

1811943 황성미 10일차 과제

▼ 6.3 Color

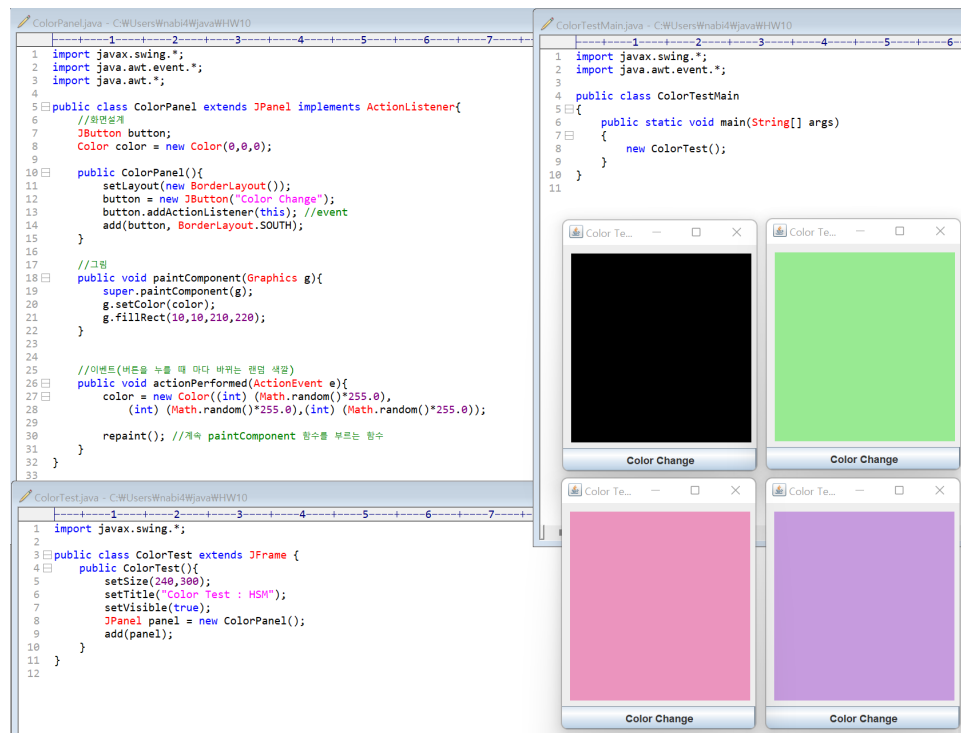
java.awt.Color class

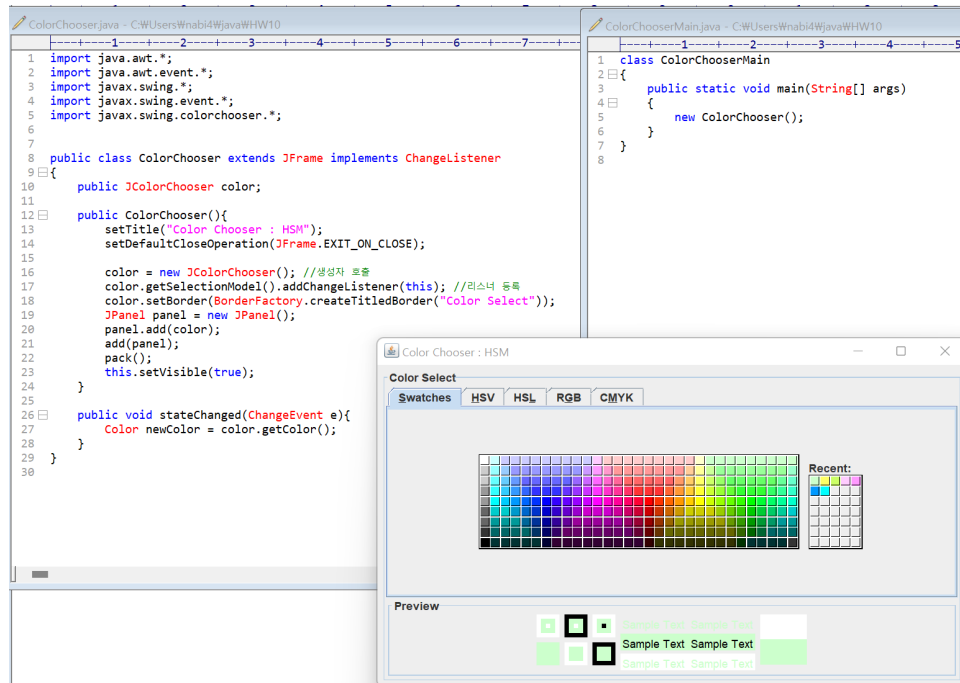
고정된 값을 쓰거나 RGB 값을 사용해도 됨

▼ Method

setBackground(Color c); , setColor(Color c); , getColor()

▼ Lab





Color Chart가 나와서 직접 팔레트에서 색을 고를 수 있음

▼ 6.4 Java 2D

line, rectangle 등 다양한 모양을 그릴 수 있음 → 차트 그리는데에 활용 가능!

java.awt.Graphics , java.awt.Graphics2D, java.awt.geom

▼ Rendering Context

context 변형도 가능

▼ Method

setStroke(Stroke s), setPaint(Paint paint), setComposite(Composite comp),
setTransform(Transform x), setFont(Font font)

▼ java.awt.geom.package

많은 클래스를 이용하여 다양한 모양을 그릴 수 있음

▼ Method

draw~~~2D(), fill~~~2D()

▼ Draw 2D line

▼ Rectangle 2D

new Rectangle2D.Float(float height, float width, float x, float y), new
RoundRectangle2D.Float(float archeight, float arcwidth, float height, float
width, float x, float y)

▼ Ellipse 2D

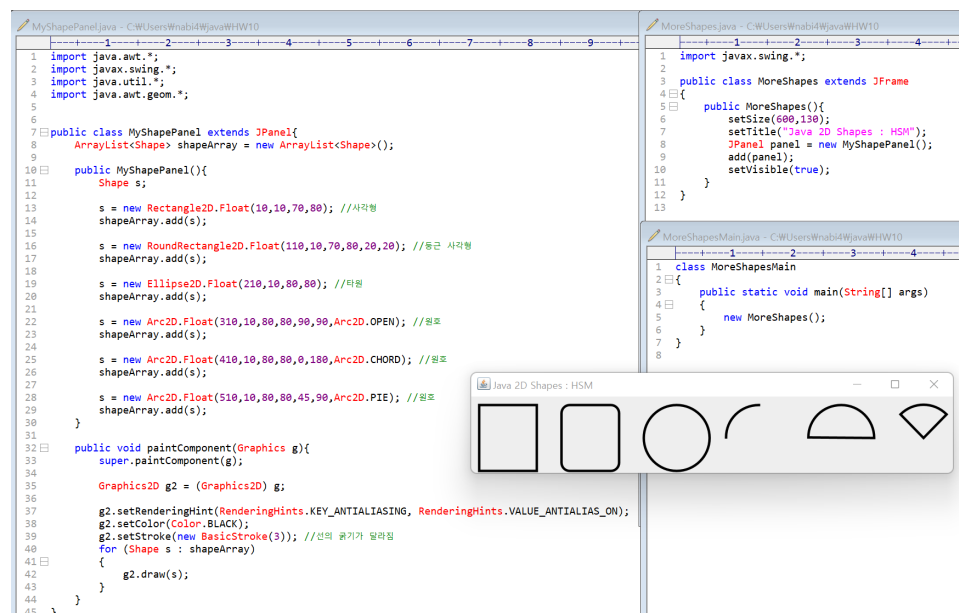
new Ellipse2D.Double(x, y, rectwidth, rectheight);

▼ Arc2D

new Arc2D.Float(float x, float y, float w, float h, float start, float extent, int
type)

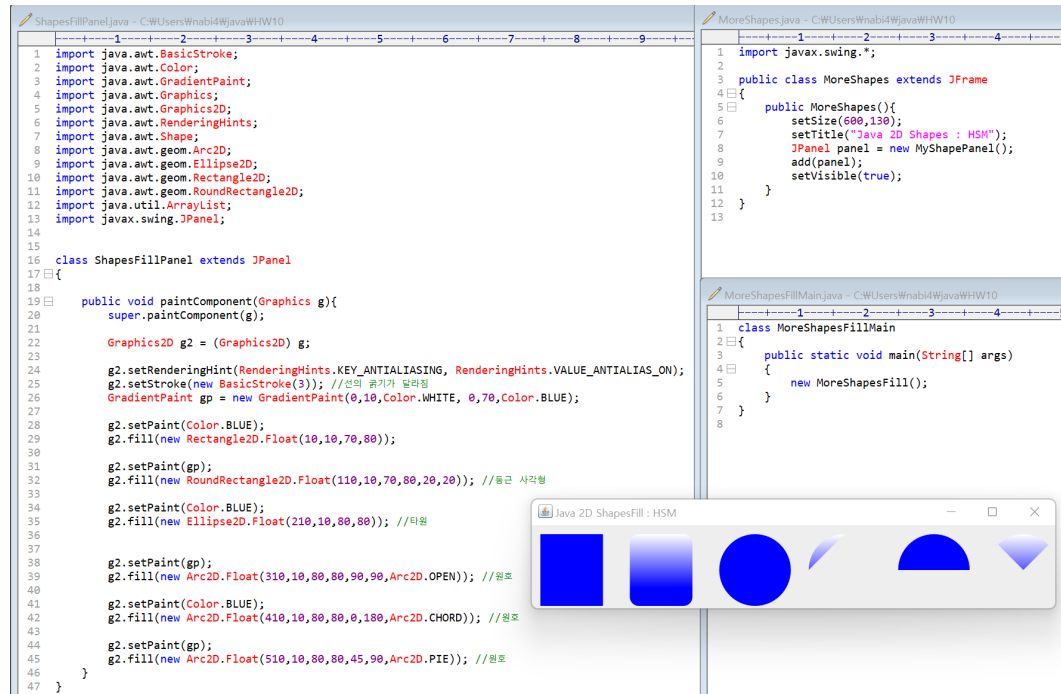
type : OPEN, CHORD, PIE

▼ 6.4.1Lab

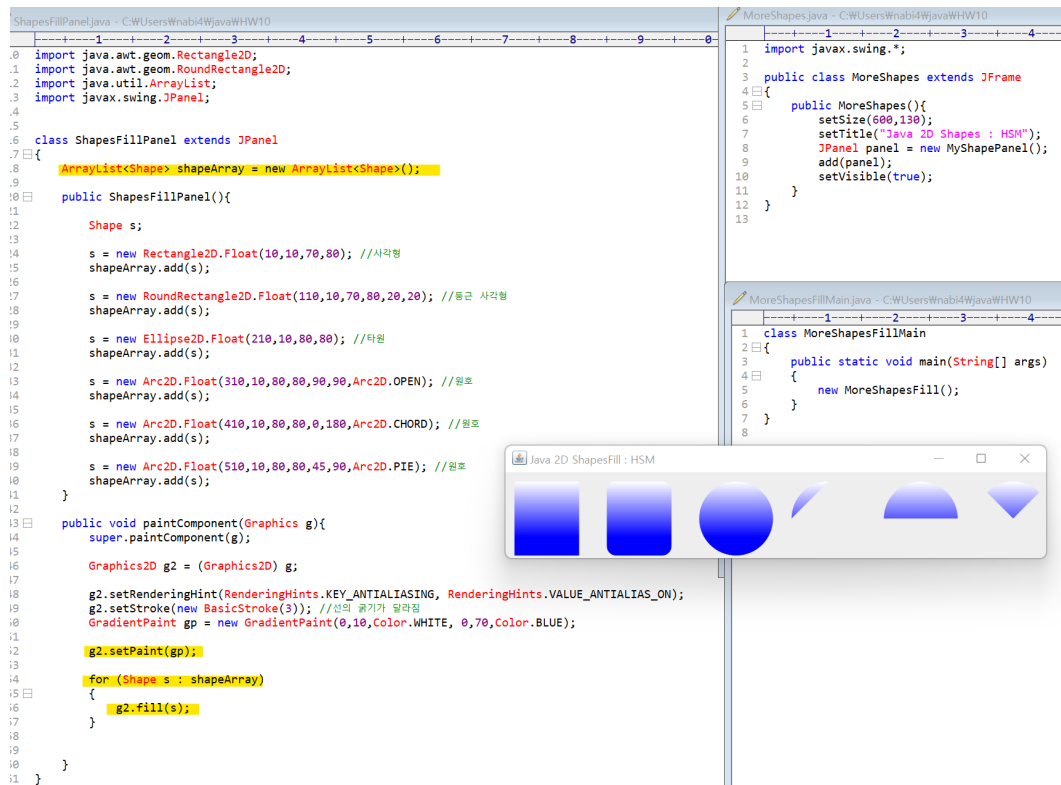


각 메서드에 맞게 원하는 그림을 그려주면 됨.
BasicStroke에 따라 선의 굵기가 달라짐

▼ 6.4.2 Shape Filling : Lab2(색깔 채우기)



위에서 그린 그림에 색을 채워봄(draw 대신 fill을 이용)



Lab1에서 사용한 ArrayList와 for문을 활용하여 그려봄(setPaint를 gp로 해두었기때문에 모두 다 그라데이션이 된 도형이 그려진 것을 볼 수 있음)

▼ 6.4.3 Lab3(Gradient Filling)

▼ Constructor

GradientPaint(float x1, float y1, Color color1, float x2, float y2, Color color2, Boolean cyclic)

▼ Using

```
GradientPaint gp = new GradientPaint(0, 0, Color.WHITE, 0, 100, Color.RED);  
setPaint(gp);
```

▼ Lab2에서 실시함

▼ 6.4.4 Lab4(Stroke)

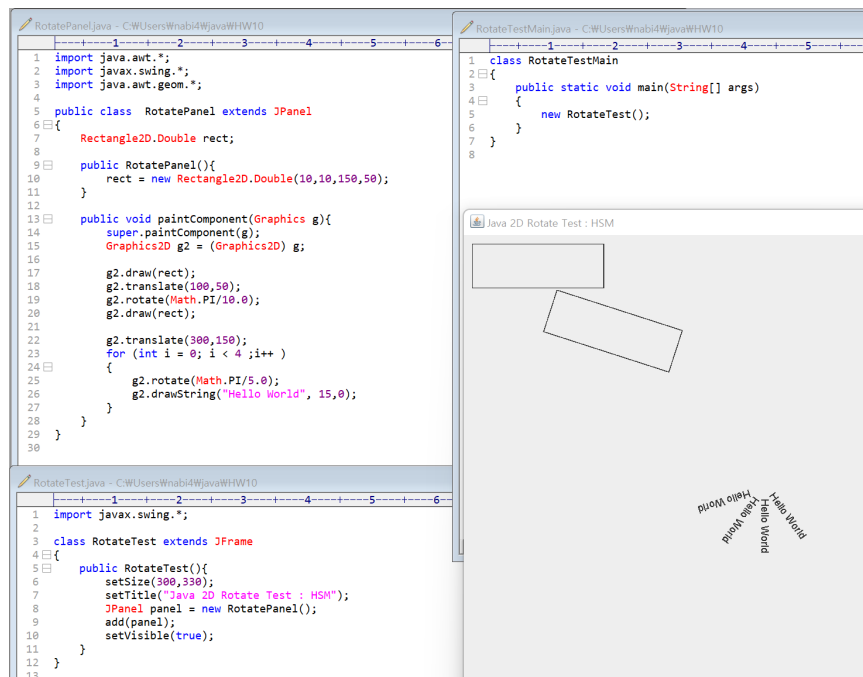
필요할 때 해보기

▼ 6.4.5 Rotate and Translate

translate(double x, double y) : 좌표값 옮기기

rotate(double theta) : 회전

▼ Lab5



▼ 6.4.6 Texture Rendering

도형에 이미지를 입히는 것

▼ Consturctor

TexturePaint(BufferedImage texture, Rectangle2D rect2d)

▼ Method

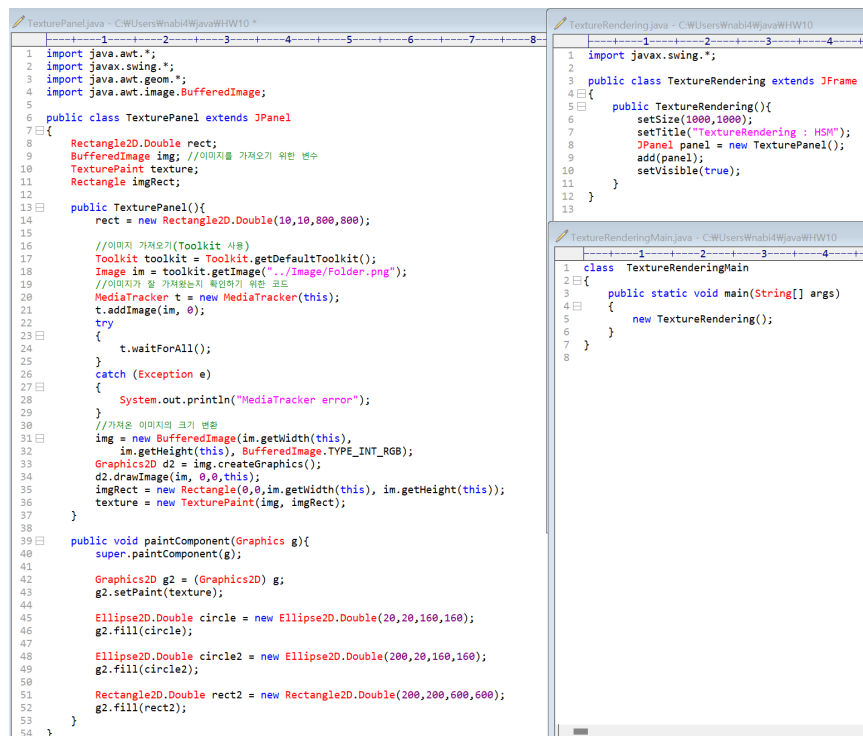
BufferedImage img = new BufferedImage(...);

Graphics2D d2 = img.createGraphics();

Rectangle imgRect = new Rectangle(...);

TexturePaint texture = new TexturePaint(img, imgRect);

▼ Lab



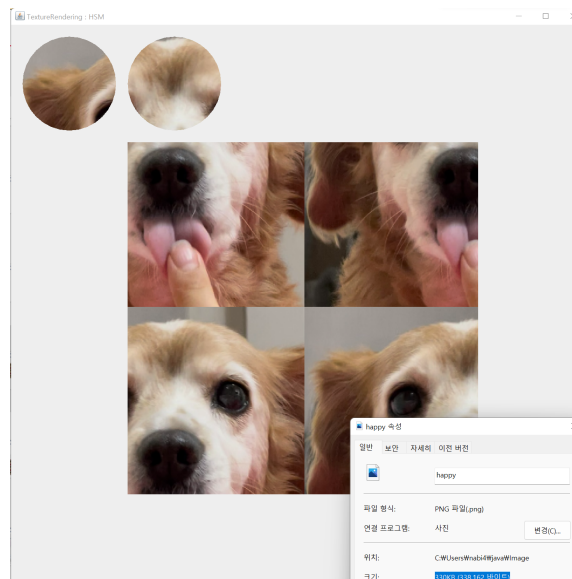
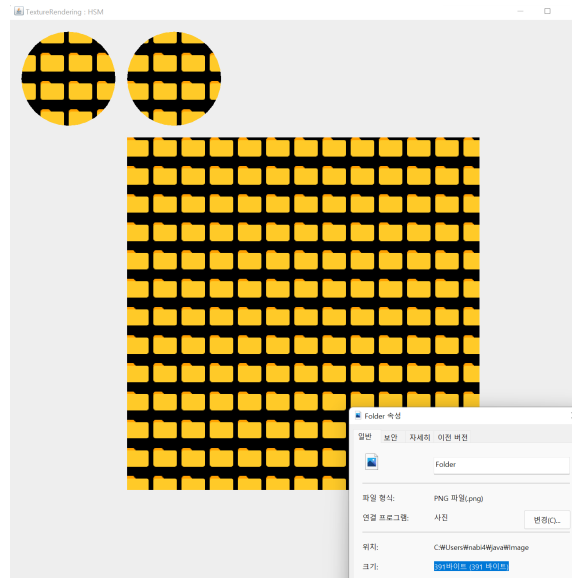
```
TexturePanel.java - C:\Users\Wnabi4\Java\HW10
1 import java.awt.*;
2 import javax.swing.*;
3 import java.awt.geom.*;
4 import java.awt.image.BufferedImage;
5
6 public class TexturePanel extends JPanel
7 {
8     Rectangle2D.Double rect;
9     BufferedImage img; //이미지를 가져오기 위한 변수
10    TexturePaint texture;
11    Rectangle imgRect;
12
13    public TexturePanel(){
14        rect = new Rectangle2D.Double(10,10,800,800);
15
16        //이미지 가져오기(Toolkit 사용)
17        Toolkit toolkit = Toolkit.getDefaultToolkit();
18        Image im = toolkit.getImage("../Image/Folder.png");
19        //이미지가 잘 가져왔는지 확인하기 위한 코드
20        MediaTracker t = new MediaTracker(this);
21        t.addImage(im, 0);
22        try
23        {
24            t.waitForAll();
25        }
26        catch (Exception e)
27        {
28            System.out.println("MediaTracker error");
29        }
30        //가져온 이미지의 크기 변환
31        img = new BufferedImage(im.getWidth(this),
32                                im.getHeight(this), BufferedImage.TYPE_INT_RGB);
33        Graphics2D d2 = img.createGraphics();
34        d2.drawImage(im, 0,0,this);
35        imgRect = new Rectangle(0,0,im.getWidth(this), im.getHeight(this));
36        texture = new TexturePaint(img, imgRect);
37    }
38
39    public void paintComponent(Graphics g){
40        super.paintComponent(g);
41
42        Graphics2D g2 = (Graphics2D) g;
43        g2.setPaint(texture);
44
45        Ellipse2D.Double circle = new Ellipse2D.Double(20,20,160,160);
46        g2.fill(circle);
47
48        Ellipse2D.Double circle2 = new Ellipse2D.Double(200,20,160,160);
49        g2.fill(circle2);
50
51        Rectangle2D.Double rect2 = new Rectangle2D.Double(200,200,600,600);
52        g2.fill(rect2);
53    }
54 }
```

```
TextureRenderingMain.java - C:\Users\Wnabi4\Java\HW10
1 import javax.swing.*;
2
3 public class TextureRendering extends JFrame
4 {
5     public TextureRendering(){
6         setSize(1000,1000);
7         setTitle("TextureRendering : HSM");
8         JPanel panel = new TexturePanel();
9         add(panel);
10        setVisible(true);
11    }
12 }
13
```

```
TextureRenderingMain.java - C:\Users\Wnabi4\Java\HW10
1 class TextureRenderingMain
2 {
3     public static void main(String[] args)
4     {
5         new TextureRendering();
6     }
7 }
8
```

그리고 싶은 도형을 그려준 후 불러온 이미지를 이용해 도형 위에 입힘

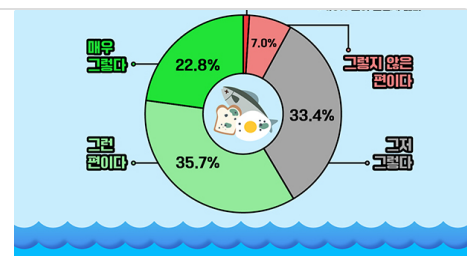
▼ 이미지 크기에 따라 사진이 많이 나올 수도 적게 나올 수도 있음



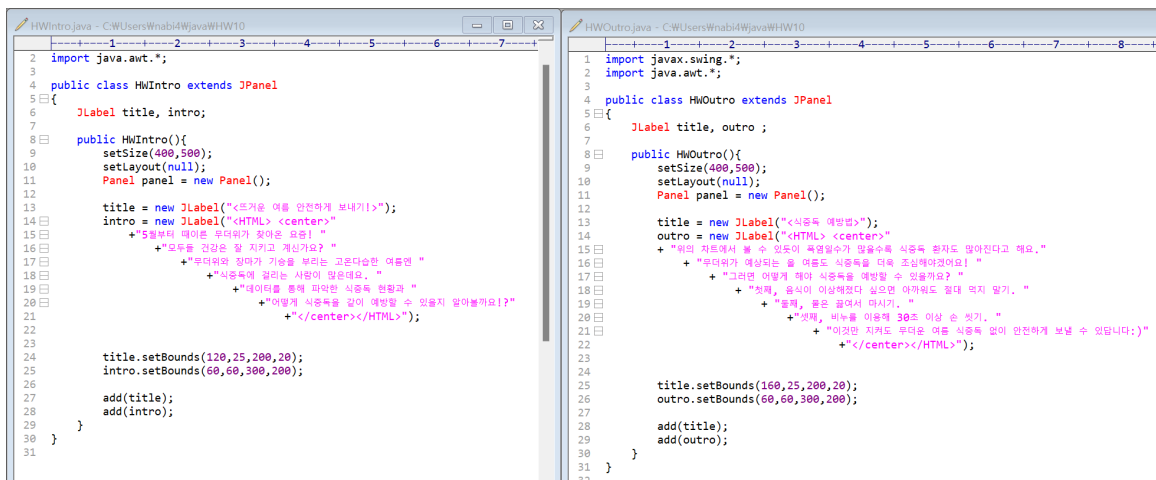
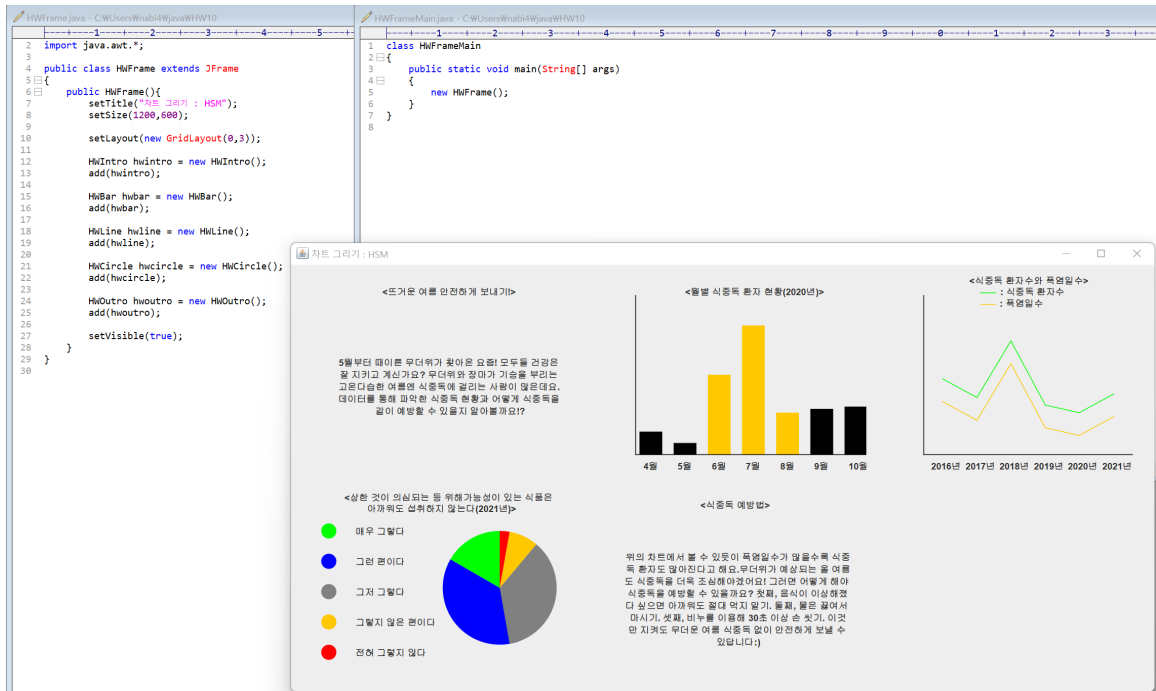
▼ HW(차트 만들기)

KOSIS 통계누리터

https://kosis.kr/edu/cardNews/detail.do?menuId=M_08_02&idx=42



다음 카드 뉴스를 오늘 배운 것들로 그려보고자 하였다



첫 페이지와 마지막 페이지


```

1 import javax.swing.*;
2 import java.awt.*;
3 import java.awt.Graphics;
4 import java.awt.Graphics2D;
5 import java.awt.RenderingHints;
6 import java.awt.geom.*;
7
8 public class HwBar extends JPanel
9 {
10     JLabel title, apr, may, jun, jul, aug, sep, oct;
11
12     public HwBar() {
13         setSize(400, 500);
14         setLayout(null);
15
16         title = new JLabel("<월별 식중독 환자 현황(2020년)>");
17
18         apr = new JLabel("4월");
19         may = new JLabel("5월");
20         jun = new JLabel("6월");
21         jul = new JLabel("7월");
22         aug = new JLabel("8월");
23         sep = new JLabel("9월");
24         oct = new JLabel("10월");
25
26         title.setBounds(140, 25, 200, 30);
27         apr.setBounds(85, 250, 30, 30);
28         may.setBounds(130, 250, 30, 30);
29         jun.setBounds(175, 250, 30, 30);
30         jul.setBounds(220, 250, 30, 30);
31         aug.setBounds(265, 250, 30, 30);
32         sep.setBounds(310, 250, 30, 30);
33         oct.setBounds(355, 250, 30, 30);
34
35         add(title);
36         add(apr);
37         add(may);
38         add(jun);
39         add(jul);
40         add(aug);
41         add(sep);
42         add(oct);
43
44     }
45
46     public void paintComponent(Graphics g) {
47         super.paintComponent(g);
48
49         Graphics2D g2 = (Graphics2D) g;
50         g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING, R
51
44
45
46
47 public void paintComponent(Graphics g){
48     super.paintComponent(g);
49
50     Graphics2D g2 = (Graphics2D) g;
51     g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING, RenderingHints.VALUE_ANTIALIAS_ON);
52
53     g2.setPaint(Color.black);
54     g2.fill(new Rectangle2D.Float(80,220,30,30));
55     g2.fill(new Rectangle2D.Float(125,235,30,15));
56
57     g2.setPaint(Color.orange);
58     g2.fill(new Rectangle2D.Float(170,145,30,105));
59     g2.fill(new Rectangle2D.Float(215,80,30,170));
60     g2.fill(new Rectangle2D.Float(260,195,30,55));
61
62     g2.setPaint(Color.black);
63     g2.fill(new Rectangle2D.Float(305,190,30,60));
64     g2.fill(new Rectangle2D.Float(350,187,30,63));
65
66     g2.drawLine(75, 250, 400,250);
67     g2.drawLine(75,250,75,40);
68
69
70
71
72 }
73

```

두번째 페이지(막대 그래프)

```

1 import javax.swing.*;
2 import java.awt.*;
3 import java.awt.Graphics;
4 import java.awt.Graphics2D;
5 import java.awt.RenderingHints;
6 import java.awt.geom.*;
7
8 public class HwLine extends JPanel
9 {
10     JLabel title, num16, num17, num18, num19, num20, num21;
11     JLabel peoplenum, sunny;
12
13     public HwLine() {
14         setSize(400, 500);
15         setLayout(null);
16
17         title = new JLabel("<식중독 환자수와 폭염일수>");
18
19         num16 = new JLabel("2016년");
20         num17 = new JLabel("2017년");
21         num18 = new JLabel("2018년");
22         num19 = new JLabel("2019년");
23         num20 = new JLabel("2020년");
24         num21 = new JLabel("2021년");
25
26         peoplenum = new JLabel("식중독 환자수");
27         sunny = new JLabel("폭염일수");
28
29         title.setBounds(135,10,250,20);
30         peoplenum.setBounds(172, 25, 250,20);
31         sunny.setBounds(175, 40, 250,20);
32         num16.setBounds(85, 250,50,30);
33         num17.setBounds(130, 250,50,30);
34         num18.setBounds(175, 250,50,30);
35         num19.setBounds(220, 250,50,30);
36         num20.setBounds(265, 250,50,30);
37         num21.setBounds(310, 250,50,30);
38
39         add(title);
40         add(peoplenum);
41         add(sunny);
42         add(num16);
43         add(num17);
44         add(num18);
45         add(num19);
46         add(num20);
47         add(num21);
48
49     }
50
51     public void paintComponent(Graphics g) {
52         super.paintComponent(g);
53
54         Graphics2D g2 = (Graphics2D) g;
55         g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING, RenderingHints.VALUE_ANTIALIAS_ON);
56
57         g2.setPaint(Color.green);
58         g2.drawLine(150,37, 170, 37);
59         g2.drawLine(100,150,145,175);
60         g2.drawLine(145,175,190,100);
61         g2.drawLine(190,100,235,185);
62         g2.drawLine(235,185,280,195);
63         g2.drawLine(280,195,325,170);
64
65         g2.setPaint(Color.orange);
66         g2.drawLine(150,52, 170, 52);
67         g2.drawLine(100,180,145,205);
68         g2.drawLine(145,205,190,130);
69         g2.drawLine(190,130,235,215);
70         g2.drawLine(235,215,280,225);
71         g2.drawLine(280,225,325,200);
72
73         g2.setPaint(Color.black);
74         g2.drawLine(75,250,350,250);
75         g2.drawLine(75,250,75,40);
76
77
78
79
80
81
82
83
84
85
86

```

세번째 페이지(꺾은선 그래프)_폭염일수와 식중독 환자수 2개의 꺾은선 그래프가 그려짐

```

1 import javax.swing.*;
2 import java.awt.*;
3 import java.awt.Graphics;
4 import java.awt.Graphics2D;
5 import java.awt.RenderingHints;
6 import java.awt.geom.*;
7
8 public class HWCircle extends JPanel
9 {
10     JLabel title, title2, verygood, excellent, good, fair, notsatisfied;
11
12     public HWCircle(){
13         setSize(400,500);
14         setLayout(null);
15
16         title = new JLabel("<상한 것의 의심되는 등 위험가능성이 있는 식품>");
17         title2 = new JLabel("<여객기도 급속히 감소(2021년)>");
18
19         verygood = new JLabel("<매우 좋음>");
20         excellent = new JLabel("<고전 편이다>");
21         good = new JLabel("<그럭저럭>");
22         fair = new JLabel("<그럭저럭 않은 편이다>");
23         notsatisfied = new JLabel("<전혀 그렇지 않다>");
24
25         title.setBounds(70,15,300,20);
26         title2.setBounds(100,30,300,20);
27         verygood.setBounds(85,60,120,20);
28         excellent.setBounds(85,100,120,20);
29         good.setBounds(85,140,120,20);
30         fair.setBounds(85,180,120,20);
31         notsatisfied.setBounds(85, 220, 120,20);
32
33         add(title);
34         add(title2);
35         add(verygood);
36         add(excellent);
37         add(good);
38         add(fair);
39         add(notsatisfied);
40     }
41
42     public void paintComponent(Graphics g){
43         super.paintComponent(g);
44         Graphics2D g2 = (Graphics2D) g;
45         g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING, RenderingHints.VALUE_ANTIALIAS_ON);
46
47         g2.setPaint(Color.green);
48         g2.fill(new Ellipse2D.Float(40,60,20,20));
49
50         g2.setPaint(Color.blue);
51         g2.fill(new Ellipse2D.Float(40,100,20,20));
52
53         g2.setPaint(Color.gray);
54         g2.fill(new Ellipse2D.Float(40,140,20,20));
55
56         g2.setPaint(Color.orange);
57         g2.fill(new Ellipse2D.Float(40,180,20,20));
58
59         g2.setPaint(Color.red);
60         g2.fill(new Ellipse2D.Float(40,220,20,20));
61
62
63         g2.setPaint(Color.green);
64         g2.fill(new Arc2D.Float(200,70,150,150,90,Arc2D.PIE));
65
66         g2.setPaint(Color.blue);
67         g2.fill(new Arc2D.Float(200,70,150,150,140,Arc2D.PIE));
68
69         g2.setPaint(Color.gray);
70         g2.fill(new Arc2D.Float(200,70,150,150,50,-130,Arc2D.PIE));
71
72         g2.setPaint(Color.orange);
73         g2.fill(new Arc2D.Float(200,70,150,150,80,-30,Arc2D.PIE));
74
75         g2.setPaint(Color.red);
76         g2.fill(new Arc2D.Float(200,70,150,150,90,-10,Arc2D.PIE));
77
78     }
79
80 }
81

```

네번째 페이지(파이차트)

▼ 7.0 Event

이벤트 : 자바 언어를 통해 UI로 상호작용을 하기 위함

GUI를 통한 / 시스템으로부터(timer or IO) 발생하는 행동

▼ Event-Driven Handling

: 마우스 버튼 클릭, 더블 클릭, 마우스 이동 등의 모든 사용자 액션에 대해 응답하는 형태로 작동하도록 프로그래밍하는 것

ECA Rule(Event : click , Condition , Action)

이전까진 프로그램 상에서 이벤트를 발생시켰다면 지금은 UI를 통해 이벤트를 발생시키는 것을 배울 것임

어떠한 event가 진행되었을 때 함수가 호출되어 진행을 시켜줌

개발자는 함수를 구현하면 됨

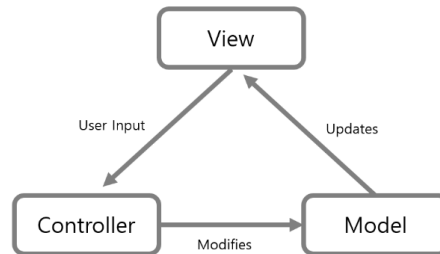
▼ MVC Model(정리하기)

Model(레이아웃 구조, static), View(dynamic)-Event Handling-Control

어떤 모델이 control을 통해서 “view”가 변화함

event driven 과 연관되는 모델임

▼ 추가



- Model : 화면 안의 네모박스에 글자가 표현된다면, 네모박스의 화면 위치 정보, 네모박스의 크기 정보, 글자 내용, 글자의 위치, 글자의 포맷 정보 등을 말함 / 재사용 가능 / 다른 인터페이스에서도 변하지 않아야 함
- View : 사용자 인터페이스 요소. 데이터 및 객체의 입력과 보여주는 출력을 담당 / 이벤트 핸들러 작동 시 변화가 생기는 부분 / 재사용 가능
- Control : 사용자의 행동에 의해 프로그램이 진행될 수 있도록 이벤트를 처리하는 부분

▼ Event Listener

이벤트가 발생하는 것을 인지하기 위한 장치

listener에 대한 action을 구현해야함

▼ Event Object

Method 이름이 정해져있음

getSource(), getId(), getActionCommand()

▼ Step

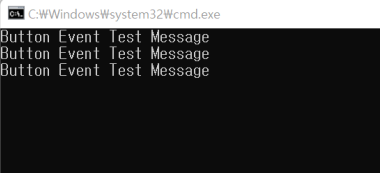
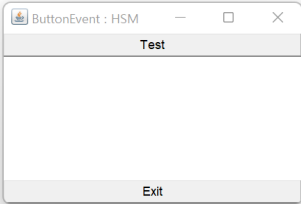
1. 객체 만들기(GUI 설계)
2. Listener 등록
3. method 구현(implement)

ActionListener : 인터페이스를 구현해줘야 함

▼ Lab

```
ButtonEvent.java - C:\Users\Wnabi4W\java\HW10
1 import java.awt.*;
2 import java.awt.event.*;
3
4 public class ButtonEvent implements ActionListener
5 {
6     Frame f;
7     Button test, exit;
8
9     public ButtonEvent(String msg){
10         f = new Frame(msg);
11         test = new Button("Test");
12         test.addActionListener(this);
13
14         exit = new Button("Exit");
15         exit.addActionListener(this);
16
17         f.add("North", test);
18         f.add("South", exit);
19
20         f.setSize(300,200);
21         f.setVisible(true);
22     }
23
24     public void actionPerformed(ActionEvent e){
25         if (e.getSource() == test)
26         {
27             System.out.println("Button Event Test Message");
28         }else if (e.getSource() == exit)
29         {
30             f.setVisible(false);
31             f.dispose();
32             System.exit(0);
33         }
34     }
35 }
36
```

```
ButtonEventMain.java - C:\Users\Wnabi4W\java\HW10
1 class ButtonEventMain
2 {
3     public static void main(String[] args)
4     {
5         new ButtonEvent("ButtonEvent : HSM");
6     }
7 }
8
```



Test 버튼을 누르면 cmd 창에 글씨가, Exit 버튼을 누르면 창이 닫기며 프로그램이 종료되는 코드 구현