

기말고사

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. (60점)

```
public class Point implements Comparable<Point> { // (3)
    protected double x;
    protected double y;
    public Point() { this(0.0, 0.0); } // (1)
    public Point(double x, double y) { this.x = x; this.y = y; } // (1)
    public Point(Point other) { set(other); } // (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 void set(Point other) { this.x = other.x; this.y = other.y; }
    public int compareTo(Point other) { return Double.compare(this.getX(), other.getX()); } // (4)
    public static Comparator<Point> YComparator = new Comparator<Point>() { // (4)
        public int compare(Point p1, Point p2) {
            return Double.compare(p1.getY(), p2.getY());
        }
    };
}

public class Point3D extends Point { // (3)
    protected double z;
    public Point3D() { this(0, 0, 0); } // (1)
    public Point3D(int x, int y, int z) { this.x = x; this.y = y; this.z = z; } // (1)
    public Point3D(Point3D other) { set(other); } // (1)
    public Point3D(Point other) { set(other); } // (1)
    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 void set(Point other) { // (2)
        if (other instanceof Point3D) _set((Point3D)other);
        else _super.set(other);
    }
    public void set(Point3D other) { this.x = other.x; this.y = other.y; this.z = other.z; } // (2)
    public static Comparator<Point3D> ZComparator = new Comparator<Point3D>() { // (4)
        public int compare(Point3D p1, Point3D p2) {
            return Double.compare(p1.getZ(), p2.getZ());
        }
    };
}
```

```

}
}
class PointTest {
    public static void print(Object[] arr) { System.out.println(java.util.Arrays.toString(arr)); }
    public static void pointTest() { // (5)
        Integer[] array1 = { 1, 2, 3 };
        print(array1); // (1) [1, 2, 3]
        String[] array2 = { "a", "b", "c" };
        print(array2); // (2) [a, b, c]
        Point[] array3 = new Point[3];
        array3[0] = new Point(1, 1);
        array3[1] = new Point3D(0, 0, 0);
        array3[2] = array3[0];
        print(array3); // (3) [ (1.0, 1.0), (0.0, 0.0, 0.0), (1.0, 1.0) ]
        Point one1 = new Point(10, 10);
        one1.print(); // (4) (10.0, 10.0)
        Point one2 = new Point(one1);
        one2.setX(20);
        one2.setY(30);
        one2.print(); // (5) (20, 30)
        Point one3 = new Point3D(100, 100, 100);
        one3.print(); // (6) (100, 100, 100)
        Point one4 = new Point3D(1, 2, 3);
        one4.print(); // (7) (1, 2, 3)
        one4.set(one1);
        System.out.println("one4.set(one1)=" + one4); // (8) (10, 10, 3)
        one4.set(one2);
        System.out.println("one4.set(one2)=" + one4); // (9) (20, 30, 3)
        one4.set(one3);
        System.out.println("one4.set(one3)=" + one4); // (10) (100, 100, 100)
    }
    public static void arrayListTest() { // (6)
        ArrayList<Point3D> qList = new ArrayList<Point3D>();
        qList.add(new Point3D(0, 1, 2));
        qList.add(new Point3D(-6, 7, 8));
        qList.add(new Point3D(-10, 20, 30));
        qList.sort((Point p1, Point p2) -> // (6)
            Double.compare(p1.getX(), p2.getX())
        );
        qList.sort(new Comparator<Point3D>() { // (6)
            public int compare(Point3D p1, Point3D p2) {
                return Double.compare(p1.getY(), p2.getY());
            }
        });
        qList.sort(Point3D.ZComparator); // (6)
    }
    public static void main(String[] args) {
        pointTest();
        arrayListTest();
    }
}

```

1.1 Point와 Point3D 클래스의 생성자를 구현하라. (10점)

```
public Point() { this(0.0, 0.0); } // (1)
public Point(double x, double y) { this.x = x; this.y = y; } // (1)
public Point(Point other) { set(other); } // (1)

public Point3D() { this(0, 0, 0); } // (1)
public Point3D(int x, int y, int z) { this.x = x; this.y = y; this.z = z; } // (1)
public Point3D(Point3D other) { set(other); } // (1)
public Point3D(Point other) { set(other); } // (1)
```

1.2 Point3D 클래스의 set(Point3D)와 set(Point)를 구현하라. (10점)

```
public void set(Point other) { // (2)
    if (other instanceof Point3D)
        set((Point3D)other);
    else
        super.set(other);
}
public void set(Point3D other) { // (2)
    this.x = other.x;
    this.y = other.y;
    this.z = other.z;
}
```

1.3 Point 클래스의 implements 와 Point3D 클래스의 extends 차이점을 설명하라. (10점)

class Point implements Comparable<Point>는 인터페이스를 상속받아서 해당 인터페이스의 메소드 즉, int compareTo(Point)를 구현하는 것이다.

class Point3D extends Point는 Point3D 클래스가 Point 부모 클래스의 protected와 public인 필드와 메소드를 상속받아서 사용가능하고 추가적인 메소드를 더 구현하는 것이다.

1.4 Point와 Point3D 클래스에서 int compareTo(Point other) 메소드를 설명하라. Comparator<Point> YComparator 나 Comparator<Point3D> ZComparator 와의 차이점을 설명하라. (10점)

Comparable 인터페이스를 상속하는 Point 클래스는 int compareTo(Point other) 메소드를 반드시 구현해야한다. Point를 collections으로 사용했을 시 sort() 기본 정렬에서 사용된다. Point의 x 값의 오름차 순서로 정렬하는 방식이다.

Comparator<Point> YComparator 나 Comparator<Point3D> ZComparator 는 y 값이나 z 값으로 오름차 순서로 정렬하고자 할 때 사용하는 방식이다. Comparator 인터페이스를 상속함으로써 구현해야하는 메소드로는 int compare(Point p1, Point p2)이다. Point/Point3D를 collections으로 사용했을 시 sort(YComparator) 또는 sort(ZComparator)를 사용하여 y 또는 z 값으로 정렬할 수 있다.

1.5 pointTest()의 실행결과를 자세히 나타내라. 호출 번호도 표시할 것. Polymorphism에 유의할 것. (10점)

```
[1, 2, 3] // (1)
[a, b, c] // (2)
[(1.0, 1.0), (0.0, 0.0, 0.0), (1.0, 1.0)] // (3)
(10.0, 10.0) // (4)
(20.0, 30.0) // (5)
(100.0, 100.0, 100.0) // (6)
(1.0, 2.0, 3.0) // (7)
one4.set(one1)=(10.0, 10.0, 3.0) // (8)
one4.set(one2)=(20.0, 30.0, 3.0) // (9)
one4.set(one3)=(100.0, 100.0, 100.0) // (10)
```

1.6 arrayListTest()에서 밑줄 친 부분의 sort 메소드를 설명하라. (10점)

```
qList.sort((Point p1, Point p2) -> // (1) lambda 방식으로 구현한 x 값으로 정렬
Double.compare(p1.getX(), p2.getX())
);
qList.sort(new Comparator<Point3D>() { // (2) anonymous method 방식으로 구현한 y 값으로 정렬
public int compare(Point3D p1, Point3D p2) {
return Double.compare(p1.getY(), p2.getY());
}
});
qList.sort(Point3D.ZComparator); // (3) Comparator 클래스를 이용한 z 값으로 정렬
```

2. ShapeDrawing 프로그램 일부이다. 다음 물음에 답하라. (30점)

```
public class ShapeDrawing extends JFrame {
    ShapeDrawingPanel polyCanvas = new ShapeDrawingPanel();
    public static void main(String[] args) {
        new ShapeDrawing();
    }
    public ShapeDrawing() {
        super("Shape Drawing Frame");
        setBounds(10, 10, 350, 350);
        setResizable(false);
        setVisible(true);
        polyCanvas.setPreferredSize(new Dimension(300, 300));
        this.add(polyCanvas);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
    private class ShapeDrawingPanel extends JPanel implements MouseListener
        ArrayList<RandomShape> shapes = new ArrayList<RandomShape>; // (1) 리스트를 생성한다.
        public ShapeDrawingPanel() {
            System.out.println("ShapeDrawingPanel");
            addMouseListener(this);
        }
        @Override
        public void paintComponent(Graphics g) { // (2)
            super.paintComponent(g);
            Graphics2D g2 = (Graphics2D) g;
            g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
                RenderingHints.VALUE_ANTIALIAS_ON);
            for (RandomShape s : shapes) // (1) 모든 도형을 그린다
                s.draw(g2);
        }
    }
}
```

```

    }
    public void mouseClicked(MouseEvent e) {
    }
    public void mousePressed(MouseEvent e) {
        System.out.println("Pressed");
        Point center = new Point((double) e.getX(), (double) e.getY());
        RandomShape s = RandomShapeFactory.getInstance(center);
        shapes.add(s); // (1) 새로운 도형을 리스트에 추가한다.
        repaint(); // (2)
    }
    public void mouseReleased(MouseEvent e) {
    }
    public void mouseEntered(MouseEvent e) {
    }
    public void mouseExited(MouseEvent e) {
    }
}
}

```

2.1 ShapeDrawingPanel 클래스 내부의 밑줄 친 부분을 채워라. (10점)

```

ArrayList<RandomShape> shapes = new ArrayList<RandomShape>(); // (1) 리스트를 생성한다.
for (RandomShape s : shapes) // (1) 모든 도형을 그린다
    s.draw(g2);
shapes.add(s); // (1) 새로운 도형을 리스트에 추가한다.

```

2.2 ShapeDrawingPanel 클래스의 내부에 paintComponent(Graphics g) 메소드와 mousePressed(MouseEvent e) 안에 repaint()가 어떤 메소드인지, 이 프로그램의 동작에 미치는 영향을 자세히 설명하라. (10점)

paintComponent 메소드는 JPanel을 상속받은 ShapeDrawingPanel 클래스가 내부적으로 그리기(Paint) 이벤트가 발생 시 호출되는 메소드이다.
 이 프로그램에서는 그리기가 호출 시 모든 도형을 다시 그린다.

repaint() 메소드는 프로그램적으로 원하는 시점에서 사용함으로써 내부적으로 그리기(Paint) 이벤트를 발생시키는 메소드이다. repaint() 가 호출되면 그리기 이벤트가 발생됨으로써 paintComponent 메소드가 호출되는 방식이다.

2.3 ShapeDrawingPanel 클래스는 인터페이스를 구현하는 방식을 사용하여 MouseListener 이벤트 핸들러를 구현하였다. 어떻게 구현하였는지 자세히 설명하라. (10점)

이벤트 핸들러의 구현은 (1) 독립적 이벤트 클래스를 사용하거나 (2) 내부 이벤트 클래스를 사용하거나 (3) 인터페이스를 구현하거나 (4) 무명 이벤트 클래스를 사용하거나 (5) 람다식을 사용하거나 다양하다. 이 프로그램에서는 MouseListener 인터페이스를 상속받아 구현한 방식으로 아래의 메소드를 구현해야한다.

```

void mouseClicked(MouseEvent e)
void mousePressed(MouseEvent e)
void mouseReleased(MouseEvent e)
void mouseEntered(MouseEvent e)
void mouseExited(MouseEvent e)

```

그리고 ShapeDrawingPanel 생성자에서 현재 패널에 addMouseListener(this);를 등록한다.

3. HW3 프로그램 일부이다. 다음 물음에 답하라. (20점)

```
public class GeometryFrame extends JFrame implements ItemListener, KeyListener {
    private Geometry geo = null;
    private int selectedIndex = -1;
    private String[] comboBoxItems = GeometryType.names();
    private JComboBox combo1 = new JComboBox(comboBoxItems);
    private JLabel label1 = new JLabel("label1");
    private JLabel label2 = new JLabel("label2");
    private JLabel label3 = new JLabel("label3");
    private JLabel label4 = new JLabel("label4");
    private JLabel label5 = new JLabel("label5");
    private JTextField textfield1 = new JTextField(20);
    private JTextField textfield2 = new JTextField(20);
    private JTextField textfield3 = new JTextField(20);
    private JTextField textfield4 = new JTextField(20);
    private JTextField textfield5 = new JTextField(20);
    private JButton button1 = new JButton("확인");
    private JButton button2 = new JButton("취소");
    private JPanel panel1 = new JPanel();
    public GeometryFrame() {
        setLayout(null); // (1)
        combo1.setBounds(20, 20, 300, 20);
        combo1.addItemListener(this);
        add(combo1);
        label1.setBounds(20, 50, 100, 20);
        add(label1);
        // 중간 생략...
        textfield1.setBounds(120, 50, 200, 20);
        textfield1.addKeyListener(this);
        add(textfield1);
        // 중간 생략...
        button1.setSize(140, 20);
        button2.setSize(140, 20);
        panel1.add(button1);
        panel1.add(button2);
        add(panel1);
    }
    public void itemStateChanged(ItemEvent e) {
        JComboBox cb = (JComboBox)e.getSource(); // (2)
        selectedIndex = cb.getSelectedIndex(); // (2)
        switch (selectedIndex) {
            // 중간 생략...
            case 2:
                label1.setText("Radius");
                label2.setText("Height");
                label1.setVisible(true);
                label2.setVisible(true);
                label3.setVisible(false);
                textfield1.setVisible(true);
                textfield2.setVisible(true);
                textfield3.setVisible(false);
                break;
        }
        // 중간 생략...
    }
}
public void keyTyped(KeyEvent e) { }
```



```
public void keyReleased(KeyEvent e) {}
public void keyPressed(KeyEvent e) {
    int key = e.getKeyCode(); // (3)
    if (key == KeyEvent.VK_ENTER) { // (3)
        switch (selectedIndex) {
            // 중간 생략...
            case 2:
                geo = new Cone (Double.parseDouble(textfield1.getText()),
                Double.parseDouble(textfield2.getText()));
                break;
            // 중간 생략...
        }
        if (geo != null) {
            textfield4.setText(""+ geo.getSurfaceArea());
            textfield5.setText(""+ geo.getVolume());
        }
    }
}
```

3.1 GeometryFrame 클래스 생성자 내부에 setLayout(null);은 무엇인지 자세히 설명하라. (5점)

setLayout(null);을 사용하면 label, textfield과 같은 GUI 요소들을 setBounds 메소드를 사용하여 절대 위치에 배치시킬 수 있다. 예를 들어, combo1.setBounds(20, 20, 300, 20); label1.setBounds(20, 50, 100, 20); textfield1.setBounds(120, 50, 200, 20); 등등

3.2 itemStateChanged(ItemEvent e)의 ItemEvent e는 무엇인지 어떻게 사용되는지 자세히 설명하라. (5점)

콤보박스의 아이템이 바뀌게 되면 받는 이벤트 값으로, e.getSource()가 콤보박스인지 확인하여 cb.getSelectedIndex();를 사용하여 인덱스값을 selectedIndex로 넘겨준다. selectedIndex를 사용하여 특정 레이블의 값을 지정하고 레이블/텍스트필드가 visible/hide한다.

3.3 keyPressed(KeyEvent e)의 KeyEvent e는 무엇인지 어떻게 사용되는지 자세히 설명하라. (5점)

텍스트필드에 키 값이 들어오면 받는 이벤트 값으로, e.getKeyCode()가 무슨 키인지 확인하여 KeyEvent.VK_ENTER 키가 들어왔으면 selectedIndex에 따라서 Geometry 객체를 생성한다.

3.4 이 프로그램의 동작 원리를 자세히 설명하라. (5점)

생성자에서는 레이블과 텍스트필드와 버튼 등을 배치된다. 그리고 콤보박스의 아이템이 바뀌게 되면 selectedIndex를 설정하고, 그에 따라 레이블과 텍스트필드의 값이 지정되거나 visible/hide하게 한다. 그리고 텍스트필드에 ENTER 키가 들어오면 selectedIndex에 따라서 Geometry 객체를 생성하여 계산을 한 후, textfield4와 textfield5에 결과값을 출력한다.