단국대학교 응용컴퓨터공학 JAVA 프로그래밍 I 기말고사 (2019년 봄학기) 2019년 6월 12일

학과 _____ 학번 _____ 이름 _____

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. 다음 코드의 실행 결과를 적어라 (parameter passing에 유의할 것). (10점)

```
public static int mystery(int n, int[] numbers) {
    n += 100;
    numbers[2]--;
    System.out.println(n + " " + Arrays.toString(numbers));
    return numbers[1] * 2;
}
public static void mystery(int[] x, int[] y) {
    int[] t = x;
    x = y;
    y = t;
    System.out.println(Arrays.toString(x) + " " + Arrays.toString(y));
}
public static void mystery(int[] arr) {
    for (int i = 1; i < arr.length - 1; i++) {
        if (arr[i] <= arr[i + 1]) {
            arr[i + 1] = i * 2;
        }
    }
}
public static void main(String[] args) {
    int a = 4;
    int b = 8;
    int[] data = {5, 10, 15};
    a = mystery(b, data);
    System.out.println(a + " " + b + " " + Arrays.toString(data));
    System.out.println();

    int[] x = {1, 2, 3};
    int[] y = {4, 5};
    mystery(x, y);
    System.out.println(Arrays.toString(x) + " " + Arrays.toString(y));
    System.out.println();

    int[] data1 = {4, 1, 3};
    int[] data2 = {2, 1, 3, 2};
    int[] data3 = {1, 1, 1, 1, 8};
    mystery(data1);
    System.out.println("data1 = " + Arrays.toString(data1));
    mystery(data2);
    System.out.println("data2 = " + Arrays.toString(data2));
    mystery(data3);
    System.out.println("data3 = " + Arrays.toString(data3));
}
```

학과 _____ 학번 _____ 이름 _____

```
108 [5, 10, 14]
20 8 [5, 10, 14]

[4, 5] [1,2,3] // 자바는 pass-by-value 방식이라서, 내부에서는 바뀌지만
[1, 2, 3] [4, 5] // 자바는 pass-by-value 방식이라서, 외부에는 영향을 주지 않는다.

data1 = [4, 1, 2]
data2 = [2, 1, 2, 4]
data3 = [1, 1, 2, 1, 6]
```

2. 다음은 Point와 Bound 클래스이다. 다음 질문에 답하라. (40점)

```
class Point {
    private int x, y;
    public Point(int x, int y) {
        this.x = x;
        this.y = y;
    }
    public int getX() {
        return x;
    }
    public int getY() {
        return y;
    }
    public boolean equals(Point other) {
        if (this == other) return true;
       return x == other.x && y == other.y;
    }
    @Override
    public String toString() {
        return "Point[" + x + "," + y + "]";
    }
}
class Bound {
    private int x, y, w, h;
    public Bound(int x, int y, int w, int h) {
        this.x = x;
        this.y = y;
        this.w = w;
        this.h = h;
    }
    public static Bound findBound(Point[] arr) {
        int xmin = arr[0].getX();
        int xmax = arr[0].getX();
        int ymin = arr[0].getY();
        int ymax = arr[0].getY();
        for (Point p : arr) {
            xmin = Math.min(p.getX(), xmin);
            xmax = Math.max(p.getX(), xmax);
            ymin = Math.min(p.getY(), ymin);
            ymax = Math.max(p.getY(), ymax);
        }
        return new Bound(xmin, ymin, xmax – xmin, ymax - ymin);
    }
    @Override
    public String toString() {
        return "Bound[" + x + "," + y + "," + w + "," + h + "]";
    }
}
```

학과 _____ 학번 _____ 이름 _____

2.1 findBound(Point[]) 메소드의 기능을 설명하고, 다음 코드 실행결과를 적어라. (5점)

```
Point[] pArray1 = {
        new Point(7, 5), new Point(8, 5), new Point(9, 7), new Point(2, 3)
};
Point[] pArray2 = {
        new Point(2, 3), new Point(0, 15), new Point(15, 10)
};
Bound<Point> b1 = Bound.findBound(pArray1);
Bound<Point> b2 = Bound.findBound(pArray2);
System.out.println(b1);
System.out.println(b2);
```

Bound 클래스의 findBound는 점 배열을 입력 받아서 (xmin, ymin) (xmax, ymax) 찾은 후
Bound(xmin, ymin, xmax – xmin, ymax – ymin)을 돌려줌

Bound[2,3,7,4]
Bound[0,3,15,12]

2.2 다음 코드의 실행결과를 적고 ==과 equals를 자세히 설명하라. (10점)

```
Point v1 = pArray1[3];
Point v2 = v1;
Point v3 = pArray2[0];
System.out.println("v1=" + v1);
System.out.println("v2=" + v2);
System.out.println("v3=" + v3);
System.out.println("v1 == v2 " + (v1 == v2));
System.out.println("v1 == v3 " + (v1 == v3));
System.out.println("v1 equals v2 " + v1.equals(v2));
System.out.println("v1 equals v3 " + v1.equals(v3));

Bound v4 = new Bound(1, 1, 10, 10);
Bound v5 = v4;
Bound v6 = new Bound(1, 1, 10, 10);
System.out.println("v4=" + v4);
System.out.println("v5=" + v5);
System.out.println("v6=" + v6);
System.out.println("v4 == v5 " + (v4 == v5));
System.out.println("v4 == v6 " + (v4 == v6));
System.out.println("v4 equals v5 " + v4.equals(v5));
System.out.println("v4 equals v6 " + v4.equals(v6));
```

v1=Point[2,3]
v2=Point[2,3]
v3=Point[2,3]
v1 == v2 true // 레퍼런스가 같은 v1 == v2
v1 == v3 false // 레퍼런스가 다름
v1 equals v2 true // 레퍼런스가 같으므로, 객체의 내용이 같음
v1 equals v3 true // 객체의 내용이 같으므로 (Point클래스의 equals 재정의 되어 있으므로)
v4=Bound[1,1,10,10]
v5=Bound[1,1,10,10]
v6=Bound[1,1,10,10]
v4 == v5 true // 레퍼런스가 같은 v4 == v5
v4 == v5 false // 레퍼런스가 다름
v4 equals v5 true // 레퍼런스가 같으므로, 객체의 내용이 같음
v4 equals v6 false // (Bound클래스에 equals 재정의 되어 있지 않으므로)

학과 _____ 학번 _____ 이름 _____

2.3 다음 코드를 설명하고, 실행결과를 적어라. (5점)

```
int[] iArray = {7, 5, 8, 5, 9, 7, 2, 3};
List<Integer> iList = new ArrayList<>();
for (int v : iArray) {
    iList.add(v);
}
System.out.println(iList.toString()); // [8, 9, 2, 3]
```

iArray 정수 배열을 이용하여 List<Integer> iList 리스트에 추가하고 그 결과를 출력
[7, 5, 8, 5, 9, 7, 2, 3]

2.4 위의 iList에서 5와 7을 모두 지우는 코드를 작성하라. (5점)

```
Iterator<Integer> it = iList.iterator();
while (it.hasNext()) {
    Integer v = it.next();
    if (v == 5 || v == 7) it.remove();
}
```

2.5 배열에서 5와 7을 모두 지우는 int[] remove57(int[] arr) 메소드를 작성하라. (10점)

```
int[] iArray = {7, 5, 8, 5, 9, 7, 2, 3};
iArray = remove57(iArray);
System.out.println(Arrays.toString(iArray)); // [8, 9, 2, 3]
```

```
public static int[] remove57(int[] arr) {
    int count = 0;
    for (int v : arr) {
        if (v == 5 || v == 7) continue;
        count++;
    }
    int[] temp = new int[count];
    int index = 0;
    for (int v : arr) {
        if (v == 5 || v == 7) continue;
        temp[index++] = v;
    }
    return temp;
}
```

2.6 Array와 ArrayList의 차이점 4개를 위 코드 예시를 들어서 자세히 설명하라. (5점)

1. 배열(Array)은 생성 후 크기가 고정이고, ArrayList는 크기가 동적으로 바뀜.
2. 배열(Array)은 int, float, double과 같은 primitive type과 Object type을 모두 사용가능 (iArray, pArray1 등), ArrayList는 Object type만 사용이 가능 (ArrayList<Integer>로 사용)
3. 배열(Array)은 요소를 추가할때 = 연산자를 사용하고, ArrayList는 add() 메소드를 사용
4. 배열(Array) 길이는 length를 사용하고, ArrayList 길이는 size() 메소드를 사용 for (int i=0; i<iArray; i++) 등. for (int i=0; i<pArray1.size(); i++) 사용

학과 _____ 학번 _____ 이름 _____

3. 다음은 클래스 상속관계에서 다형성을 보여주고 있다. 아래의 질문에 답하라. (40점)

```
enum Mode {
       B, C, D
}

interface I {
    void method1(int v);
}

interface J {
    void method2();
}

abstract class A implements I, J {
    protected Mode mode;
    protected int v;
    protected A(int v) {
       this.v = v;
    }
    public void method3() {
        System.out.println(this);
    }
    public String toString() {
        return "a";
    }
}

class B extends A {
    public B(int v) {
       super(v); // (3.2)
       this.mode = Mode.B; // (3.2)
    }
    public void method1(int v) {
        System.out.println("b1 v=" + v);
    }
    public void method2() {
        System.out.println("b2 mode=" + mode + " v="  + v);
    }
}

class C extends A {
    public C(int v) {
       super(v); // (3.2)
       this.mode = Mode.C; // (3.2)
    }
    public void method1(int v) {
        System.out.println("c1 v=" + v);
    }
    public void method2() {
        System.out.println("c2 mode=" + mode + " v="  + v);
    }
    public String toString() {
        return "c";
    }
}
```

학과 _____ 학번 _____ 이름 _____

```
public class D extends C {
    protected int v = 20;
    public D(int v) {
       super(v); // (3.2)
       this.mode = Mode.D; // (3.2)
    }
    public void method3() {
        System.out.println("d3 this.v=" + this.v + " super.v=" + super.v);
    }
}

public class Main {
    public static void main(String[] args) {
       // (3.3)
       //A a1 = new A(1);
       //A a2 = new B();
       //B a3 = a2;
       //D a4 = new D(3);
       //C a4 = new D(4);

       // (3.4)
       I i1 = new I() {
           public void method1(int v) {
              System.out.println("b1 v=" + v);
           }
       };
       i1.method1(100);
       I i2 = new B(0);
       i2.method1(100);
       J j1 = new J() {
           public void method2() {
              System.out.println("J");
           }
       };
       j1.method2();
       J j2 = new B(200);
       J2.method2();

       // (3.5)
       A[] elements = {new C(10), new B(20), new D(30)};
       for (int i = 0; i < elements.length; i++) {
              System.out.println(elements[i]);
              elements[i].method1(i);
              elements[i].method2();
              elements[i].method3();
       }
    }
}
```

3.1 abstract method가 무엇인지 간단히 설명하라. 위의 코드에서 예를 찾아서 보여라. (5점)

추상 메소드란 메소드를 선언만 해놓고 구현 내용은 없는 것으로, 추상 메소드를 상속받은 클래스는 반드시 추상메소드를 override해서 구현해야 한다. 추상 메소드를 상속받은 클래스가 추상 메소드를 구현하지 않으면 A처럼 추상 클래스가 되어야 한다.

위의 코드에서는 method1과 method2가 추상메소드이다.

학과 _____ 학번 _____ 이름 _____

3.2 클래스 B, C, D 생성자에서 사용되는 this와 super의 의미를 자세히 설명하라. (5점)

this는 나 자신 (객체)를 가리키는 레퍼런스이다.
super는 부모 (객체)를 가리키는 레퍼런스다.

B 생성자 안에 super(v)는 즉 부모클래스 생성자 A(int v) 호출하여, v 멤버필드를 v로 지정하겠다는 뜻.
this.mode = Mode.B;는 나의 mode 멤버필드를 Mode.B로 지정하겠다는 뜻.

3.3 main()의 (3.3)에서 compile error가 발생한다. 각각 그 이유를 설명하라. (10점)

```
//A a1 = new A(1); // cannot create A abstract class object
//A a2 = new B(); // compile error; no default B constructor.
//B a3 = a2; // compile error; cannot convert from A to B
//D a4 = new C(3); // compile error; cannot convert from C to D
//C a4 = new D(4); // Duplicate local variable a4
```

3.4 main()의 (3.4)에서는 무명 클래스 객체를 생성하고 있다. 실행 결과를 적어라. (10점)

```
b1 v=100 // i1의 method1을 호출
b1 v=100 // i2의 실제 객체는 B(0) 따라서 B의 method1을 호출
J // j1의 method2를 호출
b2 mode=B v=200 // j2의 실제 객체는 B(200) 따라서 B의 method2을 호출
```

3.5 main()의 (3.5) 실행 결과를 적고 이유를 자세히 설명하라 (동적바인딩 주의). (10점)

```
c // A.toString() => C.toString()
c1 v=0 // A.method1(0) => C.method1(0)
c2 mode=C v=10 // A.method2() => C.method2()
c // A.method3() 호출하는데 그 내부의 this는 c

a // A.toString() => B.toString() 없으므로 부모클래스것 A.toString() 호출
b1 v=1 // A.method1(1) => B.method1(1)
b2 mode=B v=20 // A.method2() => B.method2()
a // A.method3() 호출하는데 그 내부의 this는 b

c // A.toString() => D.toString() 없으므로 부모클래스것 C.toString() 호출
c1 v=2 // A.method1(2) => D.method1(2) 없으므로 C.method1(2) 호출
c2 mode=D v=30 // A.method2() => D.method2() 없으므로 C.method2() 호출 그러나 mode는 D
d3 this.v=20 super.v=30 // A.method3() => 오버라이딩된 D.method3() 호출
그러나 this.v는 클래스 D에서 재설정한 20이고, super.v는 D(30) 생성자에서 지정한 30
```

학과 _____ 학번 _____ 이름 _____

4. 다음 BMIPanel 클래스에 EventListener 구현을 (1)outer class를 사용하는 방식과 (2)inner class를 사용하는 방식으로 코드를 수정하여라. (10점)

```
// BMIPanelActionListener (outer class)
class BMIPanelActionListener implements ActionListener {
    BMIPanel panel = null;
    public BMIPanelActionListener(BMIPanel panel) {
        this.panel = panel;
    }
    public void actionPerformed(ActionEvent e) {
        panel.calculate();
    }
}
```

```
// BMIPanel
import java.awt.event.*;
import javax.swing.*;
public class BMIPanel extends JPanel implements ActionListener {
    JLabel label1 = new JLabel(new ImageIcon("bmi.png"));
    JLabel label2 = new JLabel("Age");
    JLabel label3 = new JLabel("Gender");
    JLabel label4 = new JLabel("Height");
    JLabel label5 = new JLabel("Weight (cm)"));
    JLabel label6 = new JLabel("BMI"));
    JTextField textfield2 = new NumberTextField(20);
    JTextField textfield4 = new NumberTextField(20);
    JTextField textfield5 = new NumberTextField(20);
    JRadioButton radio1 = new JRadioButton("MALE", true);
    JRadioButton radio2 = new JRadioButton("FEMALE");
    JComboBox<String> combobox1 = new JComboBox<String>(BMI.names());

    BMICalculator calc = new BMICalculator();
```

```
    // BMIPanelActionListener (inner class)
    private class ButtonActionListener implements ActionListener {

        public void actionPerformed(ActionEvent e) {
            calculate();
        }

    }
```

```
    public BMIPanel() {
       this.setLayout(null);

       label1.setBounds(30, 10, 240, 200);
       label2.setBounds(30, 230, 120, 30);
       label3.setBounds(30, 270, 120, 200);
       label4.setBounds(30, 310, 120, 200);
       label5.setBounds(30, 350, 120, 200);
       label6.setBounds(30, 450, 120, 200);

       textfield2.setBounds(120, 230, 160, 200);
       textfield4.setBounds(120, 310, 160, 200);
       textfield5.setBounds(120, 350, 160, 200);

       ButtonGroup rgroup = new ButtonGroup();
       rgroup.add(radio1);
       rgroup.add(radio2);
       radio1.setBounds(120, 270, 80, 30);
```

학과 _____ 학번 _____ 이름 _____

```
        radio2.setBounds(200, 270, 80, 30);

        button1.setBounds(30, 400, 250, 30);
        button1.addActionListener(this);

        combobox1.setBounds(120, 450, 160, 30);

        this.add(label1);
        this.add(label2);
        this.add(label3);
        this.add(label4);
        this.add(label5);
        this.add(label6);
        this.add(textfield2);
        this.add(textfield4);
        this.add(textfield6);
        this.add(radio1);
        this.add(radio2);
        this.add(button1);
        this.add(combobox1);
    }

    private Gender selectGender() {
        if (radio1.isSelected()) {
          return Gender.MALE;
        }
        return Gender.FEMALE;
    }

    private void calculate() {
        calc.setAge(Integer.parseInt(textfield2.getText()));
        calc.setHeight(Double.parseDouble(textfield4.getText()));
        calc.setAge(Double.parseDouble(textfield5.getText()));
        calc.setAge(selectGender());
        BMI bmi = calc.getBMI();
        Combobox1.setSelectedIndex(bmi.ordinal());
    }

    public void actionPerformed(ActionEvent e) {
       calculate(i); // 계산
    }
}
```

(1) outer class를 사용할 시

button1.addActionListener(new BMIPanelActionLister(this));

(2) inner class를 사용할 시

button1.addActionListener(new ButtonActionLister());

-끝-