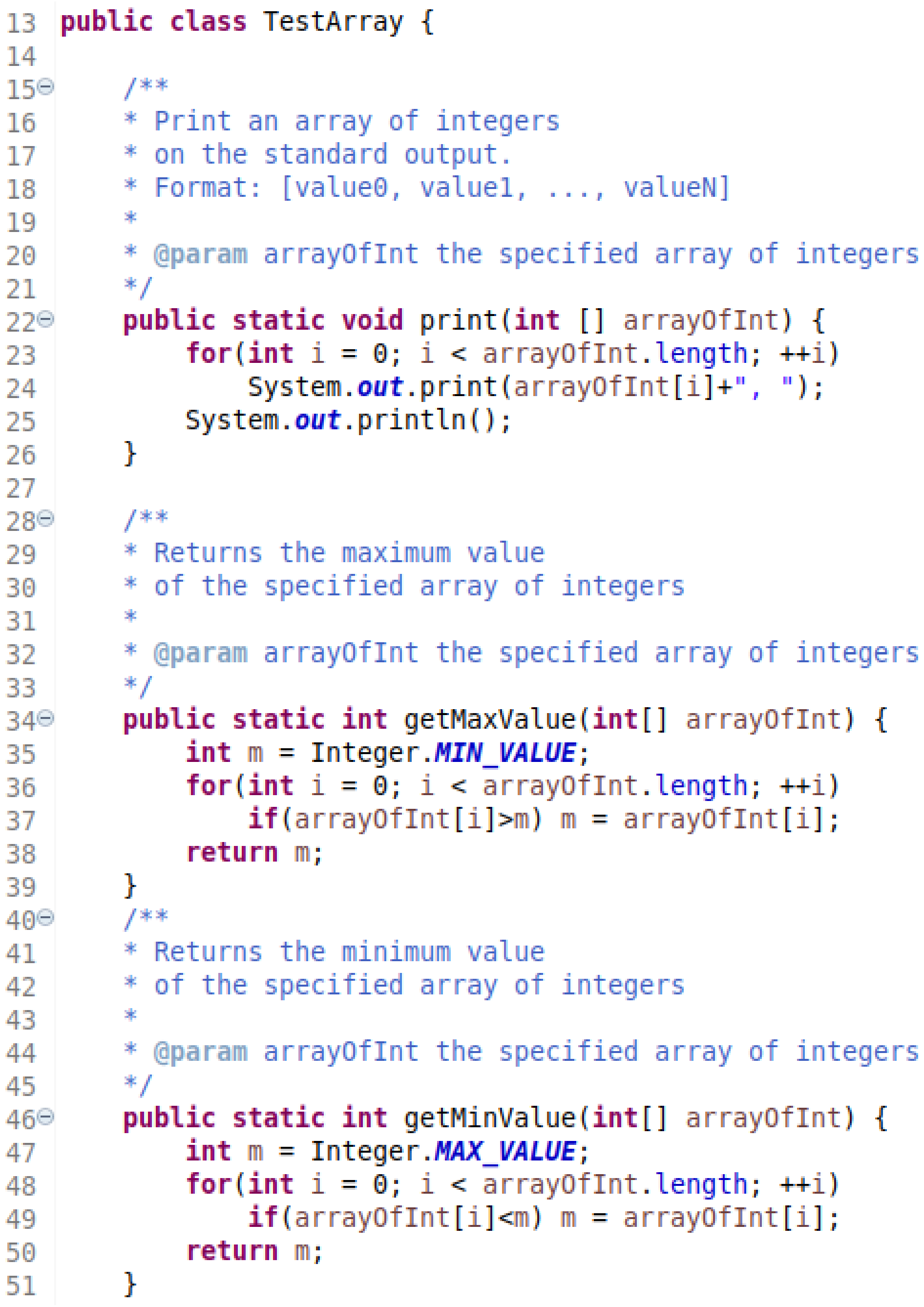
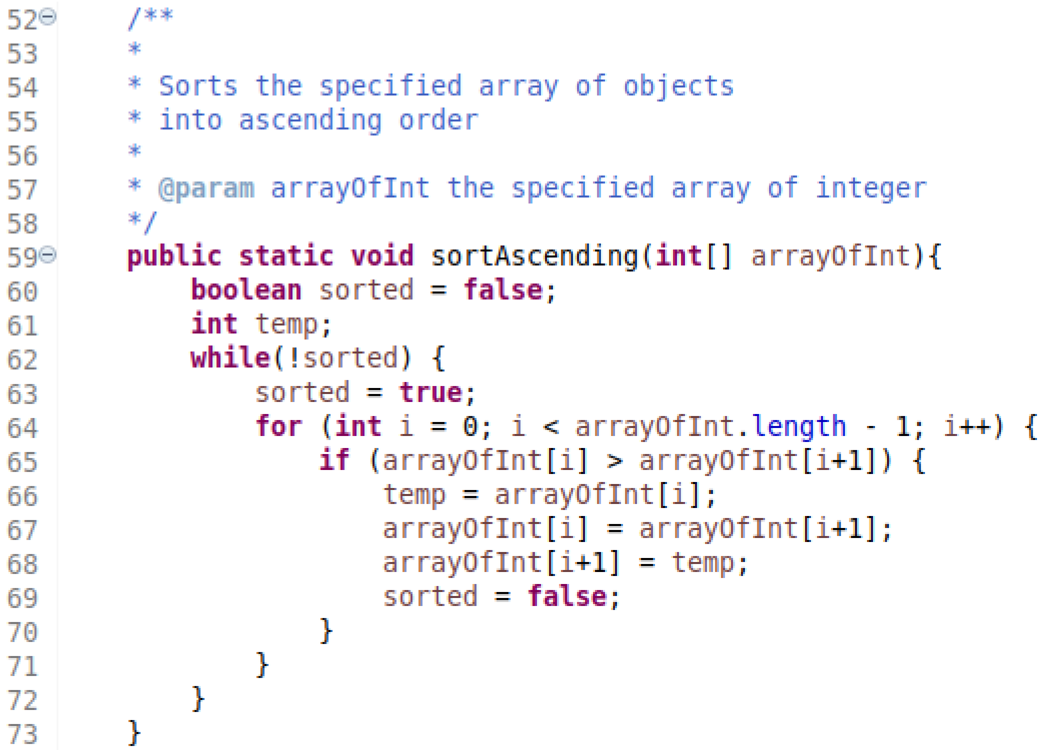
COMPTE RENDU LAB6

TestArrays

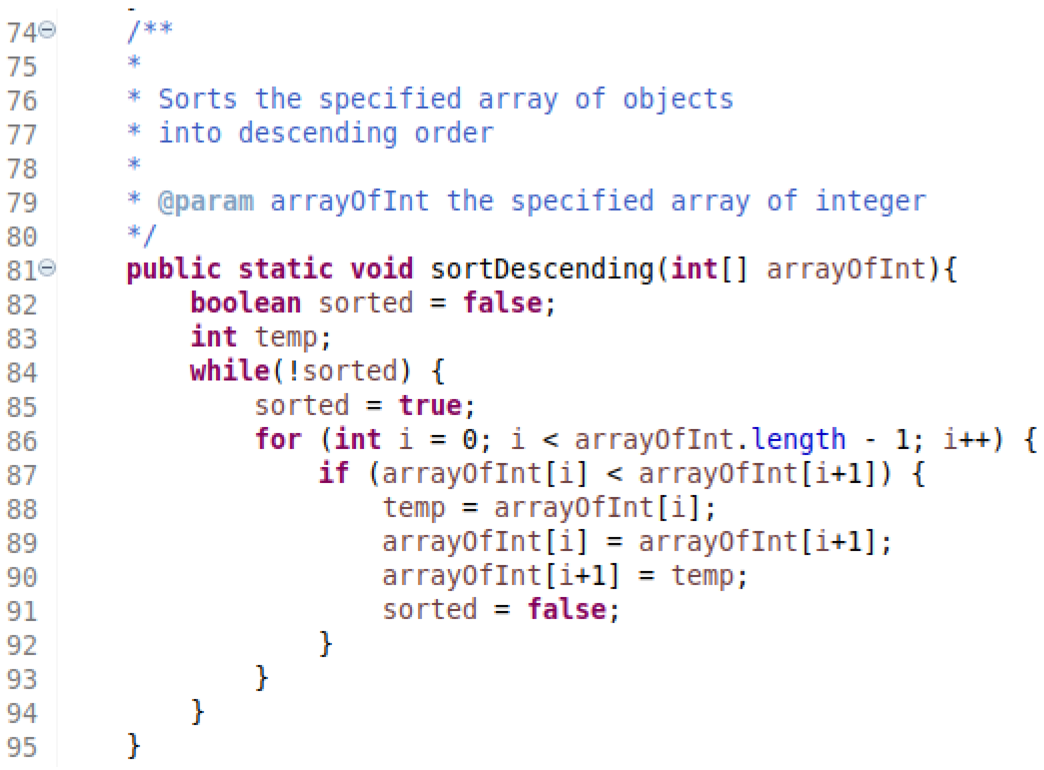
Code



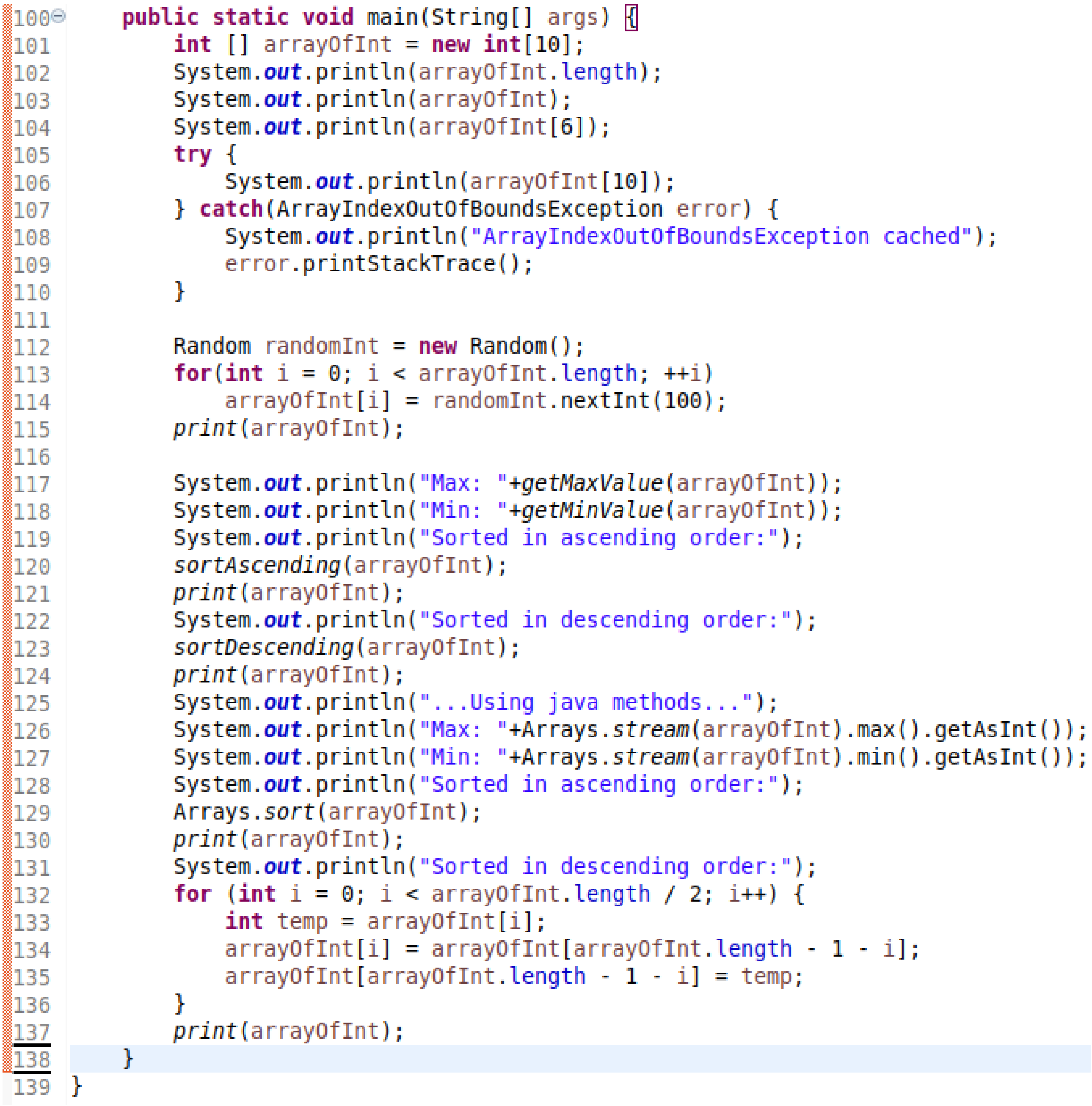
```
public class TestArray {

    /**
     * Print an array of integers
     * on the standard output.
     * Format: [value0, value1, ..., valueN]
     *
     * @param arrayOfInt the specified array of integers
     */
    public static void print(int [] arrayOfInt) {
        for(int i = 0; i < arrayOfInt.length; ++i)
            System.out.print(arrayOfInt[i]+", ");
        System.out.println();
    }

    /**
     * Returns the maximum value
     * of the specified array of integers
     *
     * @param arrayOfInt the specified array of integers
     */
    public static int getMaxValue(int[] arrayOfInt) {
        int m = Integer.MIN_VALUE;
        for(int i = 0; i < arrayOfInt.length; ++i)
            if(arrayOfInt[i]>m) m = arrayOfInt[i];
        return m;
    }
    /**
     * Returns the minimum value
     * of the specified array of integers
     *
     * @param arrayOfInt the specified array of integers
     */
    public static int getMinValue(int[] arrayOfInt) {
        int m = Integer.MAX_VALUE;
        for(int i = 0; i < arrayOfInt.length; ++i)
            if(arrayOfInt[i]<m) m = arrayOfInt[i];
        return m;
    }
```



```java
/**
*
* Sorts the specified array of objects
* into ascending order
*
* @param arrayOfInt the specified array of integer
*/
public static void sortAscending(int[] arrayOfInt){
    boolean sorted = false;
    int temp;
    while(!sorted) {
        sorted = true;
        for (int i = 0; i < arrayOfInt.length - 1; i++) {
            if (arrayOfInt[i] > arrayOfInt[i+1]) {
                temp = arrayOfInt[i];
                arrayOfInt[i] = arrayOfInt[i+1];
                arrayOfInt[i+1] = temp;
                sorted = false;
            }
        }
    }
}
```

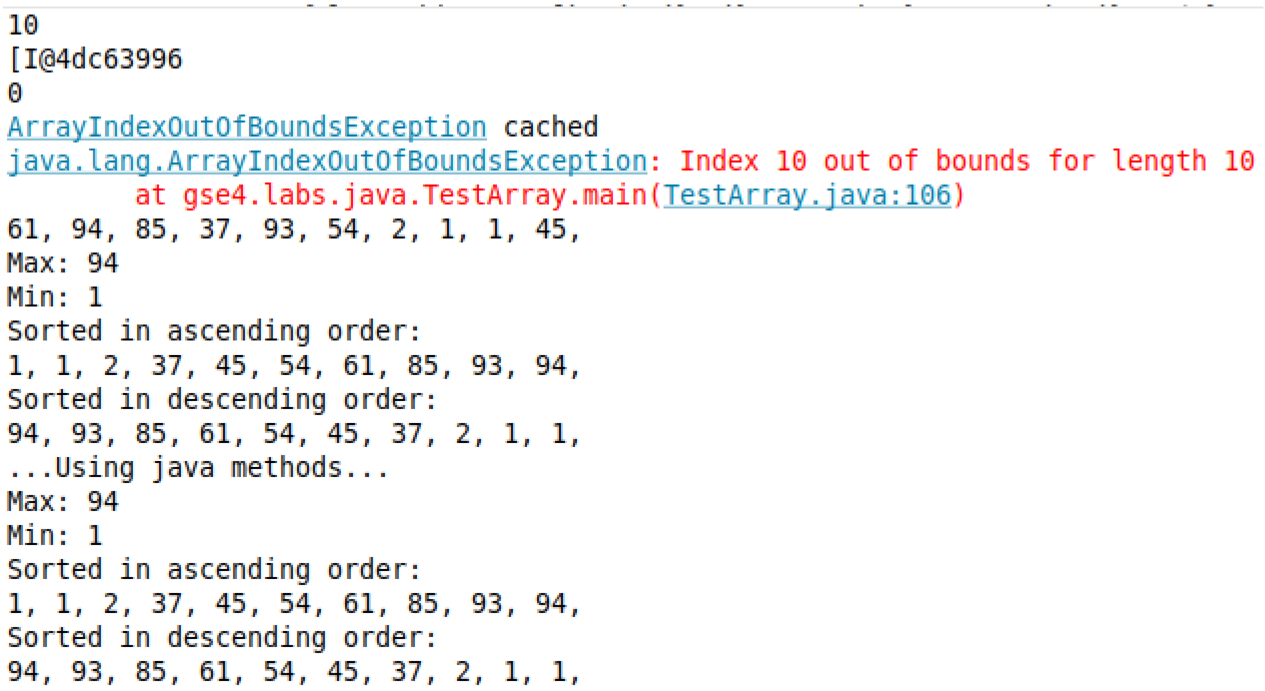


```java
/**
*
* Sorts the specified array of objects
* into descending order
*
* @param arrayOfInt the specified array of integer
*/
public static void sortDescending(int[] arrayOfInt){
    boolean sorted = false;
    int temp;
    while(!sorted) {
        sorted = true;
        for (int i = 0; i < arrayOfInt.length - 1; i++) {
            if (arrayOfInt[i] < arrayOfInt[i+1]) {
                temp = arrayOfInt[i];
                arrayOfInt[i] = arrayOfInt[i+1];
                arrayOfInt[i+1] = temp;
                sorted = false;
            }
        }
    }
}
```



```
    public static void main(String[] args) {
        int [] arrayOfInt = new int[10];
        System.out.println(arrayOfInt.length);
        System.out.println(arrayOfInt);
        System.out.println(arrayOfInt[6]);
        try {
            System.out.println(arrayOfInt[10]);
        } catch(ArrayIndexOutOfBoundsException error) {
            System.out.println("ArrayIndexOutOfBoundsException cached");
            error.printStackTrace();
        }

        Random randomInt = new Random();
        for(int i = 0; i < arrayOfInt.length; ++i)
            arrayOfInt[i] = randomInt.nextInt(100);
        print(arrayOfInt);

        System.out.println("Max: "+getMaxValue(arrayOfInt));
        System.out.println("Min: "+getMinValue(arrayOfInt));
        System.out.println("Sorted in ascending order:");
        sortAscending(arrayOfInt);
        print(arrayOfInt);
        System.out.println("Sorted in descending order:");
        sortDescending(arrayOfInt);
        print(arrayOfInt);
        System.out.println("...Using java methods...");
        System.out.println("Max: "+Arrays.stream(arrayOfInt).max().getAsInt());
        System.out.println("Min: "+Arrays.stream(arrayOfInt).min().getAsInt());
        System.out.println("Sorted in ascending order:");
        Arrays.sort(arrayOfInt);
        print(arrayOfInt);
        System.out.println("Sorted in descending order:");
        for (int i = 0; i < arrayOfInt.length / 2; i++) {
            int temp = arrayOfInt[i];
            arrayOfInt[i] = arrayOfInt[arrayOfInt.length - 1 - i];
            arrayOfInt[arrayOfInt.length - 1 - i] = temp;
        }
        print(arrayOfInt);
    }
}
```

Output

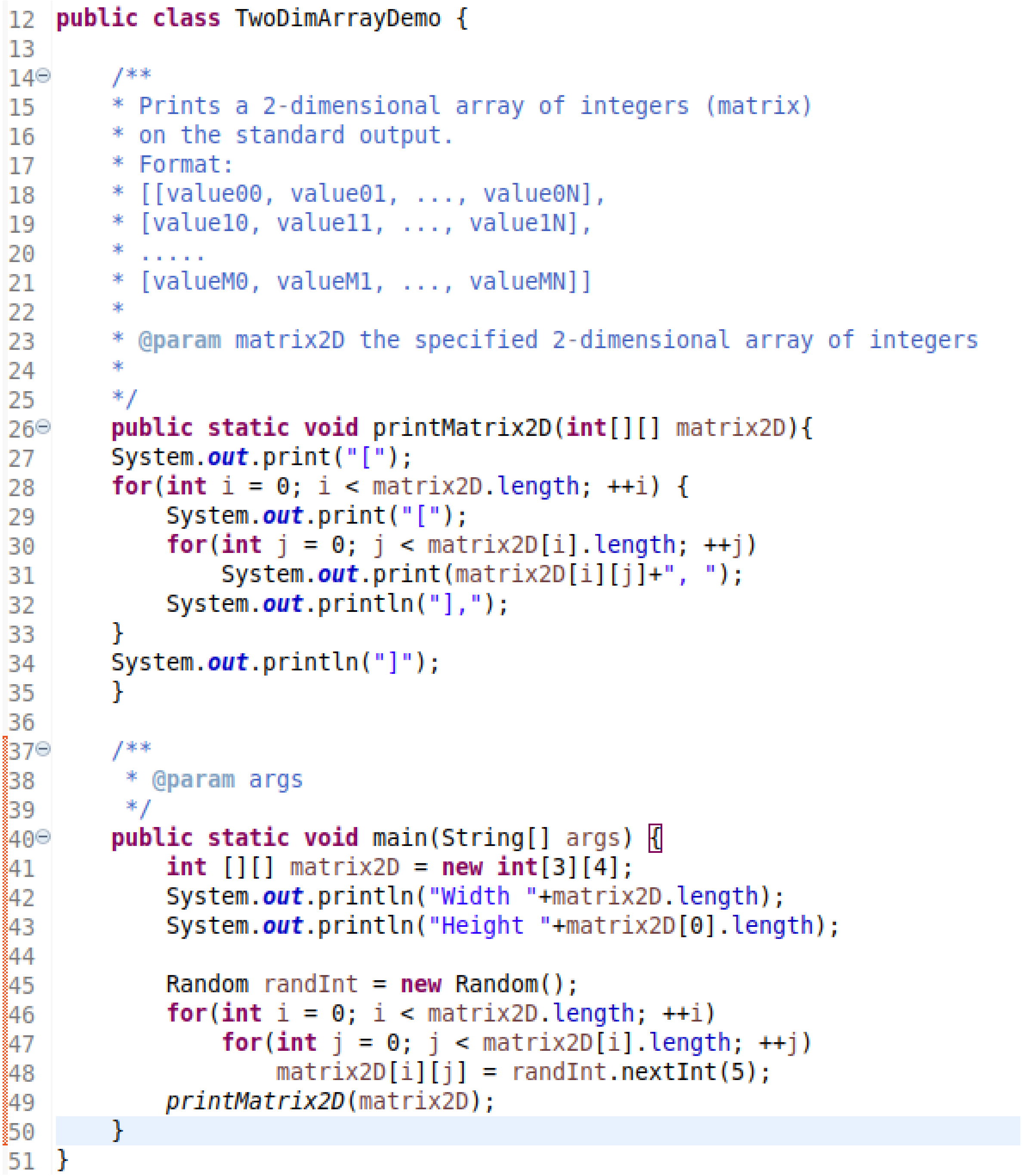


```
10
[I@4dc63996
0
ArrayIndexOutOfBoundsException cached
java.lang.ArrayIndexOutOfBoundsException: Index 10 out of bounds for length 10
        at gse4.labs.java.TestArray.main(TestArray.java:106)
61, 94, 85, 37, 93, 54, 2, 1, 1, 45,
Max: 94
Min: 1
Sorted in ascending order:
1, 1, 2, 37, 45, 54, 61, 85, 93, 94,
Sorted in descending order:
94, 93, 85, 61, 54, 45, 37, 2, 1, 1,
...Using java methods...
Max: 94
Min: 1
Sorted in ascending order:
1, 1, 2, 37, 45, 54, 61, 85, 93, 94,
Sorted in descending order:
94, 93, 85, 61, 54, 45, 37, 2, 1, 1,
```

On remarque que l’utilisation des méthodes Java donne les même résultats. Pour simplifier le code et accélérer le développement, on va donc préférer utiliser ces méthodes.

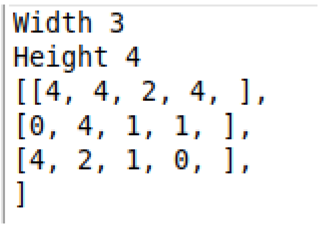
TwoDimArrayDemo

Code



```java
public class TwoDimArrayDemo {

    /**
    * Prints a 2-dimensional array of integers (matrix)
    * on the standard output.
    * Format:
    * [[value00, value01, ..., value0N],
    * [value10, value11, ..., value1N],
    * .....
    * [valueM0, valueM1, ..., valueMN]]
    *
    * @param matrix2D the specified 2-dimensional array of integers
    *
    */
    public static void printMatrix2D(int[][] matrix2D){
    System.out.print("[");
    for(int i = 0; i < matrix2D.length; ++i) {
        System.out.print("[");
        for(int j = 0; j < matrix2D[i].length; ++j)
            System.out.print(matrix2D[i][j]+", ");
        System.out.println("],");
    }
    System.out.println("]");
    }

    /**
     * @param args
     */
    public static void main(String[] args) {
        int [][] matrix2D = new int[3][4];
        System.out.println("Width "+matrix2D.length);
        System.out.println("Height "+matrix2D[0].length);

        Random randInt = new Random();
        for(int i = 0; i < matrix2D.length; ++i)
            for(int j = 0; j < matrix2D[i].length; ++j)
                matrix2D[i][j] = randInt.nextInt(5);
        printMatrix2D(matrix2D);
    }
}
```

Output



```
Width 3
Height 4
[[4, 4, 2, 4, ],
[0, 4, 1, 1, ],
[4, 2, 1, 0, ],
]
```