CS 46A Homework 8

Overview

In this assignment, you'll demonstrate your ability to write classes that use loops with other Java classes.

Learning Outcomes

By the end of this assignment, you should be able to write a Java program to …

- … utilize sentinel values to continue receiving output until a given value is provided.
- … use nested loops to run loops within loops.
- … use a random number generator to return random numbers in a given range.

Guidelines

1. Use BlueJ to create your code.
2. You must name your classes exactly as specified. Otherwise Codecheck will not be able to process your submission and you will get no credit.
3. When you are finished with your code, submit it to Codecheck one final time then download the `.signed.zip` file.
4. 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.
5. Do not open the downloaded zip files. The files are digitally signed, and the grader program will check that they have not been opened.

Problem 8A

Codecheck Link: HERE

Goal:
In this problem, you will write a class called **`DecisionsWithInput`** which takes in a variety of integers and keeps track of some statistics.

An example run of this program is as follows:

```
Enter an integer or Q to quit: 7
Enter an integer or Q to quit: 6
Enter an integer or Q to quit: 5
Enter an integer or Q to quit: 4
Enter an integer or Q to quit: 3
Enter an integer or Q to quit: 2
Enter an integer or Q to quit: 1
Enter an integer or Q to quit: x
The sum of all odd numbers: 16.
The smallest number: 1.
5 is my lucky number!
The count of positive numbers: 7.
```

Instructions:
Start a new BlueJ project called hw8a in the cs46a/homework/hw08 folder. In the BlueJ project, create a class called **`DecisionsWithInput`** (there is no starter code provided for this example).

Complete your program using the following pseudo-code:

- Find and print the sum of all the odd numbers (hint: the reminder of an odd number is not 0 when divided by 2)
- Find and print the smallest of the inputs
- Determine if the number 5 is in the input. Print "5 is my lucky number!" if it is in the inputs, otherwise, print "no fives".
- Print the number of values that are positive.

Tips and Guidelines:

- Read the inputs one time and do all of the processing as your go. Don't use arrays or array lists.
- Use a `Scanner` object and call `Scanner` methods to process input. Don't use try-catch statements or any parse methods.
- Define and use the following constant inside your `main()` method:

```
final int LUCKY_NUMBER = 5;
final int EVEN_NUMBER_BASE = 2;
```

Problem 8B

Codecheck Link: HERE

Goal:

In this problem, you will write a class called **`Triangle`** that draws a triangle using the plus symbol. The class will take an integer between 3 and 10 as input, indicating the number of rows in the triangle. For example, if the input is 4, then the program will draw the triangle with 4 rows as follows:

```
   +
  +++
 +++++
+++++++
```

Instructions:

Start a new BlueJ project called hw8b in the cs46a/homework/hw08 folder. In the BlueJ project, create a class called **`Triangle`** (there is no starter code provided for this example).

Other tips and guidelines:

- You can assume the input is an integer but you must check the range
- Use nested loops to draw the triangle – don't use if statements to check the integer value and draw the triangle accordingly.
- Use one Scanner object and call Scanner methods to process input.
- Numbers 2, 3, and 10 are not considered magic numbers for this problem.
- Be sure to add Javadoc comments to your code.
- An example run of this program is as follows:

```
Enter an integer between 3 and 10 (inclusive): -1
Out of range: -1.
Enter an integer between 3 and 10 (inclusive): 0
Out of range: 0.
Enter an integer between 3 and 10 (inclusive): 1
Out of range: 1.
Enter an integer between 3 and 10 (inclusive): 2
Out of range: 2.
Enter an integer between 3 and 10 (inclusive): 3
  +
 +++
+++++
```

Problem 8C

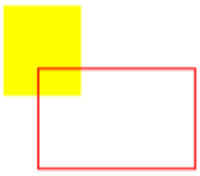
Codecheck Link: HERE

Goal:

In this problem, you will write a class called **`RandomRectangles`** which takes in an integer and draws that number of random rectangles. If the input is not a positive integer, then the program will display a message and prompt for input again. The program will also find the rectangle with the smallest area, color it yellow, and display its area.

An example run of this program is as follows:

```
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.
```



Instructions for Problem 8C:
Start a new BlueJ project called hw8c in the cs46a/homework/hw08 folder. In the BlueJ project, create a class called **`RandomRectangles`** (there is no starter code provided for this example). Add the graphics package to your project.

Complete your program using the following pseudo-code:

- Draw the rectangles in red. After you have drawn all the rectangles, fill in the smallest one with yellow. Use 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.
- 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, heights and 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) and the height a random value between 20 (inclusive) and 80 (exclusive).
- Use the given seen 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.
- Use a loop to draw the rectangles and find the smallest one. Do not store the rectangles in an array list or an array.
- Use one `Scanner` object and call `Scanner` methods to process the input.
- CodeCheck will run your program three times using different input sets. One of these outputs is shown above.