

# CECS 277 – Lab 4

## Drawing Rectangles

Create a program that draws rectangles to the screen based on inputs from the user. Prompt the user to enter the number of rectangles they would like to enter (up to 5). Then ask the user for x and y coordinates, width, and height of each of the rectangles. Then display the rectangles layered and laid out on a grid according to the user's input.

Create a Rectangle class (in a separate file: Rect.java) with the following properties:

### Instance Variables:

1. x
2. y
3. width
4. height

### Methods:

1. constructor with parameters: x, y, width, and height
2. getX – returns the x value
3. getY – returns the y value
4. getWidth – returns the width value
5. getHeight – returns the height value

In a separate file (Main.java), create a Main class and a main method. Then create a 30x30 2D array of characters filled with '.'s and an ArrayList of Rectangles.

Create the following methods:

1. displayGrid – pass in the 2D array of chars, then display the contents of the grid.
2. getRect – pass in an integer. Prompt the user to enter the x, y, width, and height of the rectangle. Check that the user's input for the coordinates is between 1-30. Then check that the width and height is not greater than the amount of space left (ex. if the user enters a 5 for the x value, then the maximum width they can enter is 26). Construct a rectangle object using the data they entered and return it.
3. fillRect – pass in the 2D array, a rectangle, and a character. Draw a rectangle made up of the passed in character on the grid at the rectangle's x and y location with the width and height of the rectangle. Don't forget to offset the user's input (1-30) to the 0 based array indexing (0-29) when drawing.
4. getIntRange – pass in a low and high value. Check that the user's input is valid and within the low-high range. Return the validated input.

Prompt the user to enter the number of rectangles they would like to enter (1-5). Then call the getRect function to have the user to enter the data for a rectangle. Add the returned rectangle to the ArrayList of rectangles. Repeat for as many rectangles as the user entered.

After the user enters the data for each rectangle, iterate through the ArrayList and call the fillRect method to draw the rectangles onto the grid (remember that x is the horizontal axis, and y is the vertical axis, which is reversed from 2D array notation [row][col]), one at a time, using a different character (your choice, anything but a '.') for each rectangle. The rectangles should overlap any previously drawn rectangles. Finally display the grid array to the user using the displayGrid method.

Use the CheckInput class functions to make sure that all user inputs are valid and within the bounds of the array, including the size of the rectangle combined with the location.

Make sure you're using the Rect class that you created for all rectangles constructed in your program (not the Rectangle class in java.awt). Document all methods using Javadocs style comments.

## Example Output:

```
How many rectangles would you like to draw (1-5)? 7
Invalid input.

How many rectangles would you like to draw (1-5)? 3
Please enter x location of rectangle 1 (1-30): 33
Invalid input.

Please enter x location of rectangle 1 (1-30): 6
Please enter y location of rectangle 1 (1-30): 6
Please enter width of rectangle 1 (1-25): 44
Invalid input.

Please enter width of rectangle 1 (1-25): 6
Please enter height of rectangle 1 (1-25): 6
Please enter x location of rectangle 2 (1-30): 2
Please enter y location of rectangle 2 (1-30): 2
Please enter width of rectangle 2 (1-29): 5
Please enter height of rectangle 2 (1-29): 5
Please enter x location of rectangle 3 (1-30): 4
Please enter y location of rectangle 3 (1-30): 8
Please enter width of rectangle 3 (1-27): 4
Please enter height of rectangle 3 (1-23): 5
```

A uniform grid of small black dots arranged in a rectangular pattern across the entire page. The dots are evenly spaced both horizontally and vertically, creating a continuous texture of points.