

객체지향프로그래밍및실습 - 3주차 실습활동지(9월18일)

성명: _____ 학과: _____ 학번: _____ 실습반: _____

제출방식: 보고서(파일명: report_4)를 제출한다. 보고서는 pdf 형식으로 저장한다.

제출기한: 9월 19일(금) 23시59분까지 (지연 제출 0점 처리함, 미완성이라도 반드시 제출하길 바람)

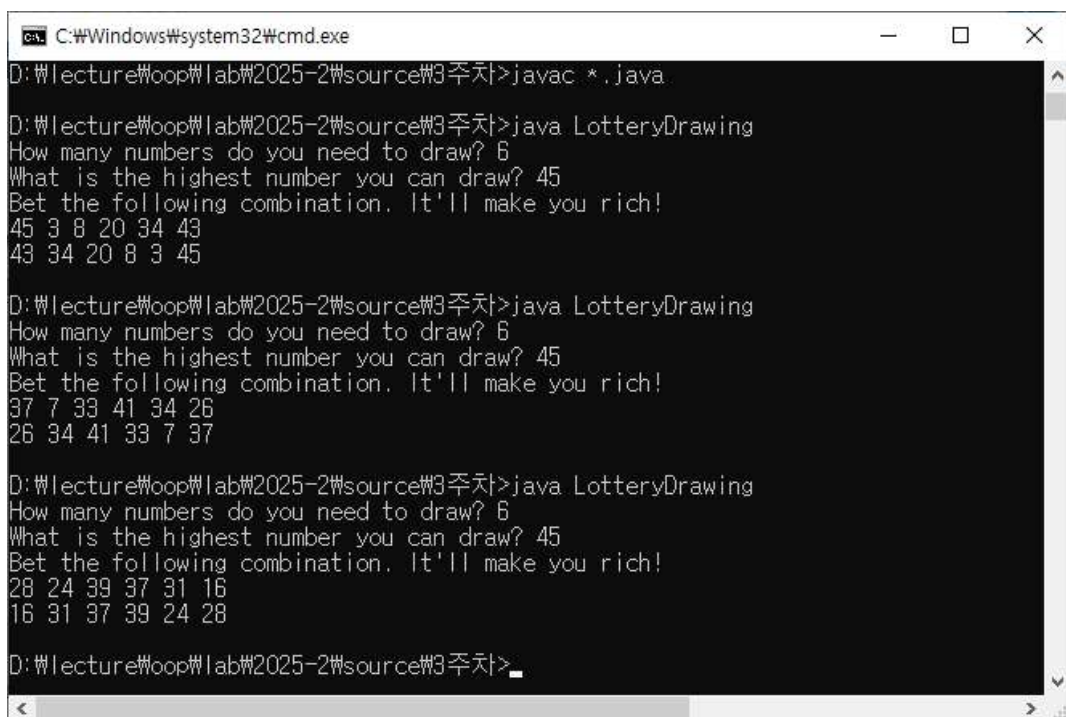
I. Objectives

1. Java 언어의 static method로 구성된 프로그램의 구조에 대해 학습한다.
2. Java 언어의 배열의 정의 및 활용법을 학습한다.
3. Java 언어에서의 매개 변수 전송에 대해 학습한다.

II. Exercises (15점)

1. Stack 클래스(Stack.java, 강의노트 Fundamental Programming Structures - Part 2 참조)를 이용하여 LotteryDrawing 클래스(첨부파일, LotteryDrawing.java)를 다음과 같이 수정하고자 한다.
 - 뽑은 결과를 result 배열 뿐만아니라 stack에도 저장한다.
 - result 배열을 정렬하지 않고 출력한 후 stack에 저장된 값들을 pop하여 순서대로 출력한다.
 - k값은 0보다 크고 7보다 작다고 가정하고 Stack 클래스의 MAX 값은 6으로 설정한다.

수정한 후 3회 실행해 보시오. (실행예시 참조) [수정한 코드 및 실행결과][3점]



```
C:\Windows\system32\cmd.exe
D:\lecture\oop\lab\2025-2\source\3주차>javac *.java
D:\lecture\oop\lab\2025-2\source\3주차>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!
45 3 8 20 34 43
43 34 20 8 3 45

D:\lecture\oop\lab\2025-2\source\3주차>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!
37 7 33 41 34 26
26 34 41 33 7 37

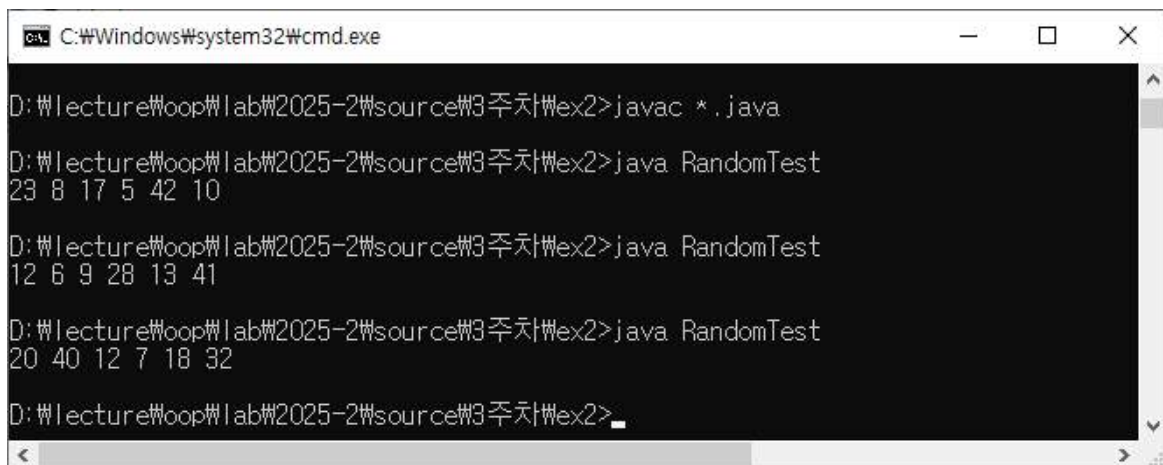
D:\lecture\oop\lab\2025-2\source\3주차>java LotteryDrawing
How many numbers do you need to draw? 6
What is the highest number you can draw? 45
Bet the following combination. It'll make you rich!
28 24 39 37 31 16
16 31 37 39 24 28

D:\lecture\oop\lab\2025-2\source\3주차>
```

2. 아래 코드는 1~45 사이의 임의의 수를 6개 선택하여 출력하는 프로그램이다(실행 예시 참조). $m \sim n$ 사이(m, n 포함, $n \geq m+6$)의 임의의 중복되지 않는 정수값 6개를 반환하는 메소드 `random6()`을 작성하시오. 단, `Math.random()` 메소드를 사용할 것. [메소드 코드 및 실행결과][4점]

```
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)
    {
        // 해당 method 작성
    }
}
```

실행예시:



```
C:\Windows\system32\cmd.exe
D:\lecture\oop\lab\2025-2\source\3주차\ex2>javac *.java
D:\lecture\oop\lab\2025-2\source\3주차\ex2>java RandomTest
23 8 17 5 42 10
D:\lecture\oop\lab\2025-2\source\3주차\ex2>java RandomTest
12 6 9 28 13 41
D:\lecture\oop\lab\2025-2\source\3주차\ex2>java RandomTest
20 40 12 7 18 32
D:\lecture\oop\lab\2025-2\source\3주차\ex2>_
```

3. 강의노트 Fundamental Programming Structures -Part 3에 나오는 LotteryDrawing 클래스를 다음과 같이 수정하고자 한다.

- result 배열을 사용하지 않고 뽑은 수를 numbers의 맨 앞부분 k개의 원소에 순서대로 저장한다.
- 뽑힌 수는 다음 턴에서 뽑히지 않아야 하며, 뽑히지 않은 수는 다음 턴에서 동일한 확률로 뽑힐 수 있어야 한다.
- 뽑은 k개의 수는 정렬하지 않고 순서대로 출력한다.
- n = 45, k = 6으로 고정한다.

아래 코드에 빈 곳 (가~나)를 적절하게 채운 후 3회 실행해 보시오. [수정된 method 코드 및 실행결과] [5점]

힌트) 2번에서 작성한 수식을 이용한다.

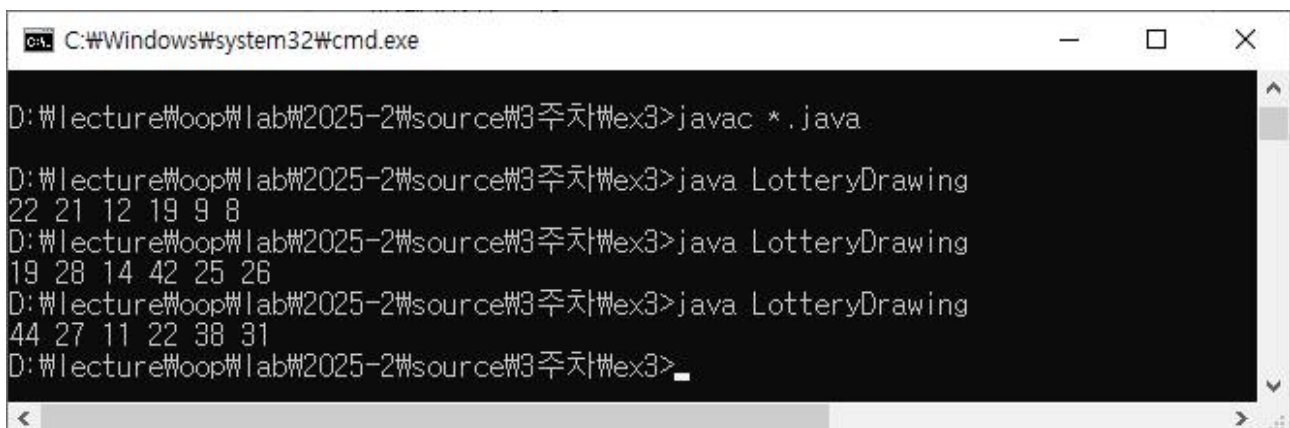
```
public static void main(String[] args)
{
    final int n = 45;
    final int k = 6;

    // 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;

    for (int i = 0; i < k; i++)
    {
        // (가) i에서 n-1 사이의 임의의 수를 뽑아 그 위치의 원소와 i번째 원소와 서로 바꾼다.

    }

    // (나) 뽑은 k개의 값을 출력한다.
}
```



```
C:\Windows\system32\cmd.exe
D:\lecture\woop\lab\2025-2\source\주차\ex3>javac *.java
D:\lecture\woop\lab\2025-2\source\주차\ex3>java LotteryDrawing
22 21 12 19 9 8
D:\lecture\woop\lab\2025-2\source\주차\ex3>java LotteryDrawing
19 28 14 42 25 26
D:\lecture\woop\lab\2025-2\source\주차\ex3>java LotteryDrawing
44 27 11 22 38 31
D:\lecture\woop\lab\2025-2\source\주차\ex3>
```

4. CompoundInterest 클래스 (첨부파일 CompoundInterest.java)를 아래와 같은 구조로 수정하고자 한다. 물음에 답하시오.

```
public class CompoundInterest
{
    public static void main(String[] args)
    {
        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);
            computeInterest(interestRate, balances);
            printResult(interestRate, balances);
        }
        public static void initialize(...) {
            //해당 메소드 작성
        }
        public static void computeInterest(...) {
            //해당 메소드 작성
        }
        public static void printResult(...) {
            //해당 메소드 작성
        }
    }
}
```