/src/**. You may change Student.jpg image with the same name containing your photo in .jpg format. In order to get your project execute correctly, your PP1 folder, in the Java workspace, should look like this: PP1 directory should look like this:



`Solver.java` is the entry (main) class of the game that has the main method, so every time you need to run the game, you should use the run command in Eclipse IDE with this class. Try to start the maze, you will be prompt to choose maze file stored in your machine, for now, select Maze1.txt, then click on Start Maze button. For now, the student icon won't move. Once all files are in place, observe what will happen when you try these steps (call some methods) in the `void run()` method of `Direction.java`:

1. Test your program by making the student image moving one square to the right, left, up, or down through calling movement methods `maze.moveRight()`, `maze.moveLeft()`, `maze.moveUp()`, or `maze.moveDown()` respectively.
2. The move methods return boolean values report if the move was successful or not.
3. Add code to print "Success" or "Failure" word in the text area of Location.java class (`location.textArea.append("Success" + "\n");`) based on the return value from each move.
4. Add calls to show the current student image row and column in the text area using this command (`location.textArea.append("Moved to row " +`

```
maze.getCurrRow() + ", column " + maze.getCurrCol() +
"\n");)
```

5. The `maze.isDone()` method in the maze object returns a boolean , this boolean is true when the student movement has found the Java logo, otherwise it is false.
6. Write enough calls to `maze.moveRight()` to find the Java logo.
7. Call `maze.isDone()` before and after the last move, and output the result.

This setup of project PP1 should be enough to start Part II; you will need to do more work to complete the PP1.