

1811943 황성미 9일차 과제

▼ 5.3.8 JTabbedPane

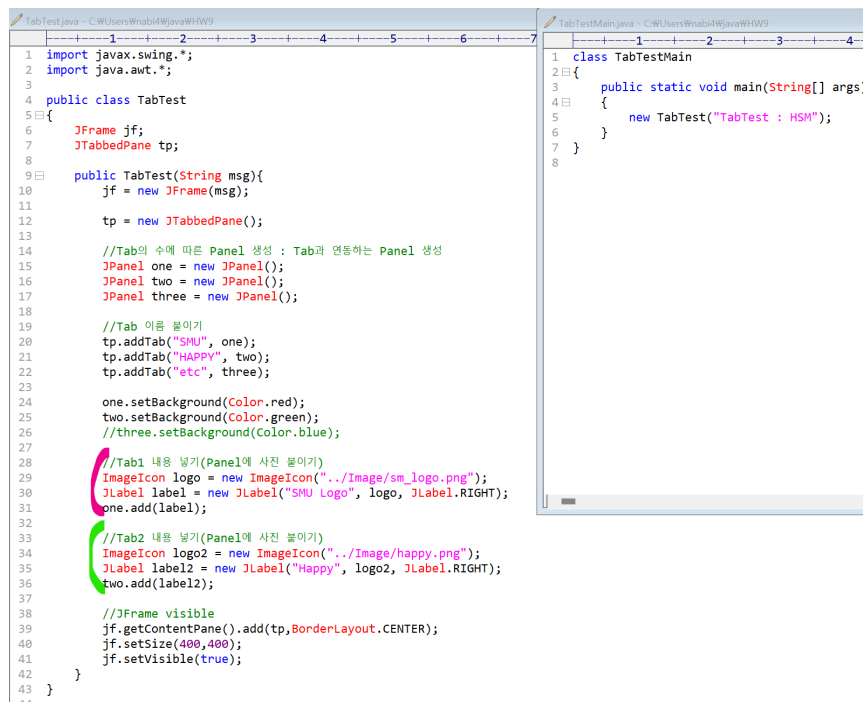
▼ Constructor

JTabbedPane(), JTabbedPane(int tabPlacement)

▼ Method

addTab(String title, Icon icon, Component comp), addTab(String title, Component comp), insertTab(String title, Component com), remove(Component comp)

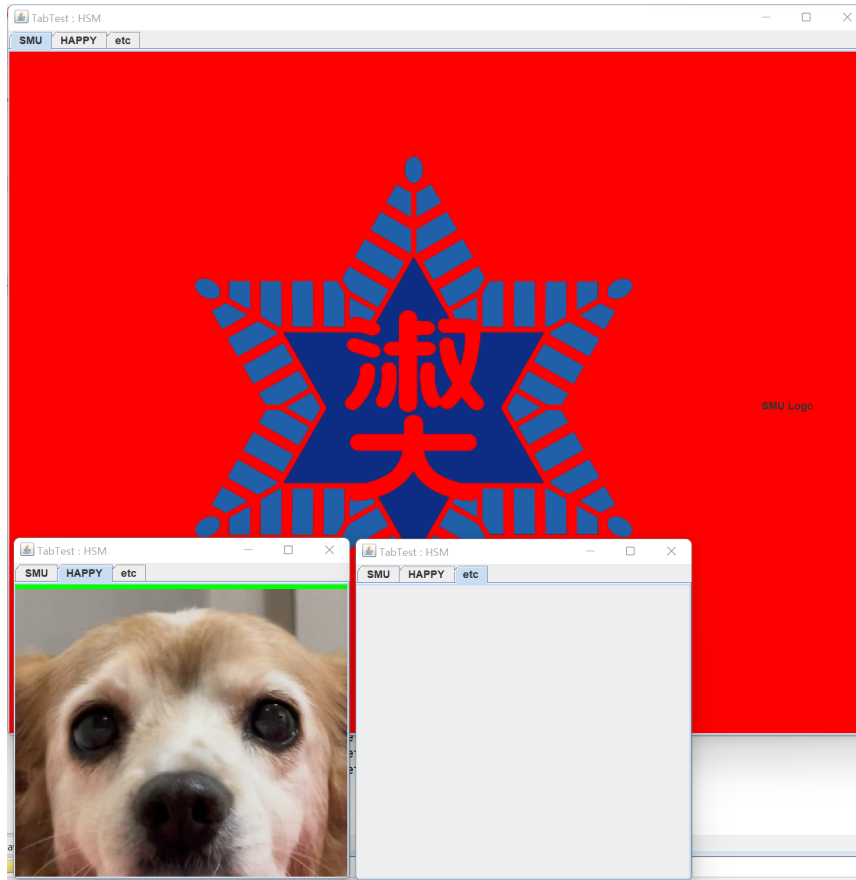
▼ Lab



```
1 import javax.swing.*;
2 import java.awt.*;
3
4 public class TabTest
5 {
6     JFrame jf;
7     JTabbedPane tp;
8
9     public TabTest(String msg){
10         jf = new JFrame(msg);
11
12         tp = new JTabbedPane();
13
14         //Tab의 수에 따른 Panel 생성 : Tab과 연동하는 Panel 생성
15         JPanel one = new JPanel();
16         JPanel two = new JPanel();
17         JPanel three = new JPanel();
18
19         //Tab 이름 붙이기
20         tp.addTab("SMU", one);
21         tp.addTab("HAPPY", two);
22         tp.addTab("etc", three);
23
24         one.setBackground(Color.red);
25         two.setBackground(Color.green);
26         three.setBackground(Color.blue);
27
28         //Tab1 내용 넣기(Panels에 사진 붙이기)
29         ImageIcon logo = new ImageIcon("../Image/sm_logo.png");
30         JLabel label = new JLabel("SMU Logo", logo, JLabel.RIGHT);
31         one.add(label);
32
33         //Tab2 내용 넣기(Panels에 사진 붙이기)
34         ImageIcon logo2 = new ImageIcon("../Image/happy.png");
35         JLabel label2 = new JLabel("Happy", logo2, JLabel.RIGHT);
36         two.add(label2);
37
38         //JFrame visible
39         jf.getContentPane().add(tp, BorderLayout.CENTER);
40         jf.setSize(400, 400);
41         jf.setVisible(true);
42     }
43 }
```

```
1 class TabTestMain
2 {
3     public static void main(String[] args)
4     {
5         new TabTest("TabTest : HSM");
6     }
7 }
8
```

각각의 tab마다 달라지는 내용을 볼 수 있음



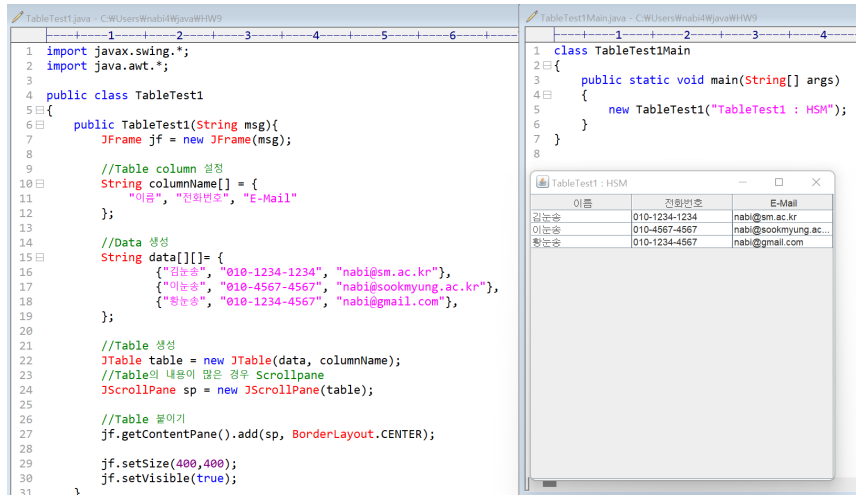
▼ 5.3.9 JTable(+ Table Model pass)

▼ Constructor

JTable() , JTable(int numRows, int numColumns), JTable(Object[][] rowData, Object[][] columnNames)

▼ 기능을 확장해서 사용하려면 TableModel까지 알아야하는데 이후에 더 공부해보자!

▼ Lab



데이터 값을 미리 집어 넣어 고정된 테이블을 만들어볼 수 있음

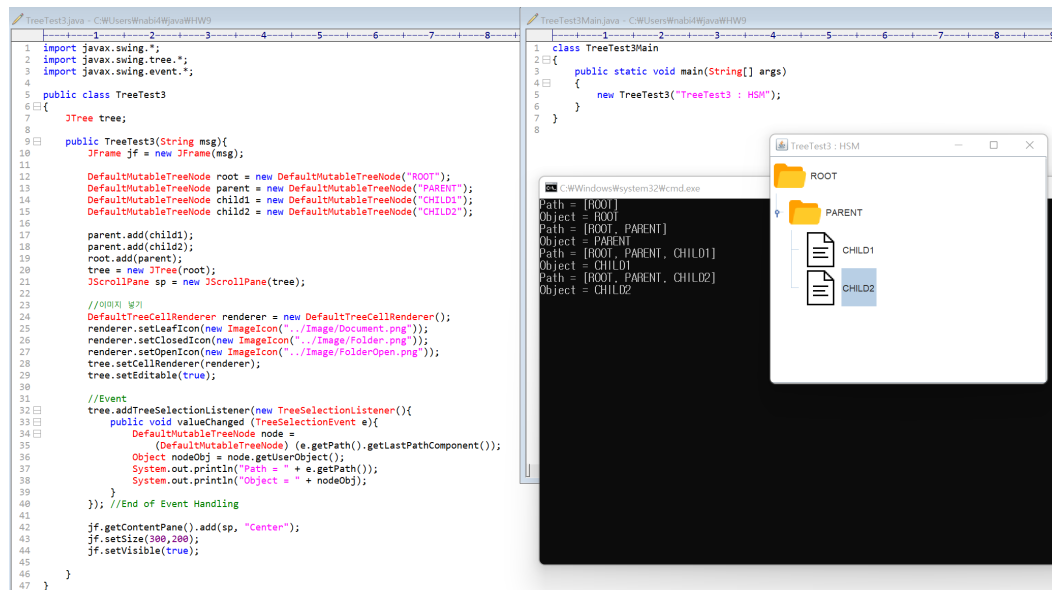
▼ 5.3.10 JTree

Model, Node interface가 있는 형태

▼ Constructor

JTree(), JTree(Object[]), JTree(TreeNode), JTree(TreeModel)

▼ Lab



Icon과 각각을 클릭하면 Event handling이 들어가도록 구현

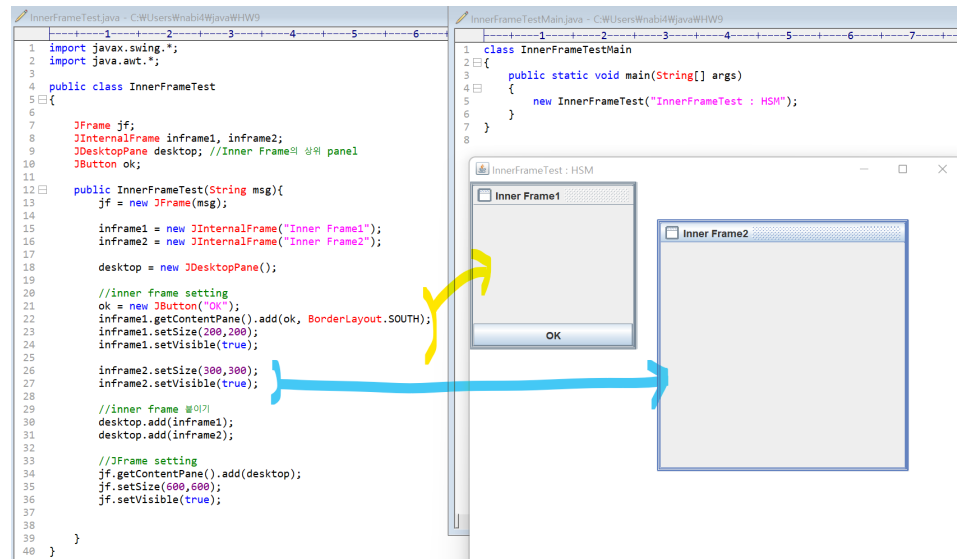
▼ 5.3.11 JInternalFrame

프레임 안에 프레임을 집어넣음

▼ Constructor

JInternalFrame(), JInternalFrame(String)

▼ Lab

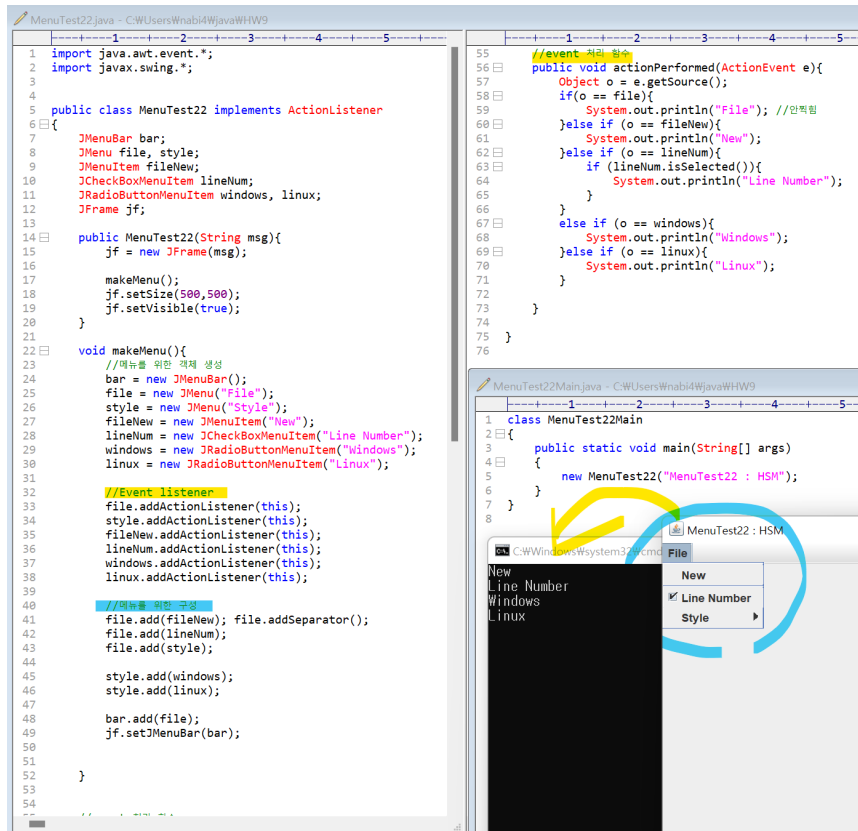


프레임 안에 JInternalFrame을 집어넣어 여러 창을 띄울 수 있음. 실행시키면 두 창이 겹쳐 보이는데
처음부터 창에 간격을 띄어두면 더 좋을듯!

▼ 5.4 JMenu

AWT 와 동일

▼ Lab



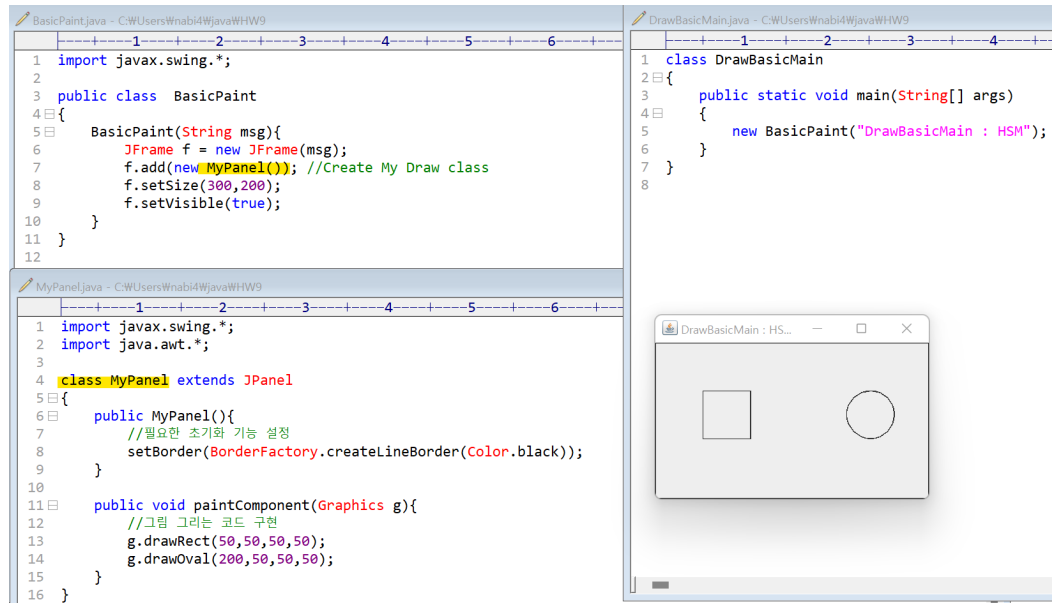
Menubar를 구성한 후 클릭하면 그 값을 보여주는 이벤트 핸들링을 구현

▼ 5.5 JDialog

프레임과 같이 창을 포함시키는 역할. 보통 팝업처럼 뜨게 하는 용도로 많이 쓰임

▼ Lab

▼ Lab



그리려는 것들을 코드로 구현한 후 (MyPanel.java) 그리려는 창에서 (BasicPaint.java) MyPanel을 호출함
그리기 위해선 paint의 메서드를 활용해야하는데 Swing에선 paintComponent()를, AWT에선 paint()를 이용함.
그림을 그리는 코드를 구현할 때 좌표값을 어떻게 설정하냐 / 어떤 메서드를 사용했는지에 따라 그림이 달라짐

▼ 6.2 Basic Shape

draw() : 그리는거, fill() : 채우는거

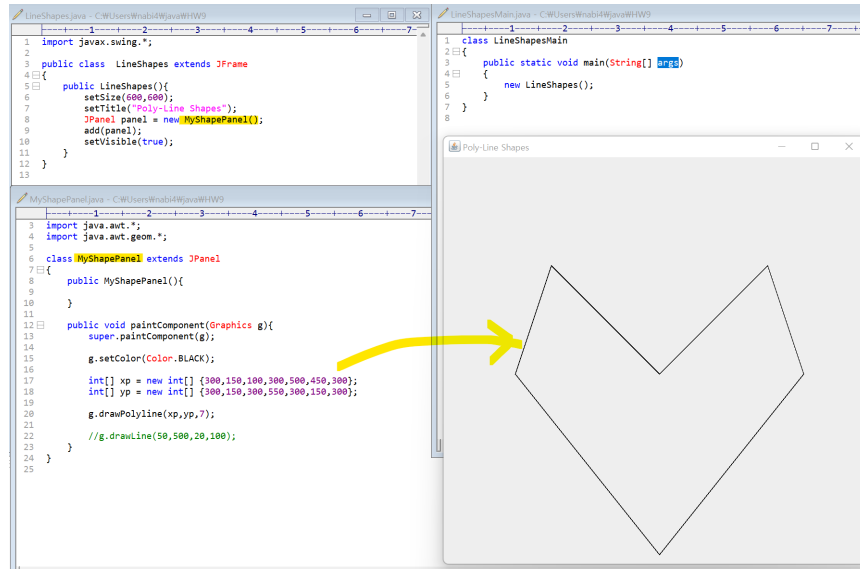
▼ method

변수들 : 좌표값(그릴 때 중요)

Basic Shape	methods
Line	drawLine(), drawPolyline()
Rectangle	drawRect(), fillRect(), clearRect()
3D Rectangle	draw3DRect(), fill3DRect(),
Round Rectangle	drawRoundRect(), fillRoundRect(),
Oval	drawOval(), fillOval(),
Arc	drawArc(), fillArc()
Polygon	drawPolygon(), fillPolygon()

추가 메서드들은 필요한 상황마다 검색해서 사용하기!

▼ Lab(drawPolyline)



원하는 그림을 MyShapepanel()에 코드로 구현한 후 이를 띄울 창을 (LineShapes.java) 구현

▼ PC는 창이 크기 때문에 panel을 여러개 띄워서 창을 분리시키는 것이 좋음

▼ 6.2.2 Rectangle draw

drawRect(int x, int y, int width, int height), fillRect(int x, int y, int width, int height) → 채워진 사각형, draw3DRect(int x, int y, int width, int height, boolean raised) → 3D 사각형, fill3DRect(int x, int y, int width, int height, boolean raised) → 채워진 3D 사각형

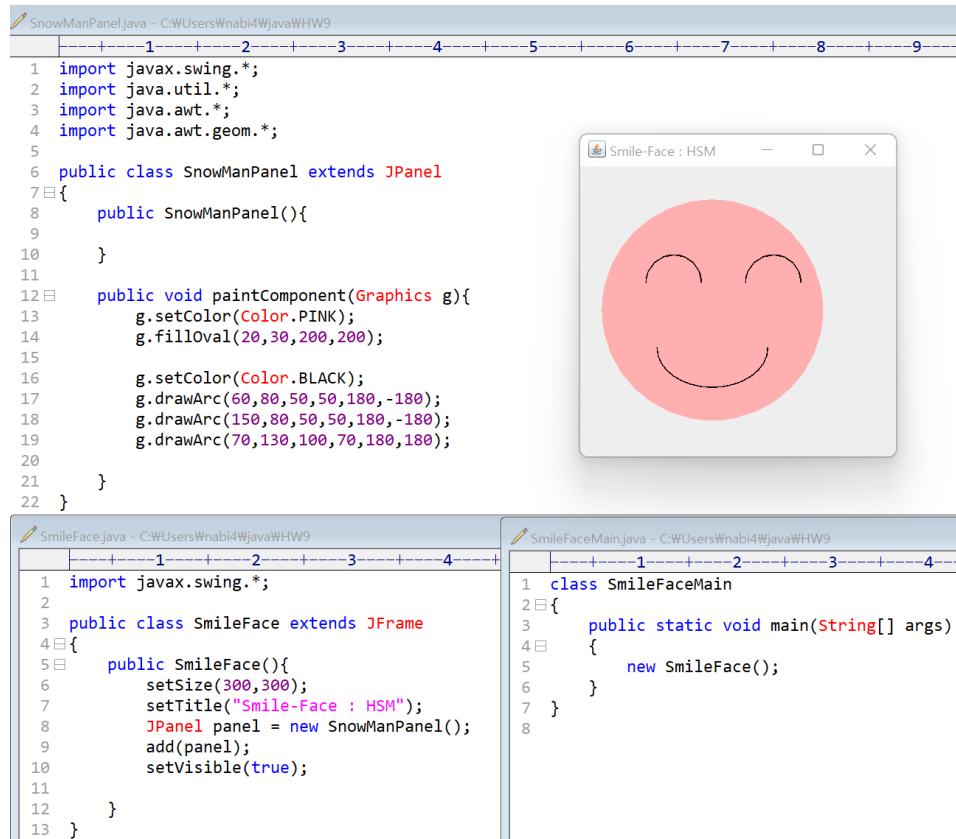
▼ 6.2.3 Oval draw

drawOval(int x, int y, int width, int height), fillOval(int x, int y, int width, int height)

▼ 6.2.4 Arc draw

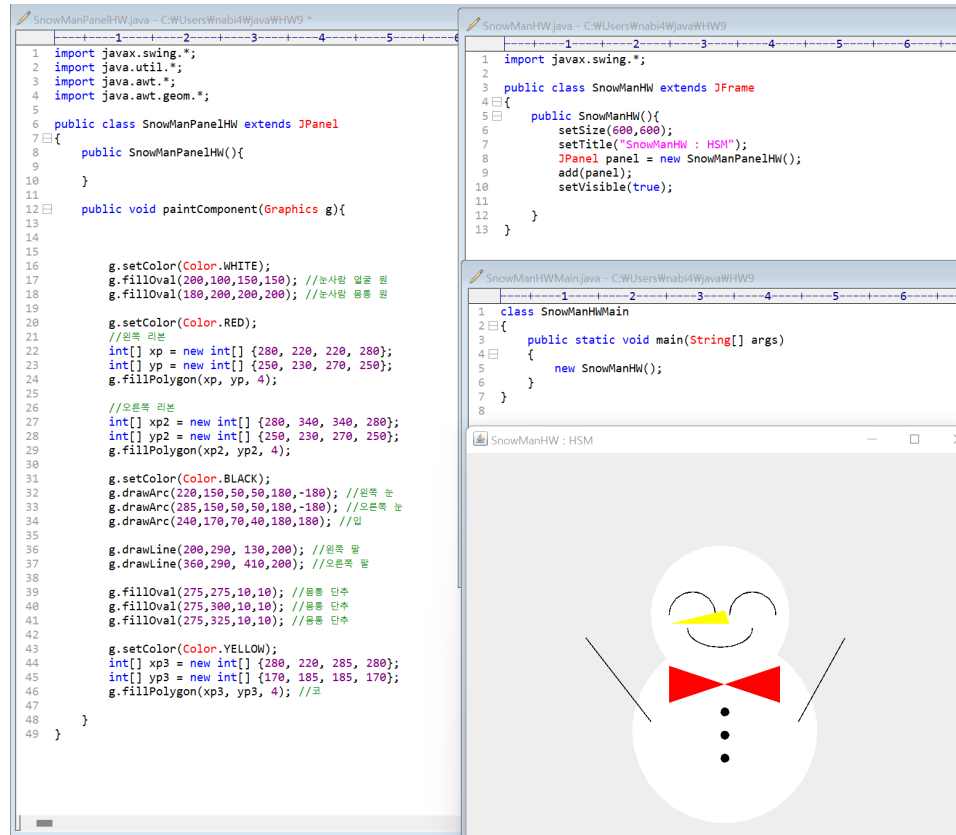
drawArc(int x, int y, int width, int height, int startAngle, int arcAngle), fillOval(int x, int y, int width, int height, int startAngle, int arcAngle)

▼ Lab



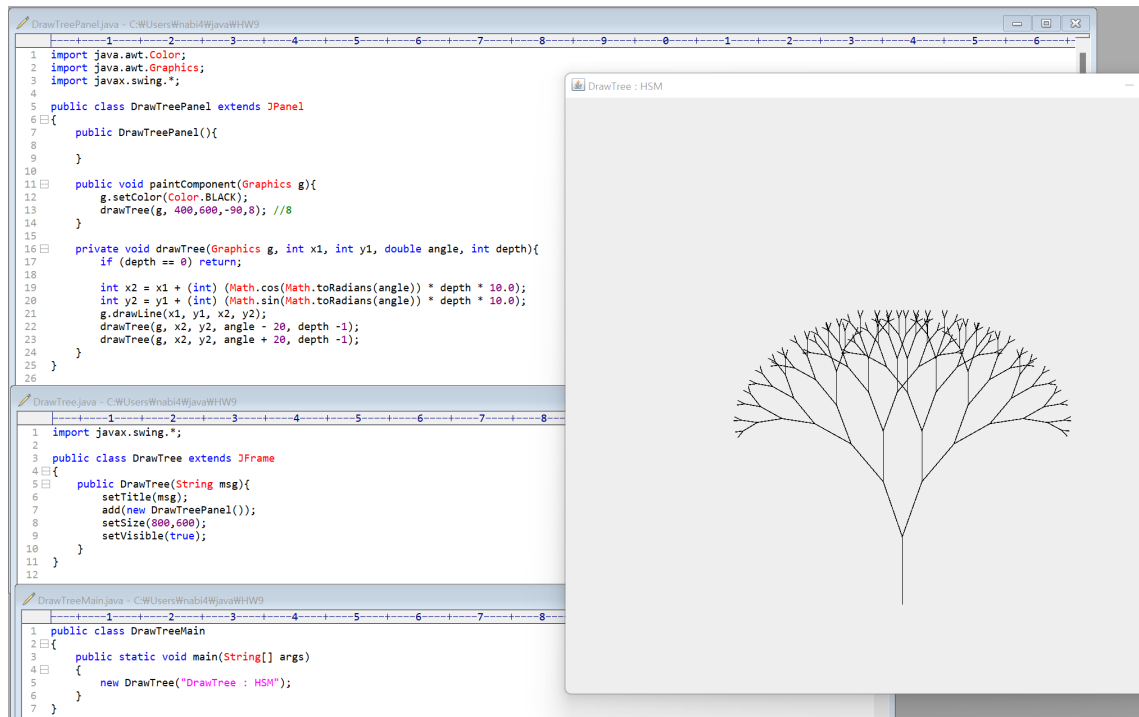
입력변수 순서가 startAngle, arcAngle인 것 주의!!
startAngle 부터 arcAngle까지 그림을 그려줌

▼ HW(Snowman)



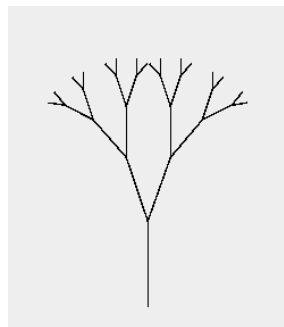
눈사람 얼굴과 몸통을 만든 후,
 위 lab에서 실시한 얼굴 모양을 눈사람의 크기에 맞게 조절.
 그 후 손과 코, 몸통의 버튼, 리본을 추가적으로 만들어주었음!

▼ Lab2(Fractal)



drawTree라는 함수를 생성하여 프랙탈을 그리는 함수를 정의해주고 이를 그리면 오른쪽과 같은 프랙탈을 얻어낼 수 있음

▼ 8(depth) 숫자에 따라서 가짓수가 달라짐



depth를 5로 준 경우