

3주차 실습 보고서



강의명	객체지향프로그래밍및실습
담당교수	류기열
학과 학년	소프트웨어학과 2학년
학번	202220209
작성자	이육준

2025. 9.18.

□ 1

```
J LotteryDrawing_q3.java 1 J CompoundInterest.java J Stack.java X
JAVA > 3 > J Stack.java > Stack > push(int)
1 public class Stack
2 {
3     private static final int MAX = 6;
4     private static int top=0;
5     private static int[] s = new int[MAX];
6     public static int pop()
```

```
J LotteryDrawing.java 1 X J Stack.java
JAVA > 3 > J LotteryDrawing.java > LotteryDrawing > main(String[])
7 public class LotteryDrawing
9     public static void main(String[] args)
22         numbers[i] = 1 + i;
23
24         // draw k numbers and put them into a second array
25         int[] result = new int[k];
26
27         for (int i = 0; i < result.length; i++)
28         {
29             // make a random index between 0 and n - 1
30             int r = (int) (Math.random() * n);
31
32             // pick the element at the random location
33             result[i] = numbers[r];
34             Stack.push(numbers[r]);
35
36         }
37
38         System.out.println(x:"Bet the following combination. It'll make you rich!");
39         for (int r : result){
40             System.out.print(r + " ");
41         }
42         System.out.println();
43         for (int i =0;i<k;i++){
44             System.out.print(Stack.pop() + " ");
45         }
46         System.out.println();

```

PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
• lwj@lwj-code:~/workspace/JAVA/3$ java LotteryDrawing
How many numbers do you need to draw? 6
What is the highest number you can draw? 45
Bet the following combination. It'll make you rich!
6 45 21 7 32 30
30 32 7 21 45 6
○ lwj@lwj-code:~/workspace/JAVA/3$
```

```
• lwj@lwj-code:~/workspace/JAVA/3$ java LotteryDrawing
How many numbers do you need to draw? 6
What is the highest number you can draw? 44
Bet the following combination. It'll make you rich!
15 15 19 2 42 26
26 42 2 19 15 15
```

```
• lwj@lwj-code:~/workspace/JAVA/3$ java LotteryDrawing
How many numbers do you need to draw? 6
What is the highest number you can draw? 43
Bet the following combination. It'll make you rich!
13 6 14 28 1 30
30 1 28 14 6 13
```

□ 2

LotteryDrawing.java 1Stack.javaRandomTest.java 2 X

JAVA > 3 > RandomTest.java > RandomTest > random6(int[], int, int)

```
1  import java.math.*;
2
3  public class RandomTest
4  {  public static void main(String[] args)
5      {
6          int n = 45;
7          int[] result = new int[6];
8          random6(result, m:1, n:45);
9          for(int e: result)
10             System.out.printf(format:"%d ", e);
11             System.out.println();
12         }
13     public static void random6(int[] result, int m, int n){
14         int r=0;
15         boolean v=false;
16         for(int i=0;i<result.length;i++){
17             do{
18                 v = false;
19                 r = (int) (Math.random()*45+1);
20                 for(int j=0;j<=i;j++){
21                     if(result[j]==r)
22                         v=true;
23                 }
24             } while(v);
25             result[i] = r;
26         }
27     }
```

PROBLEMS 7 OUTPUT DEBUG CONSOLE TERMINAL PORTS

- lwj@lwj-code:~/workspace/JAVA/3\$ java RandomTest
26 20 10 4 3 7
- lwj@lwj-code:~/workspace/JAVA/3\$ java RandomTest
41 43 42 16 9 28
- lwj@lwj-code:~/workspace/JAVA/3\$ java RandomTest
43 1 17 18 42 4
- lwj@lwj-code:~/workspace/JAVA/3\$

□ 3

J LotteryDrawing_q3.java 1 XJ Stack.javaJ RandomTest.java 2

JAVA > 3 > J LotteryDrawing_q3.java > LotteryDrawing_q3 > main(String[])

```
6  import java.math.*;
7
8  public class LotteryDrawing_q3
9  {
10     public static void main(String[] args)
11     {
12         final int n =45;
13         final int k = 6;
14
15         // fill an array with numbers 1 2 3 . . . n
16         int[] numbers = new int[n];
17         for (int i = 0; i < numbers.length; i++)
18             numbers[i] = i + 1;
19
20         // draw k numbers and put them into a second array
21         for (int i = 0; i < k; i++)
22         {
23             int r = (int) (Math.random()*(45-i)+i);
24             int tmp = numbers[i];
25             numbers[i] = numbers[r];
26             numbers[r] = tmp;
27         }
28         //print
29         for(int i = 0;i<k;i++)
30             System.out.print(numbers[i] + " ");
31         System.out.println();
32     }
33 }
```

PROBLEMS 8 OUTPUT DEBUG CONSOLE TERMINAL PORTS

- lwj@lwj-code:~/workspace/JAVA/3\$ java LotteryDrawing_q3
42 10 3 9 24 5
- lwj@lwj-code:~/workspace/JAVA/3\$ java LotteryDrawing_q3
33 2 15 16 13 28
- lwj@lwj-code:~/workspace/JAVA/3\$ java LotteryDrawing_q3
42 1 6 32 26 3

□ 4

```

J LotteryDrawing_q3.java 1    J CompoundInterest.java X    J Stack.java    J RandomTest.java 2
JAVA > 3 > J CompoundInterest.java > CompoundInterest
1 public class CompoundInterest {
2     // static final 변수들을 클래스 바로 아래에 선언
3     static final double STARTRATE = 10;
4     static final int NRATES = 6;
5     static final int NYEARS = 10;
6
7     public static void main(String[] args) {
8         double[] interestRate = new double[NRATES];
9         double[][] balances = new double[NYEARS][NRATES];
10
11         initialize(interestRate, balances); // 이중배열 초기값 설정
12         computeInterest(interestRate, balances); // 이율 계산 후 잔고 배열에 저장
13         printResult(interestRate, balances); // 이중배열 출력
14     }
15
16     public static void initialize(double[] interestRate, double[][] balances) {
17
18         // set interest rates to 10 . . . 15%
19         for(int i = 0; i < NRATES; i++){
20             interestRate[i] = (STARTRATE+i)/100.0;
21         }
22         // set initial balances to 10000
23         for(int i = 0; i < balances[0].length; i++){
24             balances[0][i] = 10000;
25         }
26     }
27
28     public static void computeInterest(double[] interestRate, double[][] balances) {
29         // compute interest for future years
30         for(int i = 1; i < balances.length; i++){
31             for(int j = 0; j < balances[i].length; j++){
32                 // get last year's balances from previous row
33                 double oldBalance = balances[i-1][j];
34                 // compute interest
35                 double interest = oldBalance * interestRate[j];
36                 // compute this year's balances
37                 balances[i][j] = oldBalance + interest;
38             }
39         }
40     }
41
42     public static void printResult(double[] interestRate, double[][] balances) {
43         // print one row of interest rates
44         for(double i : interestRate){
45             System.out.print(i + " ");
46         }
47         System.out.println();
48         // print balance table
49         for(int i = 0; i < balances.length; i++){
50             // print table row
51             for(int j = 0; j < balances[i].length; j++){
52                 System.out.printf(format: "%.2f ", balances[i][j]);
53             }
54             System.out.println();
55         }
56     }
57 }

```

```
● lwj@lwj-code:~/workspace/JAVA/3$ java CompoundInterest
    10%    11%    12%    13%    14%    15%
10000.00 10000.00 10000.00 10000.00 10000.00 10000.00
11000.00 11100.00 11200.00 11300.00 11400.00 11500.00
12100.00 12321.00 12544.00 12769.00 12996.00 13225.00
13310.00 13676.31 14049.28 14428.97 14815.44 15208.75
14641.00 15180.70 15735.19 16304.74 16889.60 17490.06
16105.10 16850.58 17623.42 18424.35 19254.15 20113.57
17715.61 18704.15 19738.23 20819.52 21949.73 23130.61
19487.17 20761.60 22106.81 23526.05 25022.69 26600.20
```

□ 코드 전문

LotteryDrawing.java

```
import java.util.Scanner;
/**
 * This program demonstrates array manipulation.
 * @version 1.20 2004-02-10
 * @author Cay Horstmann
 */
public class LotteryDrawing
{
    public static void main(String[] args)
    {
        Scanner in = new Scanner(System.in);

        System.out.print("How many numbers do you need to draw? ");
        int k = in.nextInt();

        System.out.print("What is the highest number you can draw? ");
        int n = in.nextInt();

        // fill an array with numbers 1 2 3 . . . n
        int[] numbers = new int[n];
        for (int i = 0; i < numbers.length; i++)
            numbers[i] = i + 1;

        // draw k numbers and put them into a second array
        int[] result = new int[k];

        for (int i = 0; i < result.length; i++){
            // make a random index between 0 and n - 1
            int r = (int) (Math.random() * n);

            // pick the element at the random location
            result[i] = numbers[r];
            Stack.push(numbers[r]);
        }

        System.out.println("Bet the following combination. It'll make you rich!");
        for (int r : result){
            System.out.print(r + " ");
        }
        System.out.println();
        for (int i = 0; i < k; i++){
            System.out.print(Stack.pop() + " ");
        }
    }
}
```



```

    }
    System.out.println();
}
}

```

Stack.java

```

public class Stack
{
    private static final int MAX = 6;
    private static int top=0;
    private static int[] s = new int[MAX];
    public static int pop()
    {
        if (top == 0) {
            System.out.println("Empty!");
            System.exit(-1);
        } else {
            top--;
            return s[top];
        }
    }
    public static void push(int x)
    {
        if(top==MAX) {
            System.out.println("Full!");
            System.exit(-1);
        } else {
            s[top] = x;
            top++;
        }
        return;
    }
}

```

RandomTest.java

```

import java.math.*;

public class RandomTest
{
    public static void main(String[] args)
    {
        int n = 45;
        int[] result = new int[6];
        random6(result, 1, 45);
    }
}

```

```

        for(int e: result)
            System.out.printf("%d ", e);
        System.out.println();
    }
    public static void random6(int[] result, int m, int n){
        int r=0;
        boolean v=false;
        for(int i=0;i<result.length;i++){
            do{
                v = false;
                r = (int) (Math.random()*45+1);
                for(int j=0;j<=i;j++){
                    if(result[j]==r)
                        v=true;
                }
            } while(v);
            result[i] = r;
        }
    }
}

```

LotteryDrawing_q3.java

```

/**
 * This program demonstrates array manipulation.
 * @version 1.20 2004-02-10
 * @author Cay Horstmann
 */
import java.math.*;

public class LotteryDrawing_q3
{
    public static void main(String[] args)
    {
        final int n =45;
        final int k = 6;

        // fill an array with numbers 1 2 3 . . . n
        int[] numbers = new int[n];
        for (int i = 0; i < numbers.length; i++)
            numbers[i] = i + 1;

        // draw k numbers and put them into a second array
        for (int i = 0; i < k; i++)
        {

```

```

        int r = (int) (Math.random()*(45-i)+i);
        int tmp = numbers[i];
        numbers[i] = numbers[r];
        numbers[r] = tmp;
    }
    //print
    for(int i = 0;i<k;i++)
        System.out.print(numbers[i] + " ");
    System.out.println();
}
}

```

CompoundInterest.java

```

public class CompoundInterest {
    // static final 변수들을 클래스 바로 아래에 선언
    static final double STARTRATE = 10;
    static final int NRATES = 6;
    static final int NYEARS = 10;

    public static void main(String[] args) {
        double[] interestRate = new double[NRATES];
        double[][] balances = new double[NYEARS][NRATES];

        initialize(interestRate, balances); // 이중배열 초기값 설정
        computeInterest(interestRate, balances); // 이율 계산 후 잔고 배열에 저장
        printResult(interestRate, balances); // 이중배열 출력
    }

    public static void initialize(double[] interestRate, double[][] balances) {

        // set interest rates to 10 . . . 15%
        for(int i = 0; i < NRATES; i++){
            interestRate[i] = (STARTRATE+i)/100.0;
        }
        // set initial balances to 10000
        for(int i = 0; i < balances[0].length; i++){
            balances[0][i] = 10000;
        }
    }

    public static void computeInterest(double[] interestRate, double[][]
balances) {
        // compute interest for future years
        for(int i = 1; i < balances.length; i++){
            for(int j = 0; j < balances[i].length; j++){

```

```

        // get last year's balances from previous row
        double oldBalance = balances[i-1][j];
        // compute interest
        double interest = oldBalance * interestRate[j];
        // compute this year's balances
        balances[i][j] = oldBalance + interest;
    }
}

public static void printResult(double[] interestRate, double[][] balances)
{
    // print one row of interest rates
    for(double i : interestRate){
        System.out.print("    " + (int)(i*100) + "% ");
    }
    System.out.println();
    // print balance table
    for(int i = 0; i < balances.length; i++){
        // print table row
        for(int j = 0; j < balances[i].length; j++){
            System.out.printf("%.2f ", balances[i][j]);
        }
        System.out.println();
    }
}
}

```