단국대학교 응용컴퓨터공학 JAVA 프로그래밍 I 중간고사 (2018년 봄학기) 2018년 4월 23일

학과 _____ 학번 _____ 이름 _____

중간고사

담당교수: 단국대학교 응용컴퓨터공학 박경신

- **답은 반드시 답안지에 기술할 것. 공간이 부족할 경우 반드시 답안지 몇 쪽의 뒤에 있다고 명기한 후 기술할 것. 그 외의 경우의 답안지 뒤쪽이나 연습지에 기술한 내용은 답안으로 인정 안 함. 답에는 반드시 네모를 쳐서 확실히 표시할 것.**
- **답안지에 학과, 학번, 이름 외에 본인의 암호(4자리 숫자)를 기입하면 성적공고시 학번대신 암호를 사용할 것임.**

1. 아래의 코드를 보고 빈 칸을 채워라. (30점)

```
public enum TripleOperator {
	MIN(1), MAX(2), MEDIAN(3), MEAN(4), SUM(5), RANGE(6);
	private final int name;
	TripleOperator(int name) {
		this.name = name;
	}
	public int getOperator() {
		return name;
	}
	public static TripleOperator nameOf(int value) { // (1.1) value에따라 TripleOperator반환
		switch (value) {
		case 1:
			return MIN;
		case 2:
			return MAX;
		case 3:
			return MEDIAN;
		case 4:
			return MEAN;
		case 5:
			return SUM;
		case 6:
			return RANGE;
		default:
			return null;
		}
	}
```

```
        public double calculate(int x, int y, int z) {
                switch (this) {
                case MIN:       return Math.min(Math.min(x, y), z);
                case MAX:       return Math.max(Math.max(x, y), z);
                case MEDIAN:    return Math.max(Math.min(x,y), Math.min(Math.max(x,y),z));
                case MEAN:      return (x + y + z) / 3.0;
                case SUM:       return x + y + z;
                case RANGE: return Math.max(Math.max(x, y), z) - Math.min(Math.min(x, y), z);
                default: throw new AssertionError("Unknown operations " + this);
                }
        }
}
 public class TripleValue {
        private int x; // first number
        private int y; // second number
        private int z; // third number
        public TripleValue() { // constructor
                this(0, 0, 0);
        }
        public TripleValue(TripleValue other) { // constructor
                set(other.x, other.y, other.z);
        }
        public TripleValue(int x, int y, int z) { // constructor
                set(x, y, z);
        }
        public void set(int x, int y, int z) { // set
                this.x = x;     this.y = y;     this.z = z;
        }
        public int getX() { // get
                return x;
        }
        public int getY() { // get
                return y;
        }
        public int getZ() { // get
                return z;
        }
```

```
        @Override // toString method override of Object class
        public String toString() {
                return "[" + getX() + ", " + getY() + ", " + getZ() + "]";
        }
}
public class TripleCalculator {
        private TripleValue value; // x,y,z value
        private TripleOperator op; // operator
        private double w; // result
        public TripleCalculator() { // (1.2) constructor
                this(0, 0, 0, TripleOperator.MIN);
        }
        public TripleCalculator(TripleCalculator other) {// (1.2) constructor
                set(other.getValue(), other.getOp());
        }
        public TripleCalculator(int x, int y, int z, TripleOperator op) {// (1.2) constructor
                set(x, y, z, op);
        }
        public TripleCalculator(int[] num, TripleOperator op) { // (1.2) constructor
                set(num[0], num[1], num[2], op);
        }
        public TripleCalculator(TripleValue value, TripleOperator op) { // (1.2) constructor
                set(value, op);
        }
        public void set(int x, int y, int z, TripleOperator op) { // set
                this.value = new TripleValue(x, y, z);
                this.op = op;
        }
        public void set(TripleValue value, TripleOperator op) { // set
                this.value = value;
                this.op = op;
        }
        public TripleValue getValue() { // (1.3) getValue
                return value;
        }
        public TripleOperator getOp() { // (1.3) getOp
                return op;
```

학과 _____ 학번 _____ 이름 _____

```
        }
        public double getW() {
                return op.calculate(value.getX(), value.getY(), value.getZ());// (1.3) getW
        }
        @Override // toString method override of Object class
        public String toString() {
                return "" + getOp() + value + "=" + getW();
        }
}
public class UserInput {
        private static Scanner scan = new Scanner(System.in); // Scanner (2.7)
        private UserInput() {} // constructor (2.7)
        public static int getInteger(String description) { /* 내부구현 생략 */ }
        public static int getIntegerBetween(String description, int min, int max) { /*생략*/ }
        public static boolean getExitKey() { /* 내부구현 생략 */ }
        public static TripleOperator getTripleOperator(String description) {
                int value = getIntegerBetween(description, 1, 6);
                return TripleOperator.nameOf(value); // (1.3)
        }
        public static TripleCalculator getTripleCalculator() {
                System.out.println("Triple calculator using user input");
                int x = getInteger("Please enter the first number: ");
                int y = getInteger("Please enter the second number: ");
                int z = getInteger("Please enter the third number: ");
                TripleOperator op = getTripleOperator("Please enter the operator [1.MIN,
2.MAX, 3.MEDIAN, 4.MEAN, 5.SUM, 6.RANGE]: ");
                return new TripleCalculator(x, y, z, op); // (1.3)
        }
}
```

2. 다음 질문에 답하라 (70점)

```
public class TripleCalculatorTest {
        static void print(int[] arr) {
                System.out.println(java.util.Arrays.toString(arr));
        }
        static void print(int[][] array) {
                System.out.println(java.util.Arrays.deepToString(array));
        }
```

학과 _____ 학번 _____ 이름 _____

```
static void print(TripleValue[] array) {
        System.out.println(java.util.Arrays.deepToString(array));
}
static TripleValue fill(int[] arr) {
        return new TripleValue(arr[0], arr[1], arr[2]);
}
static TripleValue[] fill(int[][] array) {
        TripleValue[] values = new TripleValue[array.length];
        for (int i = 0; i < array.length; i++) {
                values[i] = new TripleValue(array[i][0], array[i][1], array[i][2]);
        }
        return values;
}
static void square(int a) {
        a *= a;
        System.out.println("inside square a=" + a);
}
static void square1(int[] arr) {
        for (int a : arr) {
                a *= a;
                System.out.println("inside square1 a=" + a);
        }
}
static void square2(int[] arr) {
        for (int i = 0; i < arr.length; i++) {
                arr[i] *= arr[i];
                System.out.println("inside square2 a=" + arr[i]);
        }
}
public static void main(String[] args) {
        // (2.1) TripleValue
        TripleValue value1 = new TripleValue(1, 2, 3);
        System.out.println("value1=" + value1);
        TripleValue value2 = new TripleValue(value1);
        System.out.println("value2=" + value2);
        TripleValue value3 = new TripleValue();
        System.out.println("value3=" + value3);
```

```
TripleValue value4 = value1;
System.out.println("value4=" + value4);
value4.set(-1,-2,-3);
System.out.println("value1="    + value1);
System.out.println("value2=" + value2);
System.out.println("value3=" + value3);
System.out.println("value4=" + value4);
// (2.2) fill method overloading
int[] arr1 = {1, 2, 3};
int[] arr2 = {-1, -2, -3, -4};
int[][] arr3 = {{4,-3,5}, {-4,3,7}, {7,-2,4}};
print(arr1); // [1, 2, 3]
print(arr2); // [-1, -2, -3, -4]
print(arr3); // [[4,-3,5], [-4,3,7], [7,-2,4]]
TripleValue fill1 = fill(arr1);
System.out.println("fill1=" + fill1);
TripleValue fill2 = fill(arr2);
System.out.println("fill2=" + fill2);
TripleValue[] fill3 = fill(arr3);
System.out.print("fill3=");
print(fill3);
// (2.3) square/square1/square2 method parameter passing
int a = -1;
square(a);
System.out.println("after square a=" + a);
Square1(arr1);
System.out.print("after square1 arr1=");
print(arr1);
square2(arr2);
System.out.print("after square2 arr2=");
print(arr2);
// (2.4) TripleCalculator
for (TripleOperator op : TripleOperator.values()) {
        TripleCalculator triple = new TripleCalculator(10, 20, -30, op);
        System.out.println(triple);
}
```

학과 _____ 학번 _____ 이름 _____

```
        // (2.5) nested for-loop using TripleValue[] & TripleOperator
        for (TripleValue num : fill3) {
                for (TripleOperator op : TripleOperator.values()) {
                        TripleCalculator triple = new TripleCalculator(num, op);
                        System.out.printf("%s%s=%f\n", op, num, triple.getW());
                }
        }
        // (2.6) user input
        do {
                TripleCalculator triple = UserInput.getTripleCalculator();
                System.out.println(triple);
        } while(!UserInput.getExitKey());
    }
}
```

2.1 main()에 (2.1)는 TripleValue 생성자를 이용한 객체의 생성을 보여주고 있다. 실행결과를 적고 그 이유를 설명하라 (예를 들어, 어떤 TripleValue 생성자를 사용했는지). (10점)

value1=[1, 2, 3] // TripleValue(int x, int y, int z) 사용
value2=[1, 2, 3] // TripleValue(TripleValue other) 사용
value3=[0, 0, 0] // TripleValue() 사용
value4=[1, 2, 3] // = 사용 value4는 value1를 가리키고 있다
value1=[-1, -2, -3] //value4.set(-1,-2,-3) 호출 후, value1과 value4는 동시에 바뀜
value2=[1, 2, 3]
value3=[0, 0, 0]
value4=[-1, -2, -3] //value4.set(-1,-2,-3) 호출 후, value1과 value4는 동시에 바뀜

2.2 main()에 (2.2)은 method overloading의 예를 보여주고 있다. 실행결과를 적고 그 이유를 설명하라 (예를 들어, 어떤 fill과 print 메소드를 사용한 것인지). (10점)

[1, 2, 3] // void print(int[] arr) 사용
[-1, -2, -3, -4] // void print(int[] arr) 사용
[[4, -3, 5], [-4, 3, 7], [7, -2, 4]] // void print(int[][] array) 사용
fill1=[1, 2, 3] // TripleValue fill(int[] arr) 사용 후, TripleValue toString()사용
fill2=[-1, -2, -3] // TripleValue fill(int[] arr) 사용 후, TripleValue toString() 사용
fill3=[[4, -3, 5], [-4, 3, 7], [7, -2, 4]] // TripleValue[] fill(int[][] arr) 사용 후, void print(TripleValue[] array) 사용

학과 _____ 학번 _____ 이름 _____

2.3 main()에 (2.3)은 parameter passing의 예를 보여주고 있다. 실행결과를 적고, 그 이유를 설명하라 (예를 들어, 각 메소드 내부에서 인자가 어떻게 복사되어 사용되는지). (10점)

```
inside square a=1 // -1 * -1 = 1
after square a=-1 // square 메소드 안에서 int a는 값을 복사하여 사용하고 메소드 사용이 끝난후 파괴되므로, main의 a = -1는 변하지 않음
inside square1 a=1 // 1 * 1 = 1
inside square1 a=4 // 2 * 2 = 2
inside square1 a=9 // 3 * 3 = 9
after square1 arr1=[1, 2, 3] // square1 메소드 안에서 int a는 값을 복사하여 사용하고 메소드 사용이 끝난후 파괴되므로, main의 arr1 = 1, 2, 3은 변하지 않음
inside square2 a=1 // -1 * -1 = 1
inside square2 a=4 // -2 * -2 = 4
inside square2 a=9 // -3 * -3 = 9
inside square2 a=16 // -4 * -4 = 16
after sq uare2 arr2=[1, 4, 9, 16] // square2 메소드 안에서 int[] arr는 레퍼런스를 복사하여 arr2와 동일한 곳을 가리키고, 메소드 내부에서 값을 수정한 후 파괴되므로, main의 arr2 = 1, 4, 8, 16으로 변함
```

2.4 main()에 (2.4)는 TripleOperator 연산을 보여주고 있다. 실행결과를 적어라. (10점)

```
MIN[10, 20, -30]=-30.0
MAX[10, 20, -30]=20.0
MEDIAN[10, 20, -30]=10.0
MEAN[10, 20, -30]=0.0
SUM[10, 20, -30]=0.0
RANGE[10, 20, -30]=50.0
```

2.5 main()에 (2.5)는 TripleValue[] fill3 데이터를 사용하여 모든 TripleOperator 연산을 하는, nested for-loop 코드를 완성하라. 직접 코드에 작성. (10점)

2.6 main()에 (2.6)는 사용자 입력으로 x,y,z와 op를 받아서 TripleCalculator 연산을 하는 코드를 완성하라. 직접 코드에 작성. (10점)

2.7 private 접근 지정자를 설명하라. UserInput 클래스 생성자는 TripleValue나 TripleCalculator와는 달리, 생성자에 private 접근 지정자를 사용하고 있는데 그 효과를 설명하라. (10점)

Private 접근 지정자는 해당 클래스 내부에서만 사용 가능하다. UserInput 클래스는 생성자를 제외하고 모든 멤버가 static 이므로, UserInput 객체를 생성하여 static method를 사용하지 못하도록 private으로 지정하였다.
자바의 표준 클래스 라이브러리인 Math 클래스의 모든 메소드들은 static으로, 즉 클래스의 객체를 생성하지 않고 그 메소드가 정의된 클래스 이름을 통해서 호출될 수 있다. Math 클래스도 생성자에 private 접근 지정자를 사용하여 Math 객체를 생성해서 static method를 사용하지 못하게 막은 것과 동일한 원리이다.

-끝-