

[lab08 보고서]

디지털미디어학과

2019111677

김지연

(이론문제, 1 ~ 3 번)

※ 다음과 같은 이벤트 처리 코드를 보고 답하세요.

```
class MyFrame extends JFrame implements ActionListener {
    JButton button;
    public MyFrame() {
        button = new JButton("버튼을 누르시오");
        button.addActionListener(this);
        ...
    }
    public void actionPerformed(ActionEvent e) {
        System.out.println("마침내 버튼이 눌러졌습니다.");
    }
}
```

1. 이벤트 처리 코드를 무명 클래스로 다시 작성하세요.

```
class MyFrame extends JFrame {
    JButton button;
    public MyFrame() {
        button = new JButton("버튼을 누르시오");
        button.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                System.out.println("마침내 버튼이 눌러졌습니다.");
            }
        });
    }
}
```

2. 이벤트 처리 코드를 내부 클래스로 다시 작성하세요.

```
class MyFrame extends JFrame {
    JButton button;
    public MyFrame() {
        button = new JButton("버튼을 누르시오");
        button.addActionListener(new MyListener());
        ...
    }

    private class MyListener implements ActionListener{
        public void actionPerformed(ActionEvent e) {
            System.out.println("마침내 버튼이 눌러졌습니다.");
        }
    }
}
```

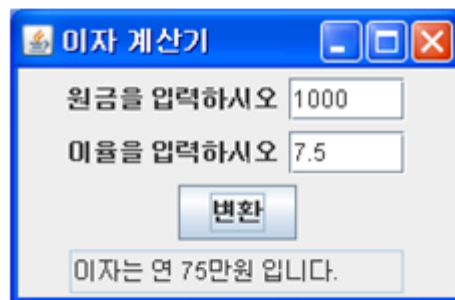
3. 이벤트 처리 코드를 람다식으로 다시 작성하세요.

```
class MyFrame extends JFrame {  
    JButton button;  
    public MyFrame() {  
        button = new JButton("버튼을 누르시오");  
        button.addActionListener((ActionEvent e)->{  
            System.out.println("마침내 버튼이 눌러졌습니다.");  
        });  
    }  
}
```

(실습문제 4, 5번)

4. 아래 그림을 참고해서, 원금과 이율을 입력하면 이자를 계산하여 주는 프로그램을 작성하세요.

(“변환” 버튼을 누르면 하단 텍스트필드에 연 이자 출력)



[분석]

원금과 이율을 입력하고 변환 버튼을 누르면 연 이자를 계산해주는 프로그램

- 화면 디자인: 레이블2개, 텍스트필드 3개, 버튼 1개, 패널 1개를 생성한다. MyFram생성자에 프레임 선언한다. 레이블 2개에 각각의 레이블을 입력해주고 텍스트필드를 5정도로 생성해준다. 변환버튼을 생성하고 이벤트 리스너를 등록해준다. 필드3은 15정도로 생성하고 비활성화하여 텍스트를 입력할 수 없게 해준다. 패널을 생성하고 패널에 각각의 컴포넌트들을 추가하고 활성화한다.

- 이벤트 처리: 리스너 클래스를 내부 클래스로 작성하는 방법으로 생성한다. 멤버 변수들을 자유롭게 사용할 수 있다는 장점이 있다. 액션리스너는 종류가 인터페이스이고 actionPerformed() 메소드를 가지고 있어서 사용자가 버튼을 누를 때마다 이 메소드가 실행된다. 이벤트 객체를 생성하고, 이벤트 처리 코드를 구현한다. 이벤트 처리 코드는 만약 button1이 눌렀다면 필드3에 연이자가 몇인지 알려주는 문장이 나오도록 한다. 연이자 계산은 원금*X*이율/100으로 계산하는데 원금과 이자가 문자열로 입력되기 때문에 수치 값으로 변환해준다. 원금은 정수, 이율은 실수로 변환한다. 전체 값은 다시 정수로 형 변환해준다.

[소스코드]

```
import javax.swing.*;
import java.awt.event.*;
import java.awt.*;

class MyFrame extends JFrame{

    private JLabel label1, label2;
    private JTextField field1, field2, field3;
    private JButton button1;
    private JPanel panel;

    public MyFrame() {
        this.setTitle("이자 계산기");
        this.setSize(240,200);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        label1 = new JLabel("원금을 입력하십시오");
        field1 = new JTextField(5);

        label2 = new JLabel("이율을 입력하십시오");
        field2 = new JTextField(5);

        button1 = new JButton("변환");
        button1.addActionListener(new MyListener());

        field3 = new JTextField(15);
        field3.setEnabled(false);
        //field3.setForeground(Color.pink); //비활성화 상태라 색이 바뀌지는
        //않음

        JPanel panel = new JPanel();
        panel.add(label1);
        panel.add(field1);
        panel.add(label2);
        panel.add(field2);
        panel.add(button1);
        panel.add(field3);
        this.add(panel);

        this.setVisible(true);

    }

    private class MyListener implements ActionListener{
        public void actionPerformed(ActionEvent e) {
            if(e.getSource() == button1) {
                field3.setText("이자는 연" + (int)
(Integer.parseInt(field1.getText())*Double.parseDouble(field2.getText())/100) +
"만원 입니다.");
            }
        }
    }
}
```

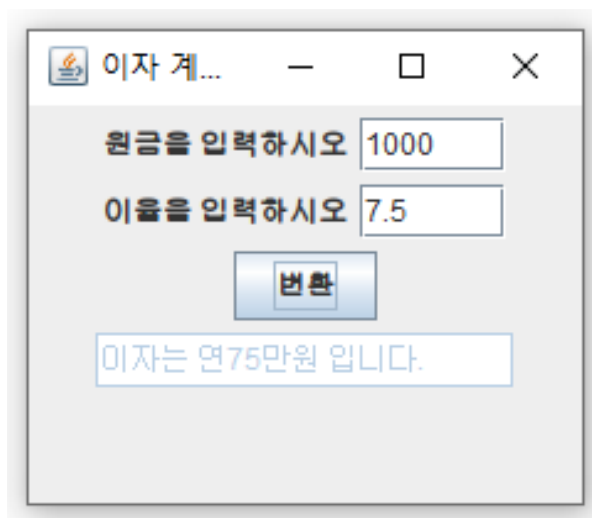
```

    }
}

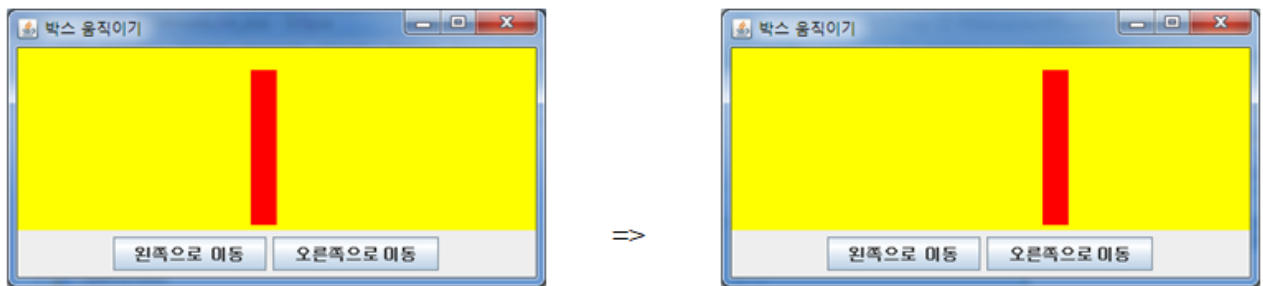
public class Calculator {
    public static void main(String[] args) {
        MyFrame e = new MyFrame();
    }
}

```

[결과화면]



5. 아래 그림을 참고해서, 버튼을 누르면 화면의 막대가 좌우로 움직이는 프로그램을 작성하세요.



[분석]

버튼을 누르면 막대가 움직이는 프로그램

- 화면 디자인: 패널 2개를 생성하여 BorderLayout으로 panel1은 CENTER에 panel2는 SOUTH에 배치한다. 패널1에는 막대기(도형)를 중앙에 배치한다. 패널1을 MyPanel로 생성하고 MyPanel을 JPanel을 상속받도록 하여 패널1에 도형을 배치한다. 도형은 paint메소드를 활용하여 색과 위치를 지정해준다. 위치는 x와 y를 정적 매개변수로 선언하여 지정해준다. 패널2에는 도형을 왼쪽으로 움직이게 하는 버튼 1개와 오른쪽으로 움직이게 하는 버튼 1개를 생성한다. 버튼에는 이벤트 리스너를 등록하여 버튼을 누를 때마다 막대기가 움직이도록 한다.

- 이벤트 처리: Move클래스가 JFrame을 상속받으면서 동시에 ActionListener인터페이스도 구현하도록 한다. Move클래스 안에 actionPerformed()를 정의한다. 이벤트 객체를 생성하고, 이벤트 처리 코드를 구현한다. 버튼1을 누르면 x의 위치가 -20이 되도록 하고 paint()메소드를 repaint()메소드를 이용해 수동으로 호출한다. 버튼2도 누르면 x의 위치가 +20이 되도록 하고 paint()메소드를 호출해준다.

move생성자를 호출한다.

[소스코드]

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class Move extends JFrame implements ActionListener {
    static int x = 330, y = 40;
    JPanel panel1 = new MyPanel();
    JPanel panel2 = new JPanel();
    JButton button1, button2;

    class MyPanel extends JPanel{
        public void paint(Graphics g) {
            super.paint(g);
            g.setColor(Color.red);
            g.fillRect(x,y,30,200);
        }
    }
}
```

```

    }
}

Move(){
    setSize(700,330);
    setTitle("박스 움직이기");
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setLayout(new BorderLayout());

    panel1.setSize(700,300);
    panel1.setBackground(Color.yellow);
    panel2.setSize(700,100);

    button1 = new JButton("왼쪽으로 이동");
    button2 = new JButton("오른쪽으로 이동");
    button1.addActionListener(this);
    button2.addActionListener(this);

    panel2.add(button1);
    panel2.add(button2);
    add(panel1, BorderLayout.CENTER);
    add(panel2, BorderLayout.SOUTH);
    setVisible(true);
}

public void actionPerformed(ActionEvent e) {
    if(e.getSource() == button1) {
        x -= 20;
        repaint();
    }
    else if(e.getSource() == button2) {
        x += 20;
        repaint();
    }
}

public static void main(String[] args) {
    new Move();
}
}

```

[결과화면]

