

객체지향프로그래밍 - 13주차 실습 답안지

성명: 조우연

학과: 수학과

학번: 201921195

※ 본 실습활동지에 대한 보고서나 코드를 작성함에 있어 교재나 강의노트를 제외한 다른 자료로부터 일부 또는 전체를 복사하였습니까? 예() 아니오(O)

1. ButtonTest를 이용하여 다음 물음에 답하시오.

(가) ButtonFrame.java에서는 ColorAction inner class로 ActionListener를 구현하여 button의 listener로 등록한다. 이를 Lambda expression으로 수정하시오.

- 수정한 코드

```
public class ButtonFrame extends JFrame {
    private JPanel buttonPanel;
    private static final int DEFAULT_WIDTH = 300;
    private static final int DEFAULT_HEIGHT = 200;

    public ButtonFrame()
    {
        setSize(DEFAULT_WIDTH, DEFAULT_HEIGHT);

        //creat buttons
        JButton yellowButton = new JButton("Yellow");
        JButton blueButton = new JButton("Blue");
        JButton redButton = new JButton("Red");
        buttonPanel = new JPanel();
        //add buttons to panel
        buttonPanel.add(yellowButton);
        buttonPanel.add(blueButton);
        buttonPanel.add(redButton);
        //add panel to frame
        add(buttonPanel);

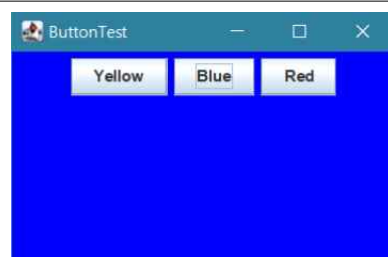
        //associate actions with buttons
        yellowButton.addActionListener(event->buttonPanel.setBackground(Color.YELLOW));
        blueButton.addActionListener(event->buttonPanel.setBackground(Color.BLUE));
        redButton.addActionListener(event->buttonPanel.setBackground(Color.red));
    }
}
```

- 수행결과

yellow 클릭시 결과



blue 클릭시 결과



red 클릭시 결과



2. DrawTest와 ActionTest를 각각 수행해 본 후 다음 물음에 답하시오.

(가) 모든 도형이 파란색으로 그려지도록 DrawTest.java를 수정하시오.

- 수정한 코드

```
class DrawComponent extends JComponent
{
    private static final int DEFAULT_WIDTH = 400;
    private static final int DEFAULT_HEIGHT = 400;
    private JPanel panel;

    public void paintComponent(Graphics g)
    {
        Graphics2D g2 = (Graphics2D)g;

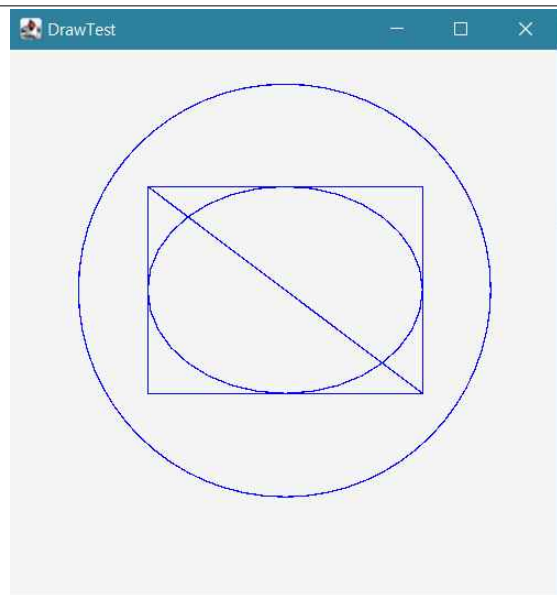
        //draw a rectangle
        double leftX = 100;
        double topY=100;
        double width = 200;
        double height = 150;

        //색상 블루로 바꾸기
        g.setColor(Color.BLUE);

        Rectangle2D rect = new Rectangle2D.Double(leftX,topY,width,height);
        g2.draw(rect);
    }
}
```

주어진 DrawComponent class에 형광펜으로 표시한 코드를 추가한 것임! DrawComponent class의 뒷부분 코드는 기존 코드와 동일하므로 생략함.

- 수행결과



(나) ActionFrame.java를 이용하여 다음과 같이 수정하시오.

- Action Frame.java.에서 정의한 Frame에 button panel 외에 DrawTest.java에서 정의한 DrawComponent를 하나 만들어 추가한다.

- 수정한 코드

```
public class ActionFrame extends JFrame {

    private JPanel buttonPanel;
    private static final int DEFAULT_WIDTH = 300;
    private static final int DEFAULT_HEIGHT = 200;

    public ActionFrame()
    {
        setSize(DEFAULT_WIDTH, DEFAULT_HEIGHT);
        buttonPanel = new JPanel();
        //define actions
        Action yellowAction = new ColorAction("Yellow", new ImageIcon("yellow-ball.gif"),Color.YELLOW);
        Action blueAction = new ColorAction("Blue", new ImageIcon("blud-ball.gif"),Color.BLUE);
        Action redAction = new ColorAction("Red", new ImageIcon("red-ball.gif"),Color.RED);

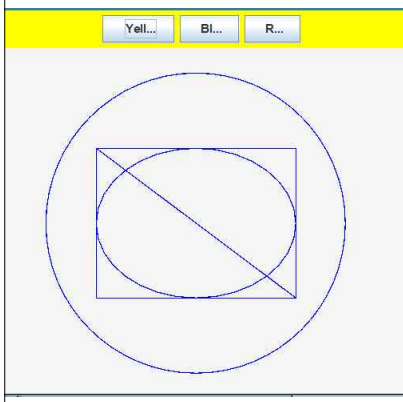
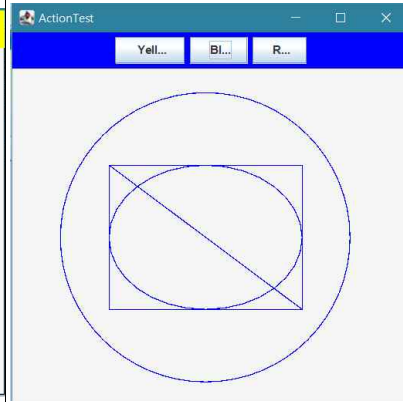
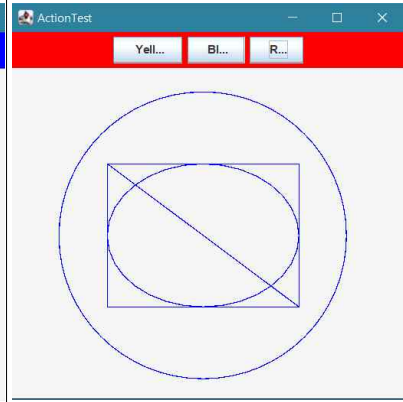
        //add buttons for these actions
        buttonPanel.add(new JButton(yellowAction));
        buttonPanel.add(new JButton(blueAction));
        buttonPanel.add(new JButton(redAction));

        //add panel to frame
        add(buttonPanel,BorderLayout.NORTH);
        add(new DrawComponent(),BorderLayout.CENTER);

        //associate the Y,B and R keys with names
        InputMap imap = buttonPanel.getInputMap(JComponent.WHEN_ANCESTOR_OF_FOCUSED_COMPONENT);
        imap.put(KeyStroke.getKeyStroke("ctrl Y"),"panel.yellow");
        imap.put(KeyStroke.getKeyStroke("ctrl B"),"panel.blue");
        imap.put(KeyStroke.getKeyStroke("ctrl R"),"panel.red");
    }
}
```

주어진 ActionFrame class에 형광펜으로 표시한 코드를 추가한 것! ActionFrame class의 뒷부분 코드는 기존 코드와 동일하므로 생략함.

- 수행결과

yellow 클릭 시 결과	blue 클릭 시 결과	red 클릭 시 결과
		

(가)코드를 변형하여 작성하였기 때문에 도형이 파란색으로 나타난다.

- Button을 누를 때 마다 DrawComponent에 그려진 도형이 button의 color로 변경되도록 한다.

- 수정한 코드

```
public class ActionFrame extends JFrame {
    private JPanel buttonPanel;
    private static final int DEFAULT_WIDTH = 300;
    private static final int DEFAULT_HEIGHT = 200;
    private Color color;

    public ActionFrame()
    {
        setSize(DEFAULT_WIDTH, DEFAULT_HEIGHT);
        buttonPanel = new JPanel();
        //define actions
        Action yellowAction = new ColorAction("Yellow", new ImageIcon("yellow-ball.gif"), Color.YELLOW);
        Action blueAction = new ColorAction("Blue", new ImageIcon("blue-ball.gif"), Color.BLUE);
        Action redAction = new ColorAction("Red", new ImageIcon("red-ball.gif"), Color.RED);

        //add buttons for these actions
        buttonPanel.add(new JButton(yellowAction));
        buttonPanel.add(new JButton(blueAction));
        buttonPanel.add(new JButton(redAction));

        //add panel to frame
        add(buttonPanel, BorderLayout.NORTH);
        add(new DrawComponent(), BorderLayout.CENTER);

        //associate the Y, B and R keys with names
        InputMap imap = buttonPanel.getInputMap(JComponent.WHEN_ANCESTOR_OF_FOCUSED_COMPONENT);
        imap.put(KeyStroke.getKeyStroke("ctrl Y"), "panel.yellow");
        imap.put(KeyStroke.getKeyStroke("ctrl B"), "panel.blue");
        imap.put(KeyStroke.getKeyStroke("ctrl R"), "panel.red");

        //associate the names with actions
        ActionMap amap = buttonPanel.getActionMap();
        amap.put("panel.yellow", yellowAction);
        amap.put("panel.blue", blueAction);
        amap.put("panel.red", redAction);
    }

    public class ColorAction extends AbstractAction {
        public ColorAction(String name, Icon icon, Color c)
        {
            putValue(Action.NAME, name);
            putValue(Action.SMALL_ICON, icon);
            putValue(Action.SHORT_DESCRIPTION, "Set panel color to "+name.toLowerCase());
            putValue("color", c);
        }

        public void actionPerformed(ActionEvent event)
        {
            Color c = (Color)getValue("color");
            buttonPanel.setBackground(c);
            color = c;
            repaint();
        }
    }
}

class DrawComponent extends JComponent
{
    private static final int DEFAULT_WIDTH = 400;
    private static final int DEFAULT_HEIGHT = 400;
    private JPanel panel;

    public void paintComponent(Graphics g)
    {
        Graphics2D g2 = (Graphics2D)g;

        //draw a rectangle
        double leftX = 100;
        double topY = 100;
        double width = 200;
        double height = 150;

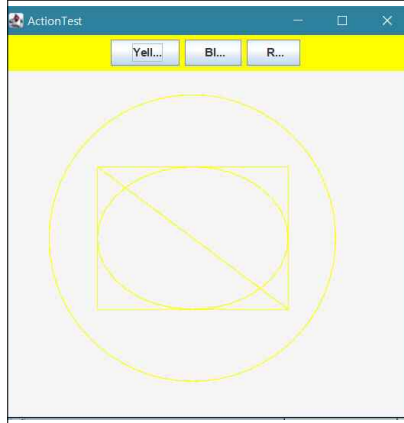
        //draw rectangle
        g.setColor(color);

        Rectangle2D rect = new Rectangle2D.Double(leftX, topY, width, height);
        g2.draw(rect);
    }
}
```

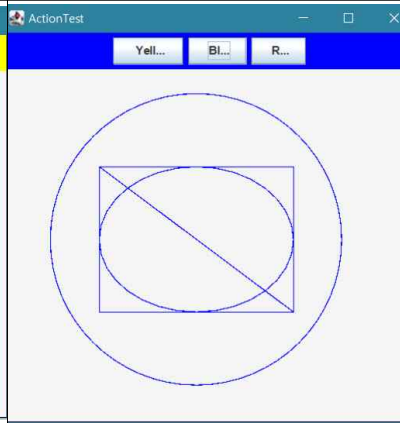
기존의 코드에 형광펜으로 표시한 코드가 수정된 코드이다. DrawComponent의 코드 뒷부분은 기존과 동일하므로 생략

- 수행결과

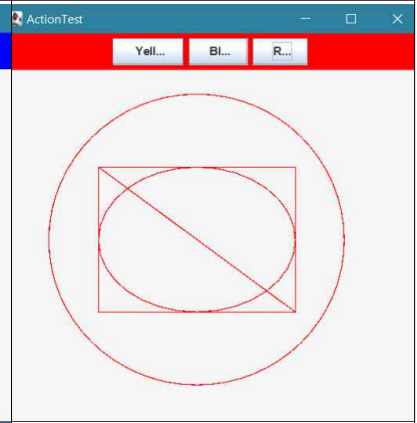
yellow 클릭시 결과



blue 클릭시 결과



red 클릭시 결과



3. MouseTest를 실행해 본 후 다음 물음에 답하시오.

(가) double click할 때 사각형이 지워지는 대신 오른쪽 마우스 버튼을 누를 경우 지워지도록 수정하시오.

- 수정 코드

```
private class MouseHandler extends MouseAdapter //MouseAdapter implements MouseListener
{
    public void mousePressed(MouseEvent event)
    {
        //add a new square if the cursor isn't inside a square
        current=find(event.getPoint());
        if(current==null)add(event.getPoint());
        if(current!=null && (event.getModifiersEx() & InputEvent.BUTTON3_DOWN_MASK)!=0)
            remove(current);
    }

    public void mouseClicked(MouseEvent event)
    {
        //remove the current square if double clicked
        current = find(event.getPoint());

        //if(current!=null && event.getClickCount()>=2)
        //    remove(current);
    }
}
```

더블 클릭 시 사각형이 지워지는 코드를 삭제하고, 오른쪽 마우스 버튼을 누를 시 사각형이 지워지는 코드를 추가하였다.

(나) click하면 모든 사각형이 지워지는 Button을 추가하시오.

- 수정한 코드 (in MouseFrame class)

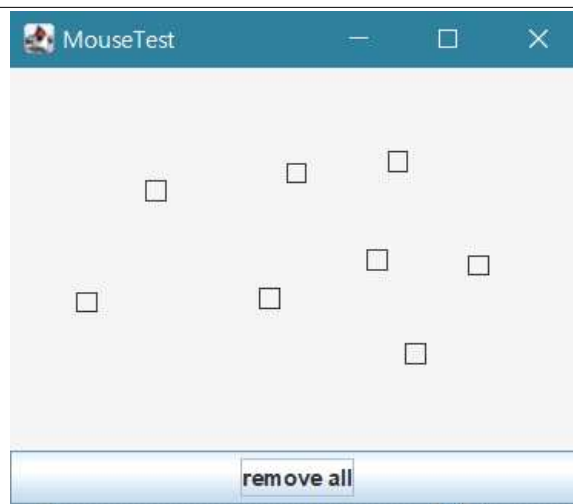
```
public class MouseFrame extends JFrame {  
  
    public MouseFrame()  
    {  
        JButton removeButton = new JButton("remove all");  
        add(removeButton, BorderLayout.SOUTH);  
        MouseComponent MouseComp = new MouseComponent();  
        add(MouseComp);  
        removeButton.addActionListener(event->MouseComp.removeall());  
        pack();  
    }  
}
```

- 추가한 코드 (in MouseComponent class as method)

```
public void removeall()  
{  
    squares.clear();  
    repaint();  
}
```

- 실행결과

remove all button click 전



remove all button click 후

