

CS 46A Fall 2022

Homework 08

Requirements

1. You must name your classes **exactly** as specified. Otherwise Codecheck will not be able to process your submission and you will get no credit.
2. When you are finished with your code, submit it to Codecheck one final time then download the .signed.zip file.
3. You must upload all three signed.zip files together to Canvas and you should double check the files in Canvas to make sure all three zip files are uploaded.
4. Do not open the downloaded zip files. The files are digitally signed, and the grader program will check that they have not been opened.
5. Due time: 10 pm, Saturday, Oct 29. Submissions before the due time are not late.

You will lose five points if your submission is marked Late in Canvas.

6. Grace time: 10 am, Sunday, Oct 30. Submissions before the grace time will not be rejected.

You will receive no points if your submission is rejected by Canvas.

Remember to follow our Programming Style Requirements!

Problem 8A

You will need to import the graphics package into your BlueJ project.

Write a Java application **RandomRectangles** which prompt for a positive integer then draws that number of random rectangles. If the input is not a positive integer, the program will display a message and prompt for input again until a positive integer is entered. The program also finds the rectangle with the smallest area and display the number of the rectangles and the smallest area.

Draw the squares in red. After you have drawn all the rectangles, fill the rectangle with the smallest area with yellow. Use the predefined colors from the Color class. If more than one rectangle has the smallest area, use the last one with that area as the smallest.

You should declare and use the following constants:

```
public static final int MAX_X = 100;
public static final int MAX_Y = 500;
public static final int MIN_WIDTH = 30;
public static final int MAX_WIDTH = 100;
```

```
public static final int MIN_HEIGHT = 20;
public static final int MAX_HEIGHT = 80;

public static final int GENERATOR_SEED = 202210;
```

All rectangles have random widths and heights at random locations. The (x, y) coordinates are for the upper-left corner of the rectangle. Make the x coordinate a random number between 0 (inclusive) and 100 (exclusive), the y coordinate a random number between 0 (inclusive) and 500 (exclusive). Make the width a random value between 30 (inclusive) and 100 (exclusive), the height a random value between 20 (inclusive) and 80 (exclusive).

You must use the given seed `GENERATOR_SEED`, 202210, to generate the random numbers. For each rectangle, generate the random values in this order: x coordinate, y coordinate, width, and height. All values are integers.

You must use one for loop to draw the rectangles and find the smallest one, and not store the rectangles in an array list or an array.

You must use one Scanner object and call Scanner methods to process input. You cannot use try-catch statement or call any parse method.

The Codecheck server will run your program three times using different input sets and the expected output is attached at the end of the document.

[Codecheck link for problem 8A](#)

Problem 8B

Create a BlueJ project with three classes, **Frog**, **FrogList**, and **FrogListTester**. Copy **Frog** and **FrogListTester** from Codecheck, and do not change them in any way. You will write the entire class **FrogList**.

Class **FrogList** manages an array list of class **Frog** and the default constructor initializes the list to an empty list.

The class has the following methods:

- **public void add(Frog frog)** Adds a frog at the beginning of the list.
- **public Frog get(int index)** Returns the Frog object at position index without changing the list if the index is valid and returns null otherwise.
- **public int countInRange(int lowLimit, int highLimit)** Gets the number of frogs in the list that have a weight in the range between **lowLimit** and **highLimit**, inclusive.

- **public String toString()** Returns the string returned from calling the toString() method on the list.

Use the enhanced for loop for the **countInRange()** method.

Javadoc is required on class **FrogList**.

[Codecheck link for problem 8B](#)

Problem 8C

Create a BlueJ project with three classes, **Stock**, **StockList**, and **StockListTester**. Copy **Stock** and **StockListTester** from Codecheck and do not change them in any way. You will write the entire class **StockList**.

The **StockList** has an instance variable of **ArrayList<Stock>** and a constructor which initializes the instance variable to an empty list.

The class has the following methods:

- **public void add(Stock s)** Adds a stock s to the end of the list.
- **public void swap(int index1, int index2)** Swaps the element at index1 with the element at index2. If either index is out of bounds, do not change anything.
- **public String cheapest()** Gets the symbol for the Stock with the lowest price per share. If more than one Stock has the same price, return the symbol for the first. If the list is empty, return null.
- **public String toString()** Returns the string returned from calling the toString() method on the list.

Use only one temporary variable for the swap() method.

Use the enhanced for loop for the **cheapest()** method.

Javadoc is required on class **StockList**.

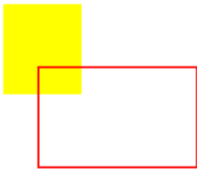
[Codecheck link for problem 8C](#)

Testing RandomRectangles.java

Test 1

```
Enter a positive integer for the number of rectangles: Java
Not an integer: "Java".
Enter a positive integer for the number of rectangles: Programming
Not an integer: "Programming".
Enter a positive integer for the number of rectangles: Q
Not an integer: "Q".
Enter a positive integer for the number of rectangles: 5
The number of rectangles: 5.
The min area: 1755.
```

Image:



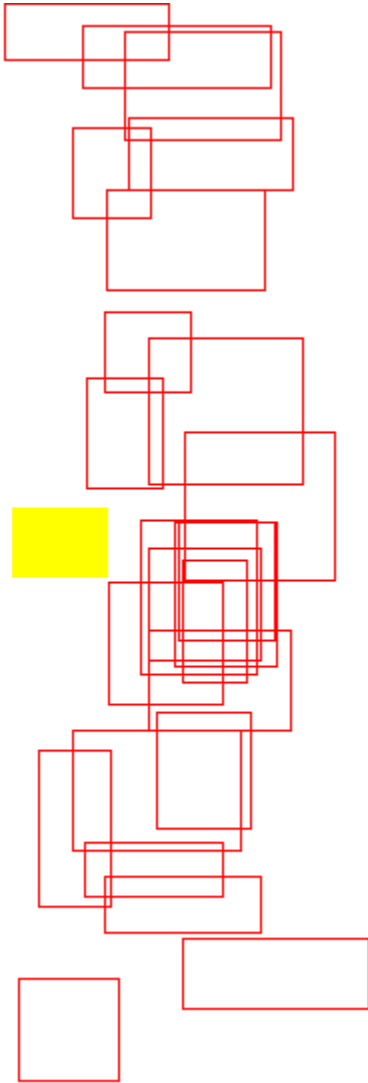
pass

Test 2

```
Enter a positive integer for the number of rectangles: -1
```

Not positive: -1.
Enter a positive integer for the number of rectangles: **0**
Not positive: 0.
Enter a positive integer for the number of rectangles: **2.5**
Not an integer: "2.5".
Enter a positive integer for the number of rectangles: **25**
The number of rectangles: 25.
The min area: 1680.

Image:



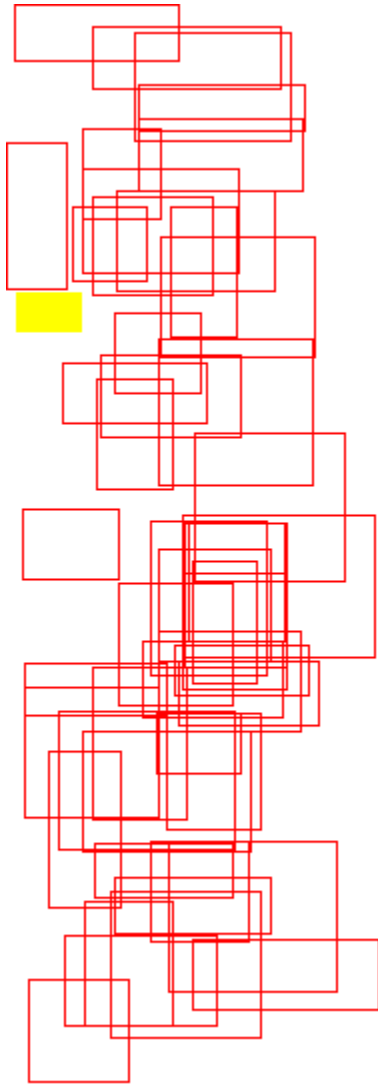
pass

Test 3

Enter a positive integer for the number of rectangles: **50**
The number of rectangles: 50.

The min area: 660.

Image:



pass