## Part II: Finding Java Logo in a Maze

In this project, you should follow a set of rules moving a student i around the maze to find the Java logo. You must follow the precise rules specified below to finding the Java logo.

### Instructions:

Remove (comment) all code from Part I that moves the student icon in the maze. Then add code to implement the movement rules shown below. This will require multiple conditional statements and control loops. Note that, some maze configurations cannot be solved using these rules, but your program won't be tested with them. You should read a complete specification of the Maze methods, which you will call in the void `run()` method code of `Direction.java` class:

```
// Get height (rows) of maze
public int getHeight()

// Get width (columns) of maze
public int getWidth()

// Get current row of the student icon
public int getCurrRow()

// Get current column of the student icon
public int getCurrCol()

// Move methods, returns true if move successful, false is encountered an
// obstacle
public boolean moveRight()
public boolean moveLeft()
public boolean moveUp()
public boolean moveDown()

// Returns true when the student finds the java logo, false otherwise
public boolean isDone()
```

### Movement's Rules

1. The student icon is always in the row and column zero
2. The Java logo can be anywhere in the maze
3. Maze contains moving obstacles shown as "Wrong Way" signs
4. The first row and column is index 0, the second row and column is index 1, and so on. The number zero is even number
5. Following the steps below, you should move student icon from left to right
6. On even columns, you must move up to down, on odd columns, you must move down to up
7. Call `maze.isDone()` for every move to see if you have found the Java logo

8. After completing each column (either moving to the top or down), use move up/down method to proceed to the next row, stop when you reach the last cell (with index of row 0 and number of column-1) or find the Java logo
9. You should check the column and row boundaries to start moving in the reverse direction when boundaries are reached
10. You should check the return value from move methods for every call, true means no obstacle, false means obstacle (cannot move).
11. At row 0 or the last row, where you cannot move up or down any further, respectively, move right in order to either move down in even rows or up in odd rows
12. If you encounter an obstacle when moving down on even columns:
    - Move right, down, down, and then left
    - Adjust the control loops for the extra moves!
13. If you encounter an obstacle when moving up on odd columns:
    - Move left, up, up, and then right.
    - Adjust the control loop for the extra moves!
14. When the Java logo is found, you must immediately break out of all loops, and show an appropriate message in the text area indicating Java Logo is found

You should test your code with the provided 2 mazes files. In addition to checking visually, you should show in the text area the row and column of the student, and you can use that to debug your code. For example, here is the output of moving the student from the top left corner to the right:

```
Maze width: 10
Maze height: 5
Start row 0, column 0
Moved to row 0, column 1
Moved to row 0, column 2
Moved to row 0, column 3
.
.
.
.
```

**Program Requirements:**

Your program (the code that is written in the `run()` method of the `Direction.java` class) must meet the following requirements:

- Follow the exact path/direction as specified in the movement rules
- Correctly follows the route with the provided mazes files
- Can only move to an adjacent position (square)
- Only adds/edits code in the specified class or method
- Cannot move diagonally or outside the maze