Exercise 3

Objectives

The aim of this exercise is to learn the basic structure of Java programs. We will solve problems using variables, arrays, various operators, and flow control statements. Although there are many similarities to the syntax of procedural languages such as C, the following are important concepts of the Java language:

- Arrays as objects (array assignment operations)
- Multi-dimensional array

Also included are some parts not covered in the lecture, such as the standard input method, which should be used as a template without going into too much detail.

A: Static Fields [1 pt]

Your task is to extend the Point class. In the extended Point class, X_MIN, Y_MIN, X_MAX, and Y_MAX can be set for the range of possible x-coordinates and y-coordinates, respectively. However, these values are shared by the class, not by individual instances.

```
class RangingPointApplication{
   public static void main(String[] args){
      Point p1 = new Point();
      Point p2 = new Point();
      Point.X_MIN = -100;   // update of the static field
      Point.X_MAX = 100;
      Point.Y_MIN = -100;
      Point.Y_MAX = 100;

      p1.move(50, 50);

      p1.move(100, 100);
      p2.move(-100, -100);

      System.out.println("(" + p1.getX() + ", " + p1.getY() + ")");
      System.out.println("(" + p2.getX() + ", " + p2.getY() + ")");

      p2.X_MAX = 200;
      p2.Y_MAX = 200;
      p1.X_MIN = -200;
      p1.Y_MIN = -200;

      p1.move(100, 100);
      p2.move(-100, -100);

      System.out.println("(" + p1.getX() + ", " + p1.getY() + ")");
      System.out.println("(" + p2.getX() + ", " + p2.getY() + ")");
   }
}
```

Test the Point class with RangingPointApplication shown above. (You cannot modify it)

Note that the move method ignores the process if the resulting point position is out of the range.

If RangingPointApplication produces the following output, it can be judged that the Point class has been implemented correctly.

Output

```
(50, 50)
(-100, -100)
(150, 150)
(-200, -200)
```

Submission Files	Types
Point.java	Java Class

B: Array Assignment [1 pt]

Create and execute the program ArrayAssignmentApplication shown below.

```
class ArrayAssignmentApplication{
   public static void main(String[] args){
      int a[] = {1, 2, 3};
      int b[] = {4, 5, 6};

      System.out.println("Array a : " + a[0] + ", " + a[1] + ", " + a[2]);
      System.out.println("Array b : " + b[0] + ", " + b[1] + ", " + b[2]);
      System.out.println();

      b = a;

      System.out.println("Array a : " + a[0] + ", " + a[1] + ", " + a[2]);
      System.out.println("Array b : " + b[0] + ", " + b[1] + ", " + b[2]);
      System.out.println();

      a[0] = 777;

      System.out.println("Array a : " + a[0] + ", " + a[1] + ", " + a[2]);
      System.out.println("Array b : " + b[0] + ", " + b[1] + ", " + b[2]);
   }
}
```

```
Array a : 1, 2, 3
Array b : 4, 5, 6

Array a : 1, 2, 3
Array b : 1, 2, 3

Array a : 777, 2, 3
Array b : 777, 2, 3
```

Why are both a[0] and b[0] "777" ? Please explain (in `ArrayAssignment.txt`)

Submission Files	Types
ArrayAssignment.txt	Texts in English

C: Sorting Arrays [2 pt]

Complete the following SortingArraysApplication class. This application performs a sorting algorithm on each of the six array objects, which are recorded as two-dimensional arrays. The sorting algorithm arranges the elements of the target array in ascending order.

```
class SortingArraysApplication{
    public static void main(String[] args){

        int[][] arrays = {
            {3, 2, 1},
            {9, 7, 10, 4, 8, 2, 5, 3, 1, 6},
            {7, 2, 3, 2, 2, 1, 1, 1},
            {2, 1},
            {100},
            {888888888, -888888888}
        };

        /**

        your code

        */

        for ( int i = 0; i < arrays.length; i++ ){
            for ( int j = 0; j < arrays[i].length; j++ ){
                System.out.print(" " + arrays[i][j]);
            }
            System.out.println();
        }
    }
}
```

If SortingArraysApplication produces the following output, it can be judged that your program has been implemented correctly.

Output

```
 1 2 3
 1 2 3 4 5 6 7 8 9 10
 1 1 1 2 2 2 3 7
 1 2
 100
 -888888888 888888888
```

Submission Files	Types
SortingArraysApplication.java	Java Class

D: Stack [2 pt]

Create a LIFO data structure, the Stack class. This time, the elements of the stack should be integers and should be held in a simple array with sufficient capacity (for example, 100). Implement the push method to add an element and the pop method to remove an element.

Create a StackApplication class that simulates the Stack class you created. The simulator should read a sequence of integers a_i and perform the following operations:

- adds a_i to the stack when the integer a_i is greater than or equal to 1
- removes an element from the stack and outputs it when a_i is 0, and
- terminates the program when a_i is a negative number.

You can use the following template.

```
import java.util.Scanner;

class StackApplication{
    public static void main(String[] args){
        Stack st = new Stack();

        Scanner sc = new Scanner(System.in);
        while( true ){
            int x = sc.nextInt();
            if ( x < 0 ) break;
            // your code



        }
    }
}
```

Check the behavior of the program created with the following input/output data.

Sample Input	Sample Output
8 7 5 3 9 8 11 0 0 10 21 8 6 4 0 0 0 7 2 3 0 9 0 0 0 0 15 17 0 -1	11 8 4 6 8 3 9 2 7 21 17

Submission Files	Types
Stack.java	Java Class
StackApplication.java	Java Class

Summary

In this exercise, we learned Java language structures such as flow control for implementing algorithms. We also learned how to manipulate array objects, which are reference types as well as two-dimensional array. We also implemented a generic data structure like Stack. Thus, in this course, we may implement data structures and algorithms that are already available in standard libraries from scratch as teaching materials. These are actually prepared as libraries, but let's look at their features and learn from them.