Assignment 7
Due Aug 5th, @ 11:55pm

Objectives

- Review Java arrays
- Implement methods in a simple class
- Practice with incremental development and stub methods

Introduction

In this assignment we will use a random number generator to simulate a lottery. For this assignment we will simulate a simplified version of Lotto 6/49. Each ticket has six unique numbers who's values are between 1 and 49.

In the first part of the assignment you will implement static methods to generate six unique random numbers.

In the second part of the assignment you will complete a class called `LotteryNumbers` which will allow us to simulate a lottery.

Part I

Download `Part1.java`.

You will notice that this program generates six random numbers. If you run the program several times, you may notice a problem. It is possible for this code to generate the same random number more than once. Generally speaking, lottery tickets require the numbers to be unique.

You should modify `Part1.java` so that the `generate` method does not create duplicate numbers. It is easiest to get this right by adding additional static methods, rather than putting all the code inside the existing loop.

Part II

In part II you will complete the implementation of the `LotteryTicket` class. Instances of this class will model the numbers that are contained on a lottery ticket and the winning numbers that are drawn each week. You can see a skeleton of the code in `LotteryTicket.java`

If you look at `LotteryDraw.java` you will see how the `LotteryTicket` class is used. The code in `LotteryDraw` first creates your ticket and then simulates 10 lottery draws. The code checks to see if your ticket matches the winning numbers – if it does, you win.

If you compile and run `LotteryDraw`, you should see output as shown below:

```
My lucky ticket is:
{fixme}
Performing 10 draws. Good luck!
You didn't match {fixme}
You didn't match {fixme}
You didn't match {fixme}
You didn't match {fixme}
You didn't match {fixme}
You didn't match {fixme}
You didn't match {fixme}
You didn't match {fixme}
You didn't match {fixme}
You didn't match {fixme}
```

Your job is to complete the stub methods in `LotteryTicket.java` so that the output when running `LotteryDraw` looks something like:

```
My lucky ticket is:
{9,12,13,34,46,48}
Performing 10 draws. Good luck!
You didn't match {3,4,17,34,37,46}
You didn't match {3,7,10,17,21,29}
You didn't match {6,16,27,37,43,45}
You didn't match {3,18,30,38,43,48}
You didn't match {2,8,16,24,36,49}
You didn't match {2,17,27,45,47,48}
You didn't match {11,16,27,33,37,39}
You didn't match {8,15,21,25,30,45}
You didn't match {1,13,26,39,40,44}
You didn't match {8,22,23,25,26,37}
```

Be sure to read the comments at the start of each method carefully – the comments tell you the expected behavior.

Hints:

- You should be able to reuse most of the code from Part I in Part II.
- Pay particular attention to the constructor for `LotteryNumbers`. Remember that you need to allocate the array to hold the numbers inside the constructor. After you've allocated the array you need to populate it with random numbers.

Submission

Submit your modified version of the two files `Part1.java` and `LotteryTicket.java`. Be sure to include your name and student number at the top of each file in comments.

Submit the files using the automated submission web page on the course's conneX site. Please do not use the submission system used for labs. In the very unlikely event that there is a problem with conneX which prevents you from submitting your files, you may e-mail them to the course instructor (gsrivast@uvic.ca). This is for emergency use only.

Grading

You submit something that compiles with no warnings 1 mark
Your part I works correctly 2 marks
Your part II works correctly 7 marks

If you do not submit correctly named files or your programs do not compile, you will receive 0 for the assignment. Do not change any class names or method names. The method names and the method signatures (i.e. the result types and the argument types) are part of the interface for the class. To receive full credit, the code you submit for the assignment must match that interface.