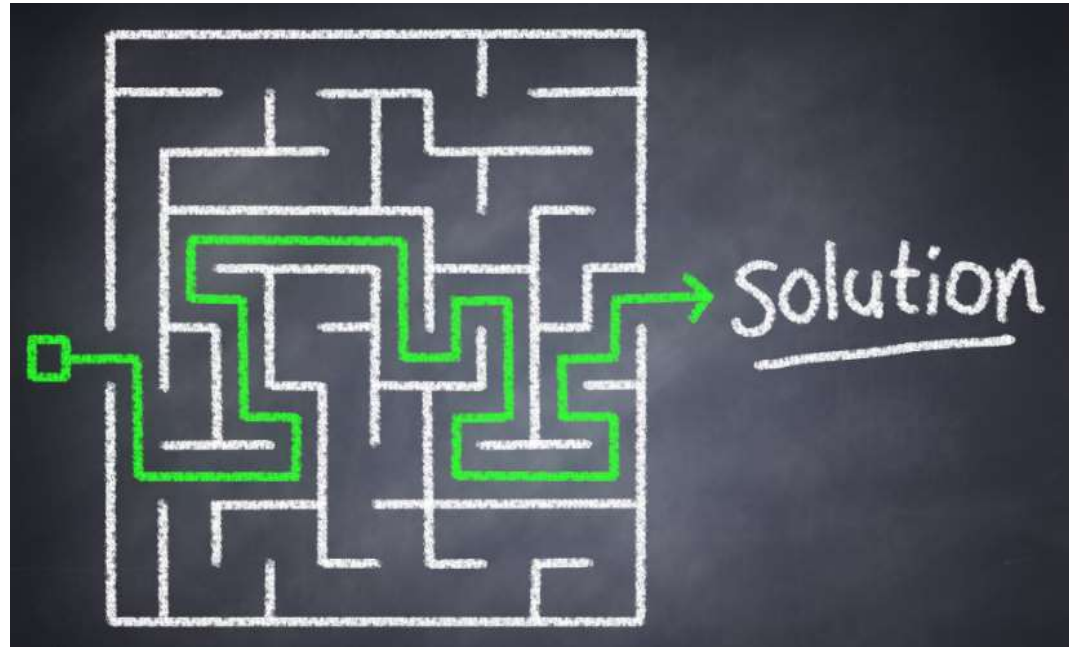
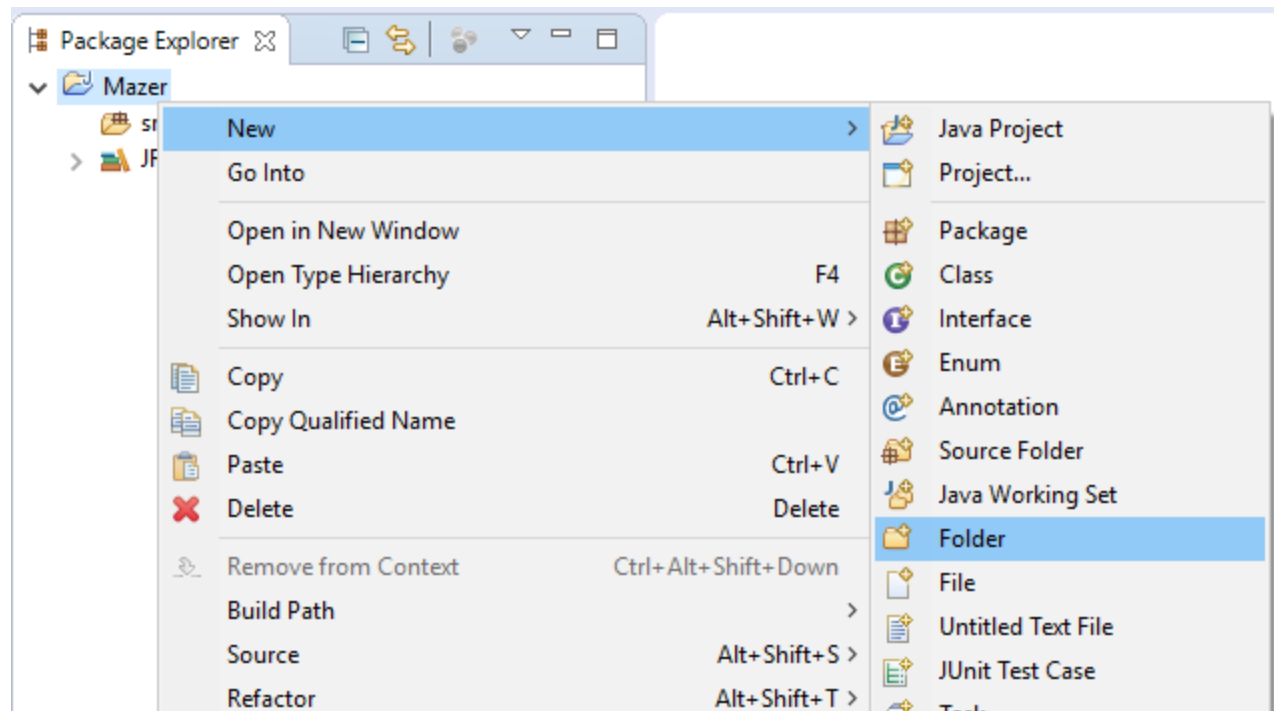


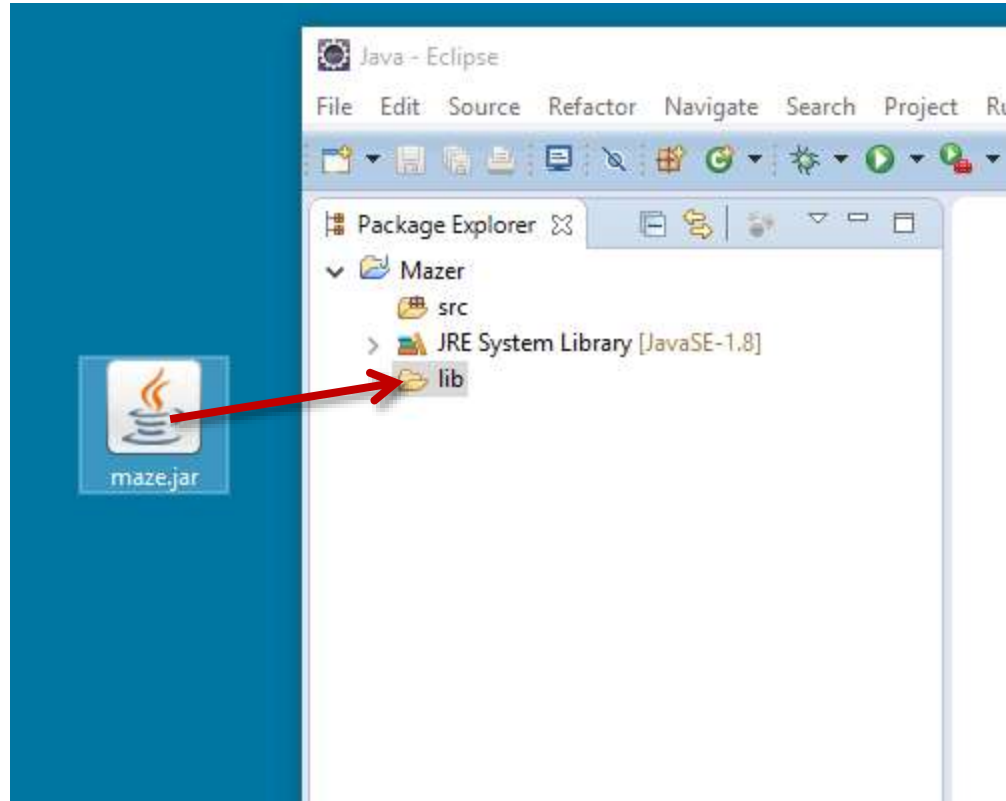
# Solution 3: The Maze Solver



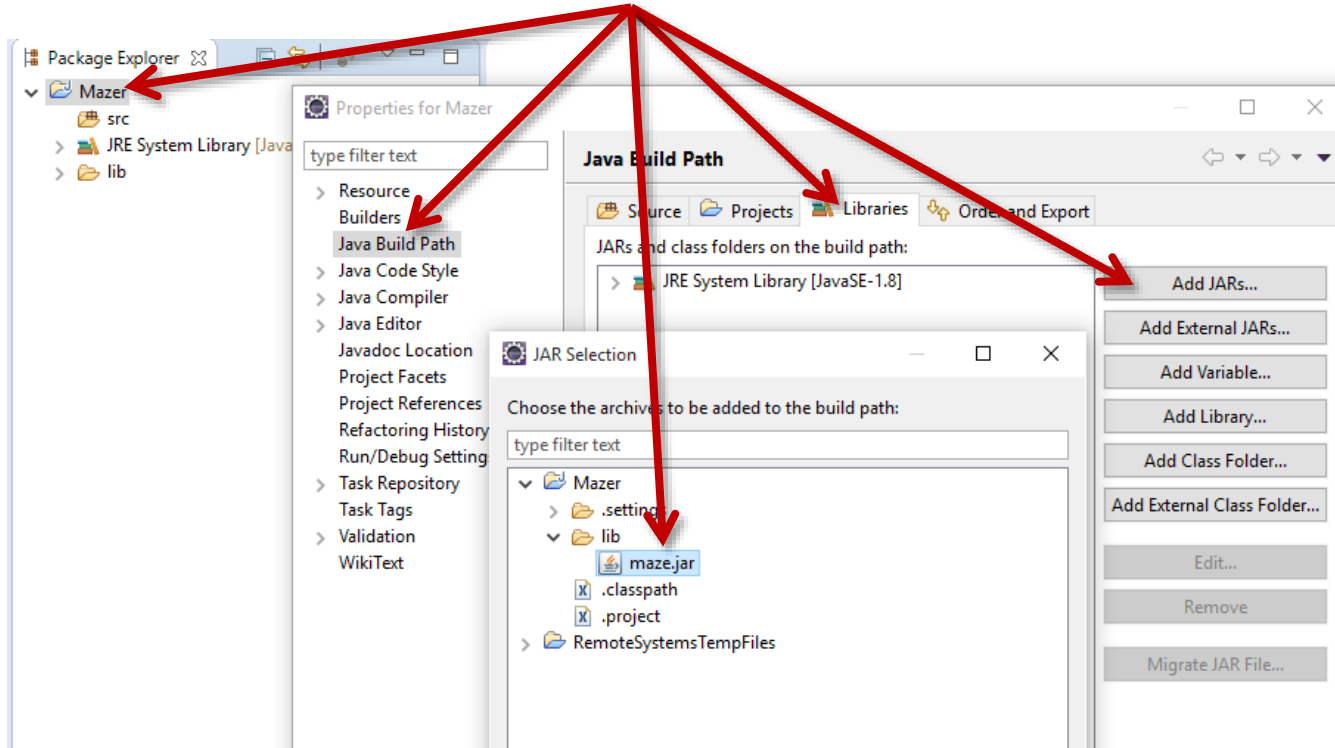
# Step 1: Create a Project



# Drag in the JAR



# Add JAR to the build path



# Browse the JAR

The screenshot shows an IDE interface with the Package Explorer on the left and the Javadoc view on the right. The Package Explorer displays the project structure, including the 'Maze' package and its classes. The 'isOpenFront()' method is selected in the Package Explorer. The Javadoc view shows the signature and description of this method.

**Package Explorer Structure:**

- Mazer
  - src
  - JRE System Library [JavaSE-1.8]
  - Referenced Libraries
    - maze.jar
    - images
    - maze
      - Maze.class
        - Maze
          - CHARS
          - DIR\_OFFS
          - MAZEWALLS
          - <clinit>(): void
          - cells
          - dir
          - exit
          - gotOut
          - player
          - rand
          - uiUpdater
          - Maze()
          - delay(): void
          - isOpenFront(): boolean
          - isOpenLeft(): boolean
          - isOpenRight(): boolean
          - moveForward(): void
          - newMaze(): void
          - randomFree(): Point
          - toString(): String
          - turnClockwise(): void
          - turnCounterClockwise(): void
- MazeUI.class
- META-INF
- lib

@ Javadoc Problems Declaration Console

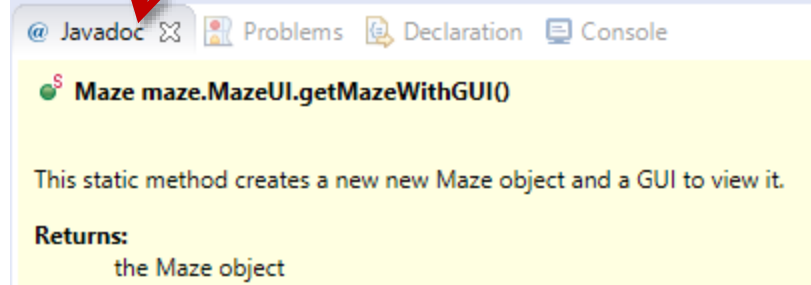
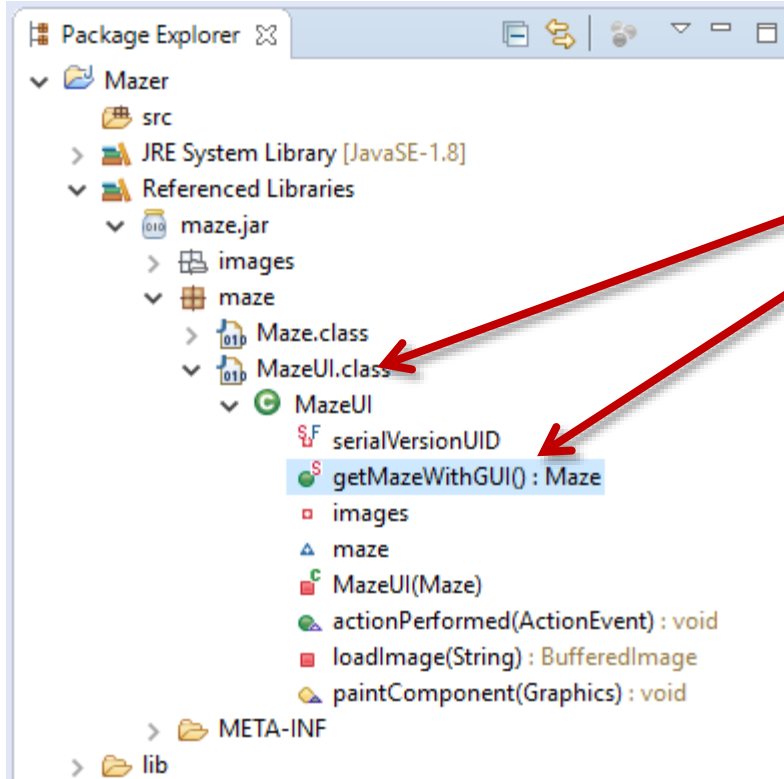
● **boolean maze.Maze.isOpenFront()**

This method returns true if the cell in front of the player is open.

**Returns:**

true if open of false if a wall

# Browse the JAR



# Make a Maze

\*Solver.java

```
1 import maze.Maze;
2 import maze.MazeUI;
3
4 public class Solver {
5
6     public static void main(String[] args) {
7
8         Maze maze = MazeUI.getMazeWithGUI();
9
10        maze.
11    }
12
13 }
14
15 }
```

- equals(Object obj) : boolean - Object
- getClass() : Class<?> - Object
- hashCode() : int - Object
- isOpenFront() : boolean - Maze
- **isOpenLeft() : boolean - Maze**
- isOpenRight() : boolean - Maze
- moveForward() : void - Maze
- newMaze() : void - Maze
- notify() : void - Object
- notifyAll() : void - Object
- toString() : String - Maze
- turnClockwise() : void - Maze
- turnCounterClockwise() : void - Maze
- wait() : void - Object
- wait(long timeout) : void - Object
- wait(long timeout, int nanos) : void - Object

Press 'Ctrl+Space' to show Template Proposals



# Remote Control

```
import maze.Maze;
import maze.MazeUI;

public class Solver {

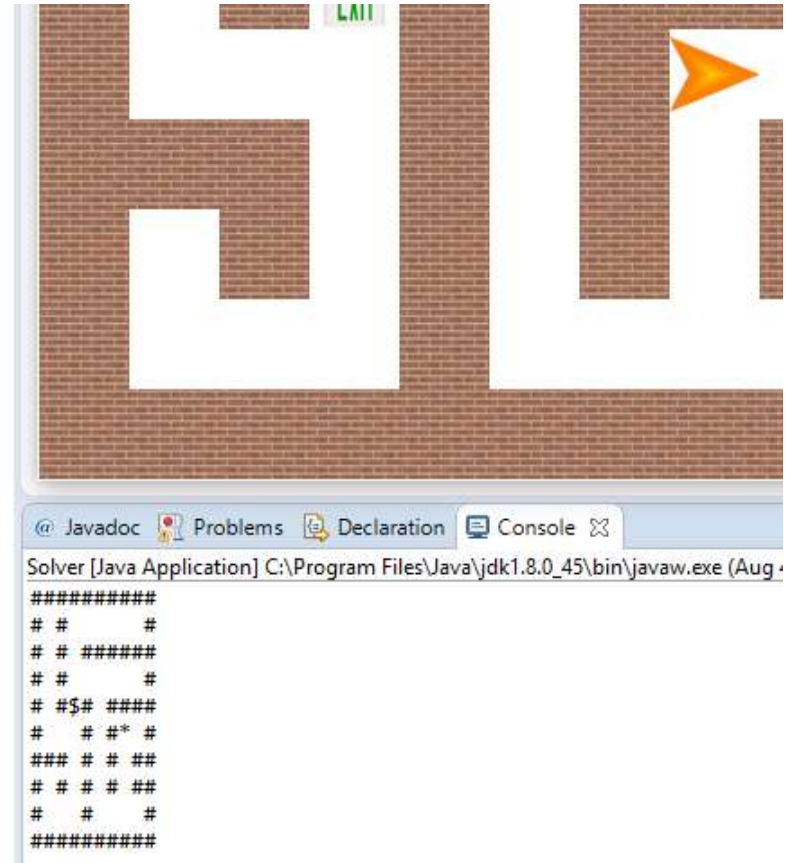
    public static int getInteger() {}

    public static void main(String[] args) {

        Maze maze = MazeUI.getMazeWithGUI();
        System.out.println(maze);

    }

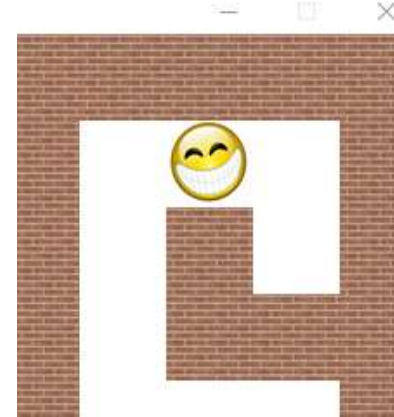
}
```





# Remote Control

```
while(true) {  
  
    boolean left = maze.isOpenLeft();  
    boolean right = maze.isOpenRight();  
    boolean front = maze.isOpenFront();  
  
    System.out.println(left+":"+front+":"+right);  
  
    int command = getInteger();  
    switch(command) {  
    case 0:  
        maze.moveForward();  
        break;  
    case 1:  
        maze.turnClockwise();  
        break;  
    case -1:  
        maze.turnCounterClockwise();  
        break;  
    }  
}
```



```
@ Javadoc Problems Declaration Console  
Solver [Java Application] C:\Program Files\Java\jdk1.8.0_45\bin\javaw.  
false:true:false  
0  
false:true:false  
1  
false:false:true  
false:true:true  
0  
You are out of the maze!  
false:true:false
```

# Robot Solver

```
import maze.Maze;
import maze.MazeUI;

public class Solver {

    public static void main(String[] args) {

        Maze maze = MazeUI.getMazeWithGUI();

        while(true) {

            if(maze.isOpenLeft()) {
                // Left is open ... turn and go left
                maze.turnCounterClockwise();
                maze.moveForward();
            } else if(maze.isOpenFront()) {
                // Forward is open ... go forward
                maze.moveForward();
            } else {
                // Left and forward blocked
                maze.turnClockwise();
            }

        }

    }

}
```

