단국대학교 응용컴퓨터공학 JAVA 프로그래밍 I 중간고사 (2016년 가을학기) 2016년 10월 27일

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. Point 클래스이다. 다음 물음에 답하라. (40점)

```
public class Point {
    protected double x;
    protected double y;
    public static final Point Zero = new Point(0, 0);
    public static final Point One = new Point(1, 1);
    public Point() { this(0.0, 0.0); }
    public Point(double x, double y) { set(x, y); } // (1)
    public Point(Point other) { set(other); } // (1)
    public Point(double[ ] xy) { set(xy); } // (1)
    public void print() { System.out.println("(" + x + ","+ y + ")"); }
    public String toString() { return "(" + x + ","+ y + ")"; }
    public double getX() { return x; }
    public double getY() { return y; }
    public void setX(double x) { this.x = x; }
    public void setY(double y) { this.y = y; }
    public Point clone() { return new Point(this.x, this.y); } // (2) 자기 자신을 복제하는 메소드
    public void set(Point other) { this.x = other.x; this.y = other.y; }
    public void set(double x, double y) { this.x = x; this.y = y; }
    public void set(double[ ] xy) {
        if (xy.length == 2) { this.x = xy[0]; this.y = xy[1]; }
    }
    public void set() { this.x = 1; this.y = 2; }
    public Point set(double i) { Point p = new Point(i, i); return p; }
    public static double getDistance(Point p, Point q) {
        return Math.sqrt((p.x – q.x)*(p.x – q.x) + (p.y – q.y)*(p.y – q.y));
    }
    public double getDistance(Point other) {
        return getDistance(this, other);
    }
    public double getMagnitude() {
        return getDistance(this, Zero);
    }
}
public class Point3D extends Point {
    public static final Point3D Zero = new Point3D(0, 0, 0);
    public static final Point3D One = new Point3D(1, 1, 1);
    protected double z;
    public Point3D() { this(0, 0, 0); }
    public Point3D(int x, int y, int z) { set(x, y, z); }
```

학과 _____ 학번 _____ 이름 _____

```
    public void print() { System.out.println("(" + x + ","+ y + ","+ z +")"); }
    public String toString() { return "(" + x + ","+ y + ","+ z +")"; }
    public double getZ() { return z; }
    public void setZ(double z) { this.z = z; }
    public Point3D clone() { return new Point3D(this.x, this.y, this.z); } // (2) 자기복제 메소드
    public void set(Point other) { // 내부구현 (3)
        if (other instanceof Point3D) _____
        _____
    }
    public void set(Point3D other) { this.x = other.x; this.y = other.y; this.z = other.z; }
    public void set(double x, double y, double z) { this.x = x; this.y = y; this.z = z;}
    public void set(double[] xyz) { // 내부구현 (4)
        if (xyz.length == 3) _____
        _____
    }
    public void set() { this.x = 10; this.y = 20; this.z = 30; }
    public static double getDistance(Point3D p, Point3D q) {
        return Math.sqrt((p.x – q.x)*(p.x – q.x) + (p.y – q.y)*(p.y – q.y) + (p.z – q.z)*(p.z – q.z));
    }
    public double getDistance(Point3D other) {
        return getDistance(this, other);
    }
    public double getMagnitude() {
        return getDistance(this, Zero);
    }
}
```

1.1 Point 클래스의 생성자를 구현하라. (5점)

```
public Point(double x, double y) { set(x, y); } // (1) set(double, double)을 이용하여 정의

public Point(Point other) { set(other); } // (1) set(Point)을 이용하여 정의

public Point(double[] xy) { set(xy); } // (1) set(double[])을 이용하여 정의
```

1.2 Point와 Point3D 클래스의 clone() 자기자신을 복제하는 메소드를 구현하라. (5점)

```
public Point clone() {
        Point p = new Point(this.x, this.y);
        return p;
}

public Point3D clone() {
        Point3D p = new Point3D(this.x, this.y, this.z);
        return p;
}
```

1.3 Point3D 클래스의 set(Point)를 구현하라. (5점)

```
public void set(Point other) { // 내부구현 (3)
    if (other instanceof Point3D)
        set((Point3D)other);
    else
        super.set(other);
}
```

학과 _____ 학번 _____ 이름 _____

1.4 Point3D 클래스의 set(double[ ])를 구현하라. (5점)

```
public void set(double[ ] xyz) { // 내부구현 (4)
    if (xyz.length == 3) {
        this.x = xyz[0]; this.y = xyz[1]; this.z = xyz[2];
    }
    else if (xyz.length == 2)
        super.set(xyz);
}
```

1.5 Method overloading를 Point와 Point3D 클래스에서 예를 찾아 설명하라. (10점)

Method overloading은 동일한 함수명에 매개변수가 다른 함수를 둘 이상 정의하는 것으로 동일한 함수 기능을 수행하지만 다른 매개변수의 경우를 처리할 때 사용함

Point클래스의 void set(Point), void set(double, double), void set(double[]), void set(),Point set(double)
Point3D클래스의 void set(Point3D), void set(double, double, double)가 있다.

Point클래스의 double getDistance(Point, Point), double getDistance(Point)
Point3D클래스의 double getDistance(Point3D, Point3D), double getDistance(Point3D)가 있다.

Method overriding을 Point와 Point3D 클래스에서 예를 찾아 설명하라.

Method overriding은 상속받은 파생클래스에서 동일한 함수명에 동일한 매개변수 정의하여 함수를 재정의하는 것으로 상속되어진 함수의 기능을 변경해서 재사용하고 싶은 때 사용함

Point3D 클래스에서 Point 클래스의 method를 overriding 한 것들은
void set(), void set(double[]),void set(Point), double getMagnitude(), String toString(),
void print(), Point clone()가 있다.

실행시 dynamic binding되어 해당 객체의 실제 타입에 맞는 override된 method가 호출

```
class PointTest {
    public static void main(String[ ] args) {
        Point one1 = Point.One;
        Point other1 = one1.clone();
        other1.print(); // (1)
        Point3D one2 = Point3D.One;
        Point3D other2 = one2.clone();
        other2.print(); // (1)
        one1.set();
        System.out.println("one1.set()=" + one1); // (2)
        one1.set(Point.Zero);
        System.out.println("one1.set(Point.Zero)=" + one1); // (3)
        one1.set(one2);
        System.out.println("one1.set(one2)=" + one1); // (4)
        Point one3 = new Point3D(1, 2, 3);
        one3.print(); // (5)
        one3.set(one1);
        System.out.println("one3.set(one1)=" + one3); // (6)
        one3.set(one2);
        System.out.println("one3.set(one2)=" + one3); // (7)
        one3.set();
        System.out.println("one3.set()=" + one3); // (8)
        double[] xy = { 100, 200 };
        one3.set(xy);
```

학과 _____ 학번 _____ 이름 _____

```
        System.out.println("one3.set()=" + one3); // (9)
        Point other3 = one3.clone();
        other3.print(); // (10)
    }
}
```

1.6 main의 실행결과를 자세히 나타내라. 호출 번호도 표시할 것. set 메소드 매개인수에 주의할 것. (10점)

```
(1.0, 1.0) // (1)
(1.0, 1.0, 1.0) // (1)
one1.set()=(1.0, 2.0) // (2)
one1.set(Point.Zero)=(0.0, 0.0) // (3)
one1.set(one2)=(1.0, 1.0) // (4)
(1.0, 2.0, 3.0) // (5)
one3.set(one1)=(1.0, 1.0, 3.0) // (6)
one3.set(one2)=(1.0, 1.0, 1.0) // (7)
one3.set()=(10.0, 20.0, 30.0) // (8)
one3.set(xy)=(100.0, 200.0, 30.0) // (9)
(100, 200, 30) // (10)
```

2. Bound 클래스의 생성자를 구현하라. (10점)

```
public class Bound {
    private Point bottomLeft;
    private Point bottomRight;
    private Point topLeft;
    private Point topRight;
    private double width;
    private double height;
    public Bound() { // (1) 기본 생성자
        this(new Point(0.0, 0.0), 1.0, 1.0);
    }
    public Bound(Point origin, double width, double height) {
        this.bottomLeft = origin;
        this.bottomRight = new Point(origin.getX() + width, origin.getY());
        this.topLeft = new Point(origin.getX(), origin.getY() + height);
        this.topRight = new Point(origin.getX() + width, origin.getY() + height);
        this.width = width;
        this.height = height;
    }
    public Bound(int x, int y, double width, double height) { // (2)
        this.bottomLeft = new Point(x, y);
        this.bottomRight = new Point(x + width, y);
        this.topLeft = new Point(x, y + height);
        this.topRight = new Point(x + width, y + height);
        this.width = width;
        this.height = height;
    }
    public Bound(Point bottomLeft, Point topRight) { // (3)
        this.bottomLeft = bottomLeft;
        this.bottomRight = new Point(topRight.getX(), bottomLeft.getY());
        this.topLeft = new Point(bottomLeft.getX(), topRight.getY());
        this.topRight = topRight;
        this.width = topRight.getX() – bottomLeft.getX();
        this.height = topRight.getY() – bottomLeft.getY();
    }
```

```
    public void print() { System.out.println(this); }
    public String toString() {
        return "Bound [" + bottomLeft + ","+ bottomRight + "," + topRight + "," + topLeft + "]" + width + "x" +
height;
    }
}
```

3. RegularPolygonCalculator 프로그램 일부이다. 다음 물음에 답하라. (30점)

```
public enum PolygonType {
    TRIANGLE(3), SQUARE(4), PENTAGON(5), HEXAGON(6), HEPTAGON(7), OCTAGON(8);
    private int type;
    private PolygonType (int type) { this.type = type; }
    public int getType() { return type; }
    public static PolygonType valueOf(int type) { // (1)
        // 내부 구현 필요 (type에 따른 PolygonType 반환)
    }
}
public class RegularPolygon {
    private PolygonType type;
    private int degree;
    private double radius;
    private Point center;
    private double angle;
    private Point[] points;
    public RegularPolygon(int degree, double radius, Point center) {
        this.degree = clamp(degree, 3, 8); // (2)
        this.type = PolygonType.valueOf(this.degree); // (1)
        this.radius = radius;
        this.center = center;
        this.angle = 2 * Math.PI / degree;
        this.points = createPoints(); // (3)
    }
    private int clamp(int degree, int min, int max) { // (2)
        // 내부 구현 필요 (degree 값이 min보다 작으면 min으로 max보다 크면 max으로 반환)
    }
    private Point[] createPoints() { // (3)
        // 내부 구현 필요 (degree, center, angle를 이용해서 points 계산, e.g. 오각형이면 점 5개)
    }
    private static Point getPoint(double radian, double radius, Point offset) { // 정다각형의 한 점 계산
        return new Point(radius * Math.cos(radian) + offset.getX(), radius * Math.sin(radian) + offset.getY());
    }
    public Bound getBound() {
        return new Bound(new Point(center.getX() – radius, center.getY() – radius), radius*2, radius*2);
    }
    public double getArea() {
        return radius * radius * degree * Math.sin(angle) / 2.0;
    }
    public String getAreaFormula() {
        return "Area=" + radius + "x" + radius + "x" + degree + "x sin(" + (angle * 180.0 / Math.PI) + ") / 2= ";
    }
    public void print() { System.out.println(this); }
    public String toString() {
```

학과 _____ 학번 _____ 이름 _____

```
            String s = type.toString() + " [radius="  + radius + ", center=" + center.toString() + "]\n";
            for (int i = 0; i < degree; i++) s += (points[i].toString() + "\n");
            return s;
        }
}
public class Utility {
        public static int getUserInputBetween(int min, int max) { /* 내부구현 생략 */ }
        public static double getUserInputDouble() { /* 내부구현 생략 */ }
        public static boolean getUserInputExitKey() { /* 내부구현 생략 */ }
}
public class RegularPolygonCalculator {
        public static void calculate(int degree) { calculate(degree, 1.0, new Point()); }
        public static void calculate(int degree, double radius, Point center) {
            RegularPolygon poly = new RegularPolygon(degree, radius, center);
            System.out.println(poly);
            // 내부 구현 필요 (AreaFormula, Area, Bound 출력) (4)
        }
        public static void calculate() {
            int degree = 3;
            double radius = 3.0;
            Point center = new Point();
            while (!Utility.getUserInputExitKey()) {
                System.out.println("Please enter the Polygon numberOfSides: ");
                degree = Utility.getUserInputBetween(3, 8);
                System.out.println(" Please enter the Polygon radius: ");
                // 내부 구현 필요 (radius, center 사용자 입력을 받음) (5)
                calculate(degree, radius, center);
            }
        }
}
class Main {
        public static void main(String[ ] args) {
            for (PolygonType t : PolygonType.values()) // (6)각 PolygonType의 radius=1, 점(0,0)에 따른 계산
                RegularPolygonCalculator.calculate(t.getType());
            RegularPolygonCalculator.calculate(); // 사용자 입력에 따른 계산
        }
}
```

3.1 public static PolygonType valueOf(int type)를 switch 구문을 사용하여 구현하라. (5점)

```
public static PolygonType valueOf(int type) { // (1) type값에 맞게 PolygonType 반환
    switch (type) {
    case 3: return TRIANGLE;
    case 4: return SQUARE;
    case 5: return PENTAGON;
    case 6: return HEXAGON;
    case 7: return HEPTAGON;
    case 8: return OCTAGON;
    default: return null;
    }
}
```

학과 _____ 학번 _____ 이름 _____

3.2 private int clamp(int degree, int min, int max)를 if/else 구문을 사용하여 구현하라. (5점)

```
private int clamp(int degree, int min, int max) { // (2) degree를 [min, max] 사이로 맞춰줌
    if (degree < min) degree = min;
    else if (degree > max) degree = max;
    return degree;
}
```

3.3 private Point[] createPoints()를 for구문을 사용하여 구현하라. (5점)

```
private Point[] createPoints() { // (3) -90도(즉, y-축에서부터) 시작해서 다각형의 점들 계산
    Point[] points = new Point [degree];
    double radian = Math.PI / 2.0; // -90
    for (int i = 0; i < degree; i++) {
        points[i] = getPoint(radian, radius, center);
        radian += angle;
    }
    return points;
}
```

3.4 public static void calculate(int degree, double radius, Point center)를 구현하라. (5점)

```
public static void calculate(int degree, double radius, Point center) { // (4)
        RegularPolygon poly = new RegularPolygon(degree, radius, center);
        System.out.println(poly);
        System.out.println(poly.getAreaFormula());
        System.out.println(poly.getArea());
        System.out.println(poly.getBound());
}
```

3.5 public static void calculate() 내부를 Utility.getUserInputDouble()을 사용하여 완성하라. (5점)

```
public static void calculate() { // (5)
    int degree = 3;
    double radius = 3.0;
    Point center = new Point();
    while (!Utility.getUserInputExitKey()) {
        System.out.println("Please enter the Polygon numberOfSides: ");
        degree = Utility.getUserInputBetween(3, 8);
        System.out.println("Please enter the Polygon radius: ");
        radius = Utility.getUserInputDouble();
        System.out.println("Please enter the Polygon center point x: ");
        center.setX(Utility.getUserInputDouble());
        System.out.println("Please enter the Polygon center point y: ");
        center.setY(Utility.getUserInputDouble());
        calculate(degree, radius, center);
    }
}
```

학과 _____ 학번 _____ 이름 _____

3.6 main 메소드에 밑줄 친 부분 (6)과 동일한 결과가 출력될 수 있도록 do-while 로 구현하라. (5점)

```
int i = 3; // do..while
do {
    RegularPolygonCalculator.calculate(i++);
} while (i < 9);
```

4. 다음은 ArrayTest 클래스 프로그램의 main 실행결과를 자세히 나타내라. 실행결과에 sqrt를 사용해서 숫자나 계산으로 적어줄 것. Polymorphism에 유의할 것. (20점)

```
class ArrayTest {
    public static void print(Object o) { System.out.println(o); }
    public static void print(int[] arr) { System.out.println(java.util.Arrays.toString(arr)); }
    public static void print(double[] arr) { System.out.println(java.util.Arrays.toString(arr)); }
    public static void print(String[] arr) { System.out.println(java.util.Arrays.toString(arr)); }
    public static void print(Point[] arr) { System.out.println(java.util.Arrays.toString(arr)); }
    public static void fill(String[] arr) { for (int i = 0; i < arr.length; i++) arr[i] = "" + i; }
    public static void fill(String[] arr, int size) { for (int i = 0; i < size; i++) arr[i] = "a"; }
    public static void swap(Point p, Point q) {
        Point t = p;
        p = q;
        q = t;
    }
    public static void swap2(Point p, Point q) {
        double[] t = p.clone();
        p.set(q.clone());
        q.set(t);
    }
    public static void main(String[ ] args) {
            int[] array1 = { 1, 2, 3 };
            print(array1); // [1, 2, 3]
            String[] array2 = { "a", "b", "c" };
            print(array2); // [a, b, c]
            String[] array3 = { "1", "2", "3", "4", "5" };
            print(array3); // [1, 2, 3, 4, 5]

            fill(array2);
            print(array2); // (1)
            fill(array3, 2);
            print(array3); // (2)

            Point[] points = new Point[5];
            points[0] = Point.One;
            points[1] = new Point(4, 5);
            points[2] = Point3D.Zero;
            points[3] = new Point3D(3, 4, 5);
            points[4] = point[3];
            print(points); // (3)

            double[] lengths = new double [5];
```

학과 _____ 학번 _____ 이름 _____

```
        lengths[0] = points[0].getDistance(points[1]);
        print(lengths[0]); // (4)
        lengths[1] = Point.getDistance(points[0], points[1]);
        print(lengths[1]); // (5)
        lengths[2] = Point3D.getDistance(points[2], points[3]);
        print(lengths[2]); // (6)
        lengths[3] = Point3D.getDistance((Point3D)points[2], (Point3D)points[3]);
        print(lengths[3]); // (7)
        lengths[4] = points[2].getDistance(points[3]);
        print(lengths[4]); // (8)

        swap(points[0], points[1]);
        System.out.println("After swap p1=" + points[0] + " p2="  + points[1]); // (9)
        swap2(points[0], points[1]);
        System.out.println("After swap2 p1=" + points[0] + " p2=" + points[1]); // (10)
    }
}
```

4.1 main의 실행결과를 자세히 나타내라. 실행결과에 sqrt를 사용해서 숫자나 계산으로 적어줄 것. Polymorphism에 유의할 것. (20점)

```
[0, 1, 2]                                   // (1)
[a, a, 3, 4, 5]                             // (2)
[(1, 1), (4, 5), (0, 0, 0), (3, 4, 5), (3, 4, 5)]   // (3)
5                                   // (4) sqrt(25=9+16)
5                                   // (5) sqrt(25=9+16)
5                                   // (6) sqrt(25=9+16)
7.071                               // (7) sqrt(50=9+16+25)
5                                   // (8) sqrt(25=9+16)
After swap p1=(1, 1) p2=(4,5)       // (9)
After swap2 p1=(4, 5) p2=(1,1)      // (10)
```

-끝-