Jinan University

Java Programming Lab Report

Major: Computer Science & Technology
School: International School

Student name : _____
(p.s. your name on JNU academic system)

Student number : _____

Date of Submission (mm-dd-yyyy): _____

Instructor: Yuxia Sun

Table of Content

LAB 4 DATE: 4/11/2023

Student Name:_____ Student ID: _____

Problem 1. (7.13)

***7.13** (Random number chooser) Write the following method that returns a random number between `start` and `end`, excluding the `numbers`.

```
public static int getRandom(int start, int end, int... numbers)
```

For example, invoking `getRandom(1,100,4,8,95,93)` returns a random number between `1` and `100` excluding `4,8,95,` and `93`. Write a test program that invokes `getRandom(1,100,4,8,95,93)` 45 times and displays the resulting numbers 15 per line using the format `%4d`.

**** Source Code / Solution :***

```
public class RandomNumberChooser {
   public static int getRandom(int start, int end, int... numbers){
      boolean success = false;
      int random = 0;

      while(!success) {
         success = true;
         random = (int)((end - start) * Math.random() + start);
         for (int exclude : numbers) {
            if (random == exclude) {
               success = false;
               break;
            }
         }
      }

      return random;
   }

   public static void main(String[] args) {
      for(int i = 0; i < 3; i++){
         for(int j = 0; j < 15; j++){
```

```
            System.out.printf("%4d", getRandom(1,100,4,8,95,93));
        }
        System.out.println();
    }
  }
}
```

**** Output:***

```
RandomNumberChooser ×
/Users/h3art/Library/Java/JavaVirtualMachines/openjdk-18.0.2/
  59  30  56  74  24  37  64  32  43  96  24  42  33  97   9
  72  76  34  19  52  18  46  79  76  75  86  33   9  98  54
  54  91   7  98   5  27  27  33  64  88  34  87  86  59  80

进程已结束,退出代码0
```

**** Debugging/Testing:***

Bug1: The generation formula of the random number is incorrect, resulting in the generation of a random number outside the given range.
Fix: Re-deriving the relationship between Math.random() and generating random numbers, correct the formula as random = (int)((end - start) * Math.*random*() + start);

Problem 2. (7.23)(Optional)

****7.23** (*Game: locker puzzle*) A school has 100 lockers and 100 students. All lockers are closed on the first day of school. As the students enter, the first student, denoted as S1, opens every locker. Then the second student, S2, begins with the second locker, denoted as L2, and closes every other locker. Student S3 begins with the third locker and changes every third locker (closes it if it was open and opens it if it was closed). Student S4 begins with locker L4 and changes every fourth locker. Student S5 starts with L5 and changes every fifth locker, and so on, until student S100 changes L100.

After all the students have passed through the building and changed the lockers, which lockers are open? Write a program to find your answer and display all open locker numbers separated by exactly one space.

(*Hint*: Use an array of 100 Boolean elements, each of which indicates whether a locker is open (**true**) or closed (**false**). Initially, all lockers are closed.)

**** Source Code / Solution :***

```java
public class LockerPuzzle {
    private final static int TOTAL = 100;

    private static boolean[] locker = new boolean[TOTAL];

    public static void main(String[] args) {
        for (int student = 1; student <= TOTAL; student++) {
            for (int position = student - 1; position < TOTAL; position += student){
                locker[position] = !locker[position];
            }
        }

        for (int denote = 0; denote < TOTAL; denote++){
            if(locker[denote]){
                System.out.print(denote + 1);
                System.out.print(" ");
            }
        }

    }
}
```

**** Output:***

```
LockerPuzzle ×
/Users/h3art/Library/Java/JavaVirtualMachines
1 4 9 16 25 36 49 64 81 100
进程已结束,退出代码0
```

**** Debugging/Testing:***

Bug1: There is an error in the processing of array index. When there are 100 lockers, the maximum index is 99.
Fix: Modify the correspondence between student and locker to prevent ArrayIndexOutOfBoundsException.

Problem 3. (8.5)

8.5 (*Algebra: add two matrices*) Write a method to add two matrices. The header of the method is as follows:

```
public static double[][] addMatrix(double[][] a, double[][] b)
```

In order to be added, the two matrices must have the same dimensions and the same or compatible types of elements. Let **c** be the resulting matrix. Each element c_{ij} is $a_{ij} + b_{ij}$. For example, for two 3×3 matrices **a** and **b**, **c** is

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} + \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{pmatrix} = \begin{pmatrix} a_{11} + b_{11} & a_{12} + b_{12} & a_{13} + b_{13} \\ a_{21} + b_{21} & a_{22} + b_{22} & a_{23} + b_{23} \\ a_{31} + b_{31} & a_{32} + b_{32} & a_{33} + b_{33} \end{pmatrix}$$

Write a test program that prompts the user to enter two 3×3 matrices and displays their sum. Here is a sample run:

```
Enter matrix1: 1 2 3 4 5 6 7 8 9 ↵Enter
Enter matrix2: 0 2 4 1 4.5 2.2 1.1 4.3 5.2 ↵Enter
The matrices are added as follows
 1.0 2.0 3.0      0.0 2.0 4.0      1.0 4.0 7.0
 4.0 5.0 6.0  +   1.0 4.5 2.2  =   5.0 9.5 8.2
 7.0 8.0 9.0      1.1 4.3 5.2      8.1 12.3 14.2
```

**** Source Code / Solution :***

```java
import java.util.Scanner;

public class AddTwoMatrices {
   final static int STAGE = 3;

   public static double[][] initMatrix(Scanner input) {
      double[][] matrix = new double[STAGE][STAGE];
      for (int i = 0; i < STAGE; i++) {
         for (int j = 0; j < STAGE; j++) {
            matrix[i][j] = input.nextDouble();
         }
      }
      return matrix;
   }

   public static double[][] addMatrix(double[][] matrix1, double[][] matrix2) {
      double[][] result = new double[STAGE][STAGE];

      for (int i = 0; i < STAGE; i++) {
         for (int j = 0; j < STAGE; j++) {
```

```
            result[i][j] = matrix1[i][j] + matrix2[i][j];
        }
    }

    return result;
  }

  public static void printProcess(double[][] addMatrix1, double[][] addMatrix2, double[][]
result) {
    System.out.printf("%f %f %f     %f %f %f     %f %f %f\n",
            addMatrix1[0][0], addMatrix1[0][1], addMatrix1[0][2],
            addMatrix2[0][0], addMatrix2[0][1], addMatrix2[0][2],
            result[0][0], result[0][1], result[0][2]
    );
    System.out.printf("%f %f %f  +  %f %f %f  =  %f %f %f\n",
            addMatrix1[1][0], addMatrix1[1][1], addMatrix1[1][2],
            addMatrix2[1][0], addMatrix2[1][1], addMatrix2[1][2],
            result[1][0], result[1][1], result[1][2]
    );
    System.out.printf("%f %f %f     %f %f %f     %f %f %f\n",
            addMatrix1[2][0], addMatrix1[2][1], addMatrix1[2][2],
            addMatrix2[2][0], addMatrix2[2][1], addMatrix2[2][2],
            result[2][0], result[2][1], result[2][2]
    );
  }

  public static void main(String[] args) {
    double[][] matrix1, matrix2;

    Scanner input = new Scanner(System.in);
    System.out.println("Please enter two 3 * 3 matrices(real numbers seperated by
SPACE)");

    System.out.print("Enter matrix1: ");
    matrix1 = initMatrix(input);
    System.out.print("Enter matrix2: ");
    matrix2 = initMatrix(input);

    // if I close the Scanner(System.in) in a function,
```

```
        // it will close System.in as well
        // when I want to use it again, the System.in has already
        // been closed, therefore I cannot get any data from System.in
        // finally if I use Scanner.nextXXX(), then I will receive a
        // IOException
        input.close();

        System.out.println("The matrices are added as follows");
        printProcess(matrix1, matrix2, addMatrix(matrix1, matrix2));
    }
}
```

**** Output:***

```
AddTwoMatrices
/Users/h3art/Library/Java/JavaVirtualMachines/openjdk-18.0.2/Contents/Home/bin/java -javaagent
Please enter two 3 * 3 matrices(real numbers seperated by SPACE)
Enter matrix1: 1 2 3 4 5 6 7 8 9
Enter matrix2: 0 2 4 1 4.5 2.2 1.1 4.3 5.2
The matrices are added as follows
1.000000 2.000000 3.000000     0.000000 2.000000 4.000000     1.000000 4.000000 7.000000
4.000000 5.000000 6.000000  +  1.000000 4.500000 2.200000  =  5.000000 9.500000 8.200000
7.000000 8.000000 9.000000     1.100000 4.300000 5.200000     8.100000 12.300000 14.200000

进程已结束,退出代码0
```

**** Debugging/Testing:***

Bug1: If I new a Scanner with System.in as the input source in a function, and close the Scanner after the function is called, System.in will be closed and cannot be opened again, finally when I want to directly read other data, the program will throw a IOException.
Fix: Set the Scanner used by the entire program in the main function, and wait for all input to be completed before calling the close() method.

Problem 4. (8.27)

***8.27** (*Column sorting*) Implement the following method to sort the columns in a two-dimensional array. A new array is returned and the original array is intact.

```
public static double[][] sortColumns(double[][] m)
```

Write a test program that prompts the user to enter a 3 × 3 matrix of double values and displays a new column-sorted matrix. Here is a sample run:

```
Enter a 3-by-3 matrix row by row:
0.15 0.875 0.375 ↵Enter
0.55 0.005 0.225 ↵Enter
0.30 0.12 0.4 ↵Enter

The column-sorted array is
0.15 0.0050 0.225
0.3  0.12   0.375
0.55 0.875  0.4
```

**** Source Code / Solution :***

```java
import java.util.Scanner;

public class ColumnSorting {
    final static int STAGE = 3;

    public static double[][] initMatrix(Scanner input) {
        double[][] matrix = new double[STAGE][STAGE];
        for (int i = 0; i < STAGE; i++) {
            for (int j = 0; j < STAGE; j++) {
                matrix[i][j] = input.nextDouble();
            }
        }
        return matrix;
    }

    // Sort in increasing order
    public static double[] bubbleSort(double num1, double num2, double num3) {
        double[] result = new double[STAGE];
        result[0] = num1;
        result[1] = num2;
        result[2] = num3;
        for (int i = 0; i < STAGE - 1; i++) {
            for (int j = 0; j < STAGE - 1 - i; j++) {
                if (result[j] > result[j + 1]) {
                    double temp = result[j];
                    result[j] = result[j + 1];
                    result[j + 1] = temp;
```

```
            }
        }
    }
    return result;
}

public static double[][] sortColumns(double[][] matrix) {
    double[][] sortedMatrix = new double[STAGE][STAGE];

    for (int i = 0; i < STAGE; i++) {
        double[] tempColumn = bubbleSort(matrix[0][i], matrix[1][i], matrix[2][i]);
        for (int j = 0; j < STAGE; j++) {
            sortedMatrix[j][i] = tempColumn[j];
        }
    }

    return sortedMatrix;
}

public static void displayMatrix(double[][] matrix) {
    for (int i = 0; i < STAGE; i++) {
        for (int j = 0; j < STAGE; j++) {
            System.out.printf("%f ", matrix[i][j]);
        }
        System.out.println();
    }
}

public static void main(String[] args) {
    double[][] matrix;

    Scanner input = new Scanner(System.in);

    System.out.println("Enter a 3-by-3 matrix row by row:");
    matrix = initMatrix(input);
    input.close();

    System.out.println("The column-sorted array is");
    displayMatrix(sortColumns(matrix));
```

```
  }
}
```

**** Output:***

```
ColumnSorting ×
/Users/h3art/Library/Java/JavaVirtualMachines
Enter a 3-by-3 matrix row by row:
0.15 0.875 0.375
0.55 0.005 0.225
0.30 0.12 0.4
The column-sorted array is
0.150000 0.005000 0.225000
0.300000 0.120000 0.375000
0.550000 0.875000 0.400000

进程已结束,退出代码0
```

**** Debugging/Testing:***

Bug1: Failed to wrap a swap function.
Fix: In Java, pass the primitive type variable to a function, the function will only get a copy value of corresponding parameters. Unlike c/c++, I cannot use pointer to modify the variable in other function. Finally I didn't wrap a swap function.

Problem 5. (8.37)(Optional)

****8.37** (*Guess the capitals*) Write a program that repeatedly prompts the user to enter a capital for a state. Upon receiving the user input, the program reports whether the answer is correct. Assume that **50** states and their capitals are stored in a two-dimensional array, as shown in Figure 8.10. The program prompts the user to answer all states' capitals and displays the total correct count. The user's answer is not case-sensitive.

```
Alabama        Montgomery
Alaska         Juneau
Arizona        Phoenix
...            ...
...            ...
```

FIGURE 8.10 A two-dimensional array stores states and their capitals.

Here is a sample run:

```
What is the capital of Alabama? Montogomery ↵Enter
The correct answer should be Montgomery
What is the capital of Alaska? Juneau ↵Enter
Your answer is correct
What is the capital of Arizona? ...
...
The correct count is 35
```

**** Source Code / Solution :***

```java
import java.util.Scanner;

public class GuessTheCapital {
  // Define the states and their capitals as a 2D array
  private final static String[][] statesAndCapitals = {
      {"Alabama", "Montgomery"},
      {"Alaska", "Juneau"},
      {"Arizona", "Phoenix"},
      {"Arkansas", "Little Rock"},
      {"California", "Sacramento"},
      {"Colorado", "Denver"},
      {"Connecticut", "Hartford"},
      {"Delaware", "Dover"},
      {"Florida", "Tallahassee"},
      {"Georgia", "Atlanta"},
      {"Hawaii", "Honolulu"},
      {"Idaho", "Boise"},
```

```
        {"Illinois", "Springfield"},
        {"Indiana", "Indianapolis"},
        {"Iowa", "Des Moines"},
        {"Kansas", "Topeka"},
        {"Kentucky", "Frankfort"},
        {"Louisiana", "Baton Rouge"},
        {"Maine", "Augusta"},
        {"Maryland", "Annapolis"},
        {"Massachusetts", "Boston"},
        {"Michigan", "Lansing"},
        {"Minnesota", "St. Paul"},
        {"Mississippi", "Jackson"},
        {"Missouri", "Jefferson City"},
        {"Montana", "Helena"},
        {"Nebraska", "Lincoln"},
        {"Nevada", "Carson City"},
        {"New Hampshire", "Concord"},
        {"New Jersey", "Trenton"},
        {"New Mexico", "Santa Fe"},
        {"New York", "Albany"},
        {"North Carolina", "Raleigh"},
        {"North Dakota", "Bismarck"},
        {"Ohio", "Columbus"},
        {"Oklahoma", "Oklahoma City"},
        {"Oregon", "Salem"},
        {"Pennsylvania", "Harrisburg"},
        {"Rhode Island", "Providence"},
        {"South Carolina", "Columbia"},
        {"South Dakota", "Pierre"},
        {"Tennessee", "Nashville"},
        {"Texas", "Austin"},
        {"Utah", "Salt Lake City"},
        {"Vermont", "Montpelier"},
        {"Virginia", "Richmond"},
        {"Washington", "Olympia"},
        {"West Virginia", "Charleston"},
        {"Wisconsin", "Madison"},
        {"Wyoming", "Cheyenne"}
    };
```

```java
    public static void main(String[] args) {
        int correctCount = 0;

        Scanner input = new Scanner(System.in);

        for (String[] statesAndCapital : statesAndCapitals) {
            System.out.print("What is the capital of " + statesAndCapital[0] + "? ");
            String userAnswer = input.nextLine();

            if (userAnswer.equalsIgnoreCase(statesAndCapital[1])) {
                System.out.println("Your answer is correct");
                correctCount++;
            } else {
                System.out.println("The correct answer should be " + statesAndCapital[1]);
            }
        }

        System.out.println("The correct count is " + correctCount);

        input.close();
    }
}
```

**** Output:***

```
GuessTheCapital ×
What is the capital of Virginia? Richmond
Your answer is correct
What is the capital of Washington? Olympia
Your answer is correct
What is the capital of West Virginia? Charleston
Your answer is correct
What is the capital of Wisconsin? Madison
Your answer is correct
What is the capital of Wyoming? Cheyenne
Your answer is correct
The correct count is 37

进程已结束,退出代码0
```

**** Debugging/Testing:***

Bug1: Storing two-dimensional array in incorrect format, get a compile error.
Fix: Pay attention to the separation of braces and commas when initializing a two-dimensional array.