COMP10062: Assignment 4

The Assignment

This assignment is about using arrays. You will define two classes – one to represent a die with an arbitrary number of sides, and one to represent a collection of dice (these are the **model** classes). Then you will roll the entire collection repeatedly and use an array to build a histogram of the results. The **view** class for this assignment uses a `main` method and Standard Input and Output.

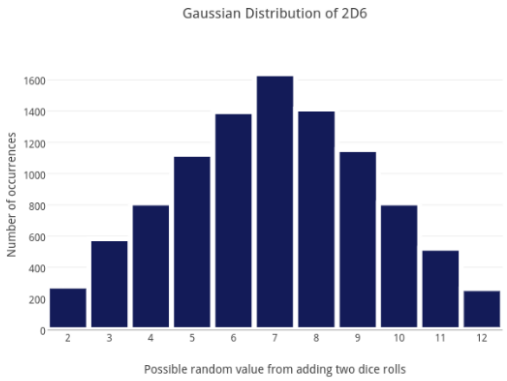
The Dice Collection (model)

A **Die** object can have any integer number of sides. The number of sides is specified when a **Die** is created and cannot be changed. A **Die** can report its number of sides, can report the side that is currently showing, and can be rolled. When a die is rolled, it generates and stores a new integer from 1 to **n**, where **n** is its number of sides. It has a **toString()** method that reports the number of sides and the number that is currently showing. That's all it can do.

A **DiceCollection** object holds a set of **Die** objects in an array. The number of sides on each **Die** is specified by passing an array of integers to the **DiceCollection** constructor. This array is used to create and store the corresponding Die objects. A **DiceCollection** object can report the current sum of all the sides showing on the dice, it can report the maximum and minimum possible sums, and it can roll all the dice at once. It has a **toString()** method that reports all the dice, the minimum possible roll, the maximum possible roll, and the current total showing on the dice. It also has a **histogram** method, described below. That's all it can do.

The Histogram Method

The **histogram** method of **DiceCollection** accepts a parameter that specifies the number of rolls it should make. Then it rolls the entire collection of dice that number of times and returns an integer array of counters (i.e., a histogram) to keep track of how many times each possible sum got rolled. For example, if the sum of the dice is 12, you should add 1 to array element 12.



See **HistogramExample.java** in this week's examples for some code that might help.

Design and Implementation

The exact details of the implementation and interface of these two classes is up to you, but you must respect the specifications given above, and you must create and hand in a UML class diagram to represent **Die**, **DiceCollection** and the association relationship between the two classes. This UML diagram can be hand drawn, or it you can use UMLet or draw.io. If you have another piece of software you would like to use, check with your instructor.

The Main Method (View)

The **main** method asks the user to enter the number of dice and number of sides for each die. Then it should create the **DiceCollection** and print it to the screen. Then it should present a menu in a loop that allows them to roll once or roll 100,000 times. The **main** method should do no calculations at all.

If they choose to roll once, show the result by printing the dice they got and the sum. If choose 100,000 rolls, call the **histogram** method described above, and print the non-zero elements of the array, as shown in the example output.

Handing In

See the due date on Canvas. Hand in by attaching a zipped version of your **.java** (not .class) files and your class diagram to the Canvas assignment.

Evaluation

Your assignment will be evaluated for performance (20%), class diagram (20%), structure (40%), and documentation (20%) using the rubric in the drop box.

Example Output

Below is an example output of the program. User input is boldfaced and red. You are free to make your interface look however you like.

```
How many dice? 4
Enter the number of sides of each die: 4 4 4 6

Dice Collection: d4=4 d4=1 d4=1 d6=4
Min=4 Max=18 Current=10

1=roll once, 2=roll 100000 times, 3=quit: 1

Dice Collection: d4=1 d4=4 d4=3 d6=3
Min=4 Max=18 Current=11

1=roll once, 2=roll 100000 times, 3=quit: 2

  4:   246
  5:  1096
  6:  2489
  7:  5253
  8:  8288
  9: 11423
 10: 13782
 11: 14621
 12: 13790
 13: 11507
 14:  8347
 15:  5257
 16:  2597
 17:  1053
 18:   251

Dice Collection: d4=1 d4=3 d4=1 d6=4
Min=4 Max=18 Current=9

1=roll once, 2=roll 100000 times, 3=quit: 3

BYE!!!
```

Optional Extra 1: Here's a nicer way to display the histogram, courtesy of Mark Yendt. Scale the numbers down and print asterisks to represent the quantities.

```
 4:  257
 5: 1056  *
 6: 2559  ***
 7: 5308  *****
 8: 8331  ********
 9:11303  ***********
10:13788  **************
11:14643  ***************
12:13747  **************
13:11398  ***********
14: 8339  ********
15: 5287  *****
16: 2670  ***
17: 1043  *
18:  271
```

Optional Extra 2: How about displaying the histogram as a bar graph using the **FXAnimationTemplate**?