

자바 그래픽

AWT – 기본적인 도형들을 쉽게 그릴 수 있다.

Java 2D – 커브를 그릴 수 있으며, 2차원 도형을 회전, 그라디언트 채우기 등 을 할 수 있다.

간단한 도형을 그리는 순서

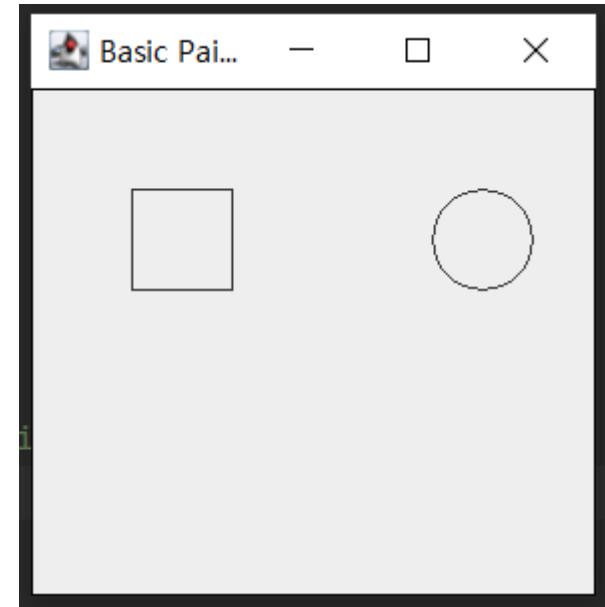
Jframe 생성-> Jpanel 생성 -> 도형 그리기 (MyPanel) -> 좌표값 계산 -> 도형 그리기 함수 호출

그래픽 좌표값은 x축의 값이 커질 수록 왼쪽에서 오른쪽, y축의 값이 커질 수록 위에서 아래로 증가한다.
X,Y축의 값을 시작으로 heigh(높이), width(너비)를 지정해준다.

6.1 Draw Basic Shape

```
BasicPaint.java x
1  import javax.swing.*;
2
3  1 usage
4  public class BasicPaint {
5      1 usage
6      public BasicPaint(String msg){
7          JFrame f = new JFrame(msg);
8          MyPanel myPanel1 = new MyPanel();//그리기 위한 Panel 생성
9          f.add(myPanel1); //mypanel붙이기
10         f.setSize( width: 300, height: 300);
11         f.setVisible(true);
12     }
13     public static void main(String [] args) { new BasicPaint( msg: "Basic Paint Test1"); }
14 }
15
```

결과화면



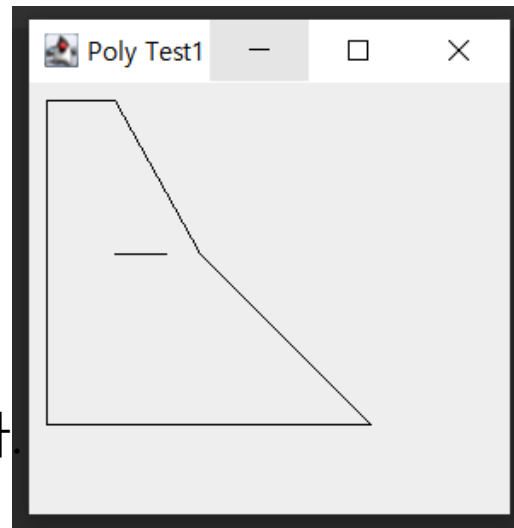
사각형 – drawrect
원 - drawoval

6.2 Line draw

```
BasicPaint.java x MyPolylinePanel.java x MyPolylineMain.java x
1 import javax.swing.*;
2 import java.awt.*;
3
4 2 usages
5 public class MyPolylinePanel extends JPanel {
6     1 usage
7     public MyPolylinePanel(){
8
9     }
10    public void paintComponent(Graphics g){
11        super.paintComponent(g);
12
13        g.setColor(Color.black);
14
15        int[] xp = new int[]{10,50,100,200,10,10};
16        int[] yp = new int[]{10,10,100,200,200,10};
17
18        g.drawPolyline(xp,yp, nPoints: 6);
19
20        g.drawLine( x1: 50, y1: 100, x2: 80, y2: 100);
21    }
```

```
BasicPaint.java x MyPolylinePanel.java x MyPolylineMain.java x
1 import javax.swing.*;
2
3 1 usage
4 public class MyPolylineMain {
5     1 usage
6     public MyPolylineMain(String msg){
7         JFrame f = new JFrame(msg);
8         MyPolylinePanel mypolyPanel1 = new MyPolylinePanel();//그리기 위한 Panel 생성
9         f.add(mypolyPanel1);
10        f.setSize( width: 300, height: 300);
11        f.setVisible(true);
12    }
13    public static void main(String [] args) { new MyPolylineMain( msg: "Poly Test1"); }
14 }
15 }
```

결과화면



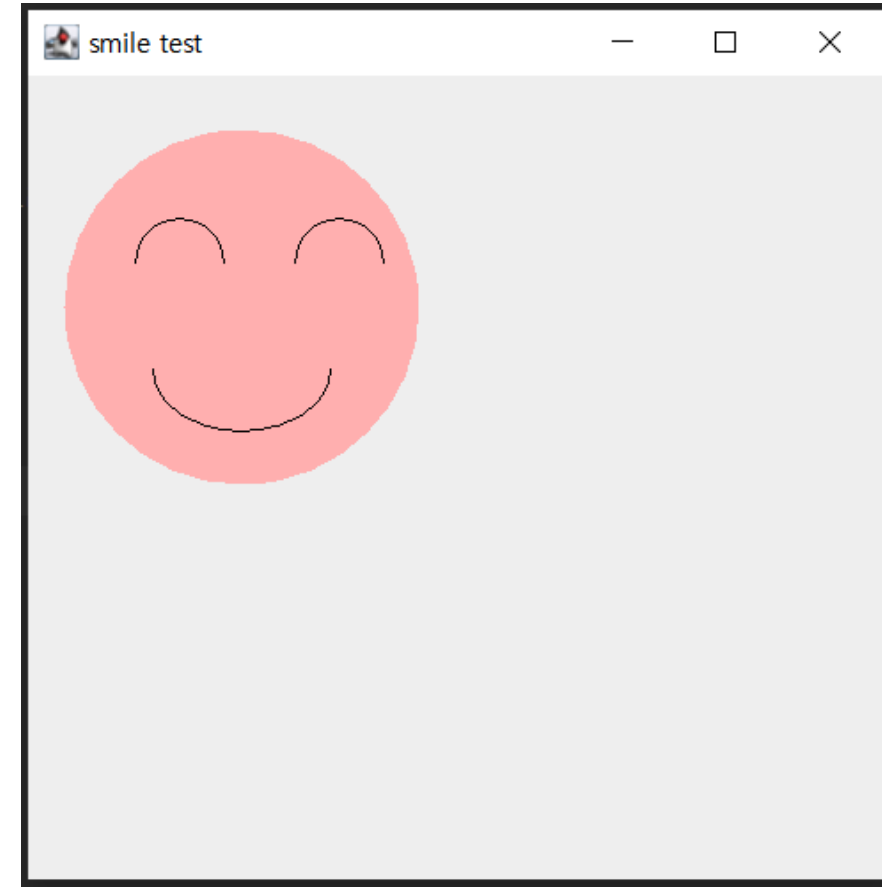
직선 – drawline

Polyline을 이용하여 점선을 가지고 직선으로 연결된다.

6.2 smile Face

```
BasicPaint.java x MyPolylinePanel.java x MyPolylineMain.java x SmileFacePanel.java x SmileFace.java x
1 import java.awt.*;
2 import javax.swing.*;
3 public class SmileFacePanel extends JPanel{
4     public SmileFacePanel(){
5         //초기화
6     }
7     public void paintComponent(Graphics g){
8         g.setColor(Color.pink);
9         g.fillOval( x: 20, y: 30, width: 200, height: 200);
10        g.setColor(Color.black);
11        g.drawArc( x: 60, y: 80, width: 50, height: 50, startAngle: 180, arcAngle: -180); //왼쪽 눈
12        g.drawArc( x: 150, y: 80, width: 50, height: 50, startAngle: 180, arcAngle: -180); //오른쪽 눈
13        g.drawArc( x: 70, y: 130, width: 100, height: 70, startAngle: 180, arcAngle: 180); //입
14
15 BasicPaint.java x MyPolylinePanel.java x MyPolylineMain.java x SmileFacePanel.java x SmileFace.java x
1 import javax.swing.*;
2 import java.awt.*;
3
4 public class SmileFace {
5     public SmileFace(String msg){
6         JFrame f = new JFrame(msg);
7         SmileFacePanel smileface = new SmileFacePanel(); //그리기 위한 Panel 생성
8         f.add(smileface);
9         f.setSize( width: 500, height: 500);
10        f.setVisible(true);
11    }
12    public static void main(String [] args){
13        new SmileFace( msg: "smile test");
14    }
15 }
16
```

결과화면



원 + 호를 사용하여 웃는 얼굴을 그렸다.
Drawarc를 사용해 arcangle의 각도 만큼의 호를 그린다.

H.W SnowMan

```
BasicPaint.java x MyPolylinePanel.java x MyPolylineMain.java x SmileFacePanel.java x SmileFace.java x SnowManPan
1 usage
5 public SnowManPanel(){
6     //초기화
7 }
8 @
9 public void paintComponent(Graphics g) {
10     final int MID = 150;
11     final int TOP = 50;
12     setBackground (Color.cyan);
13     g.setColor (Color.blue);
14     g.fillRect ( x: 0, y: 175, width: 300, height: 50); // 바닥
15
16     g.setColor (Color.yellow);
17     g.fillOval ( x: -40, y: -40, width: 80, height: 80); // 해
18
19     g.setColor (Color.white);
20     g.fillOval ( x: MID-20, TOP, width: 40, height: 40); // 모자
21     g.fillOval ( x: MID-35, y: TOP+35, width: 70, height: 50); // upper torso
22     g.fillOval ( x: MID-50, y: TOP+80, width: 100, height: 60); // lower torso
23
24     g.setColor (Color.black);
25     g.fillOval ( x: MID-10, y: TOP+10, width: 5, height: 5); // 왼쪽 눈
26     g.fillOval ( x: MID+5, y: TOP+10, width: 5, height: 5); // 오른쪽 눈
27
28     g.drawArc ( x: MID-10, y: TOP+20, width: 20, height: 10, startAngle: 190, arcAngle: 160); // 입
29
30     g.drawLine ( x1: MID-25, y1: TOP+60, x2: MID-50, y2: TOP+40); // 왼팔
31     g.drawLine ( x1: MID+25, y1: TOP+60, x2: MID+55, y2: TOP+40); // 오른팔
32
33     g.drawLine ( x1: MID-20, y1: TOP+5, x2: MID+20, y2: TOP+5); // 모자
34     g.fillRect ( x: MID-15, y: TOP-20, width: 30, height: 25);
```

```
BasicPaint.java x MyPolylinePanel.java x MyPolylineMain.java x SmileFacePanel.java x SmileFace.java x
1 import javax.swing.*;
2
3 1 usage
4 public class SnowManMain {
5     1 usage
6     public SnowManMain(String msg){
7         JFrame f = new JFrame(msg);
8         SnowManPanel snowman = new SnowManPanel(); //그리기 위한 Panel 생성
9         f.add(snowman);
10        f.setSize( width: 300, height: 300);
11        f.setVisible(true);
12    }
13
14    public static void main(String [] args) { new SnowManMain( msg: "hjsnowman"); }
15 }
16
```

결과화면



6.2 Drawing Tree

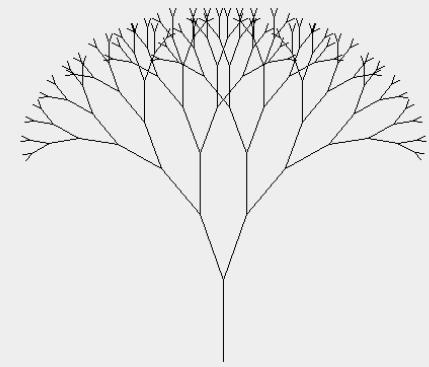
```
BasicPaint.java x MyPolylinePanel.java x TreePanel.java x TreeMain.java x MyPolylineMain.java x
1 import javax.swing.*;
2 import java.awt.*;
3
4 1 usage
5 public class TreePanel extends JPanel {
6     1 usage
7     public TreePanel(){
8     }
9     public void paintComponent(Graphics g){
10         g.setColor(Color.BLACK);
11         drawtree(g, x1: 400, y1: 600, angle: -90, depth: 8);
12     }
13
14 3 usages
15 public void drawtree(Graphics g, int x1,int y1, double angle, int depth){
16     if (depth ==0) return;
17     int x2 = x1 + (int)(Math.cos(Math.toRadians(angle))*depth*10.0);
18     int y2 = y1 + (int)(Math.sin(Math.toRadians(angle))*depth*10.0);
19     g.drawLine(x1,y1,x2,y2);
20     drawtree(g,x2,y2, angle: angle-20, depth: depth-1);
21     drawtree(g,x2,y2, angle: angle+20, depth: depth-1);
22 }
23 }
```

```
BasicPaint.java x MyPolylinePanel.java x TreePanel.java x TreeMain.java x MyPolylineMain.java x
1 import javax.swing.*;
2 1 usage
3 public class TreeMain extends JFrame{
4     1 usage
5     public TreeMain (String msg){
6         setTitle(msg);
7         add(new TreePanel());
8         setSize( width: 600, height: 600);
9         setVisible(true);
10    }
11    public static void main(String [] args) { new TreeMain( msg: "treeTest"); }
12 }
13 }
```

treeTest

결과화면

나무 줄기를 그린 후 줄기의 끝점에서 특정한 각도로 2가지 가지를 그린다.
동일한 과정을 반복하지만, 깊이를 지정하여 특정 깊이에 도달하면 멈춘다.



6.3 Color Lab1

```
MyPolylinePanel.java x TreePanel.java x TreeMain.java x ColorPanel.java x ColorMain.java
1  import javax.swing.*;
2  import java.awt.*;
3  import java.awt.event.*;
4  1 usage
5  public class ColorPanel extends JPanel implements ActionListener {
6      3 usages
7      JButton button;
8      2 usages
9      Color color = new Color( r: 0, g: 0, b: 0);
10
11      1 usage
12      public ColorPanel(){
13          setLayout(new BorderLayout());
14          //버튼 생성
15          button = new JButton( text: "Color Change");
16          button.addActionListener( l: this); //이벤트 발생
17          add(button, BorderLayout.SOUTH);
18      }
19
20      public void paintComponent(Graphics g){
21          super.paintComponent(g);
22          g.setColor(color);
23          g.fillRect( x: 50, y: 50, width: 300, height: 300);
24      }
25
26      public void actionPerformed(ActionEvent e){
27          color = new Color((int)(Math.random()*255.0),
28                          (int)(Math.random()*255.0),
29                          (int)(Math.random()*255.0));
30          repaint(); //event 에 대한 paint 할수 다시 호출함
31      }
32  }
```

```
MyPolylinePanel.java x TreePanel.java x TreeMain.java x ColorPanel.java x ColorMain.java x MyPoly
1  import javax.swing.*;
2  import java.awt.event.*;
3
4  1 usage
5  public class ColorMain extends JFrame {
6      1 usage
7      public ColorMain(String msg){
8          super(msg);
9          JPanel panel = new ColorPanel();
10         add(panel);
11         setSize( width: 400, height: 400);
12         setVisible(true);
13     }
14
15     public static void main(String [] args) { new ColorMain( msg: "JHJ Color Test"); }
16
17 }
18
```

결과화면

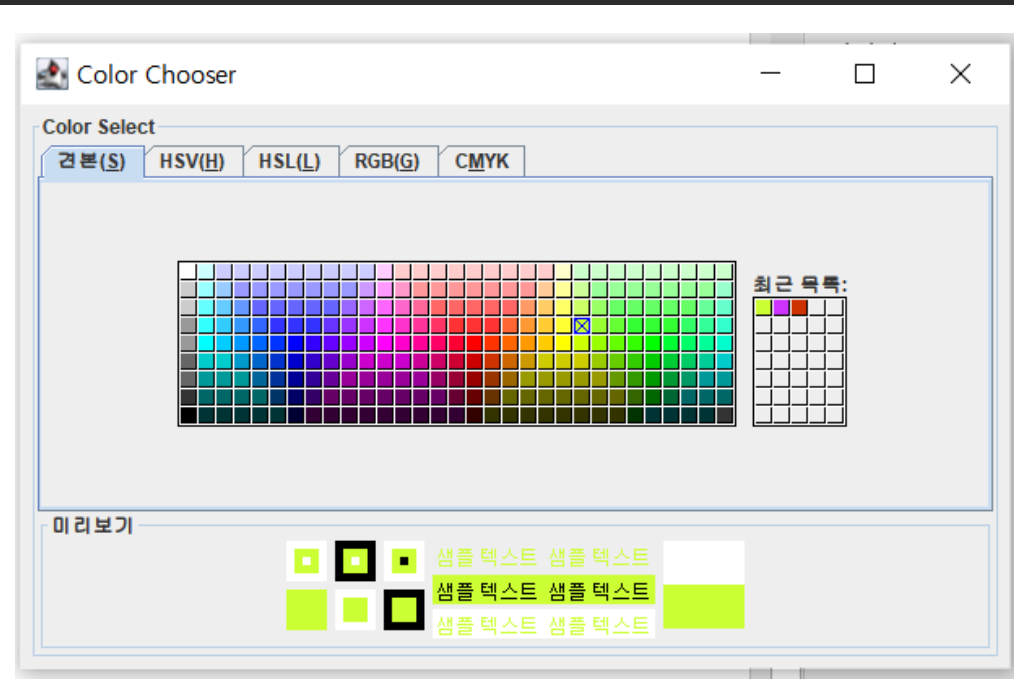


6.3 Color Lab2 Color setting

```
MyPolylinePanel.java x TreePanel.java x TreeMain.java x ColorPanel.java x ColorMain.java x
4 import java.awt.*;
5 import java.awt.event.*;
6
7 2 usages
8 public class ColorChooser extends JFrame implements ChangeListener {
9     public JColorChooser color;
10
11 1 usage
12 public ColorChooser(){
13     setTitle("Color Chooser");
14     setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
15
16     color = new JColorChooser(); //생성자 호출
17     color.getSelectionModel().addChangeListener(this); //리스너 등록
18     color.setBorder(BorderFactory.createTitledBorder("Color Select"));
19     JPanel panel = new JPanel();
20     panel.add(color);
21     add(panel);
22     pack();
23     this.setVisible(true);
24 }
25
26 public void stateChanged(ChangeEvent e) { Color newcolor = color.getColor(); }
27 }
28
```

```
MyPolylinePanel.java x TreePanel.java x TreeMain.java x ColorPanel.java x ColorMain.java x Co
1 import javax.swing.*;
2
3 1 usage
4 public class ColorChooserMain extends JFrame {
5     1 usage
6     public ColorChooserMain(){
7         super();
8         ColorChooser colorChooser = new ColorChooser(); //내가 만든 colorchooser 생성
9         add(colorChooser);
10        setSize( width: 300, height: 300);
11        setVisible(true);
12    }
13
14    public static void main (String [] args){
15        new ColorChooserMain();
16    }
17 }
18
```

결과화면



6.4 Java 2D Lab1

```
linePanel.java x TreePanel.java x TreeMain.java x ColorPanel.java x ColorMain.java x ColorChooser.java x ColorChoose
4 import java.util.*;
5 public class ShapePanel extends JPanel {
6     ArrayList<Shape> shapeArrayList = new ArrayList<>();
7
8     public ShapePanel(){
9         Shape s;
10
11         s = new Rectangle2D.Float( x: 10, y: 10, w: 70, h: 80); //사각형
12         shapeArrayList.add(s);
13         s = new RoundRectangle2D.Float( x: 110, y: 10, w: 70, h: 80, arcw: 20, arch: 20); // 둥근 사각형
14         shapeArrayList.add(s);
15         s = new Ellipse2D.Float( x: 210, y: 10, w: 80, h: 80); // 타원
16         shapeArrayList.add(s);
17         s = new Arc2D.Float( x: 310, y: 10, w: 80, h: 80, start: 90, extent: 90, Arc2D.OPEN); // 원호: Arc2D.OPEN
18         shapeArrayList.add(s);
19         s = new Arc2D.Float( x: 410, y: 10, w: 80, h: 80, start: 0, extent: 180, Arc2D.CHORD); // 원호: Arc2D.CHORD
20         shapeArrayList.add(s);
21         s = new Arc2D.Float( x: 510, y: 10, w: 80, h: 80, start: 45, extent: 90, Arc2D.PIE); // 원호: Arc2D.PIE
22         shapeArrayList.add(s);
23     }
24     public void paintComponent (Graphics g){
25         super.paintComponent(g);
26         Graphics2D g2 = (Graphics2D) g;
27         g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING, RenderingHints.VALUE_ANTIALIAS_ON);
28         g2.setColor(Color.BLACK);
29         g2.setStroke(new BasicStroke( width: 5)); //선두께
30         for (Shape s : shapeArrayList) g2.draw(s);
31     }
32 }
```

```
linePanel.java x TreePanel.java x TreeMain.java x ColorPanel.java x
1 import javax.swing.*;
2
3 public class ShapeMain extends JFrame {
4     public ShapeMain(){
5         setTitle("JHJ 2D Shapes");
6         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
7         JPanel panel = new ShapePanel();
8         add(panel);
9         setSize( width: 600, height: 150);
10        setVisible(true);
11    }
12    public static void main(String [] args){
13        new ShapeMain();
14    }
15 }
```

결과화면



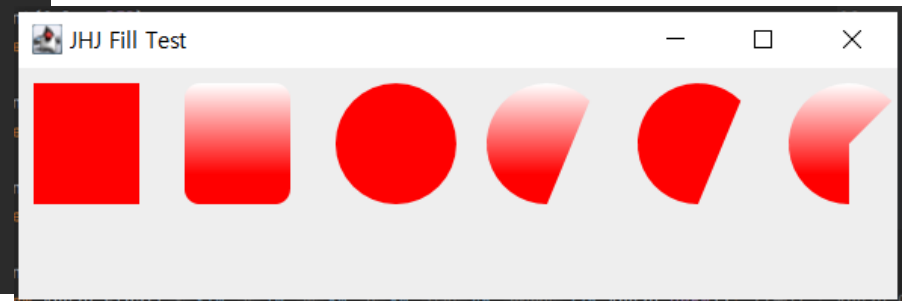
6.4 Java 2D Lab2

```

1      import javax.swing.*;
2
3      1 usage
4      public class ShapeFillMain extends JFrame {
5          1 usage
6          public ShapeFillMain(){
7              setTitle("JHJ Fill Test");
8              setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
9              JPanel panel = new ShapeFillPanel();
10             add(panel);
11             setSize( width: 600, height: 200);
12             setVisible(true);
13         }
14         public static void main(String [] args){
15             new ShapeFillMain();
16         }
17     }

```

결과화면



HW 나만의 차트 그리기

MAIN

버튼

```
1 import javax.swing.*;
2 import java.awt.*;
3
4 1 usage
5 public class Main extends JFrame {
6     1 usage
7     public Main() {
8         setTitle("JHJ 선형 Chart");
9         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
10        JPanel panel = new JPanel();
11        panel.setLayout(new GridLayout( rows: 2, cols: 3)); //가로 2줄, 세로 3줄로 layout을 만듦
12        JPanel panel1 = new lineChartPanel();
13        panel.add(panel1);
14        JPanel panel2 = new pchartPanel();
15        panel.add(panel2);
16        JPanel panel3 = new circlecartPanel();
17        panel.add(panel3);
18        JPanel panel4 = new MyChartPanel();
19        panel.add(panel4);
20        JPanel panel5 = new ButtonChart();
21        panel.add(panel5);
22        add(panel);
23        setSize( width: 1500, height: 1000);
24        setVisible(true);
25    }
26
27    public static void main(String[] args) {
28        new Main();
29    }
30 }
```

```
1 import javax.swing.*;
2 import java.awt.*;
3
4 1 usage
5 public class ButtonChart extends JPanel {
6     2 usages
7     JButton button1,button2;
8     4 usages
9     JPanel jp = new JPanel();
10
11    1 usage
12    public ButtonChart(){
13        //버튼 생성
14        setLayout(new BorderLayout());
15        button1 = new JButton( text: "Open. txt file");
16        button2 = new JButton( text: "Apply");
17
18        jp.add(button1);
19        jp.add(button2);
20        jp.setLayout(new FlowLayout());
21
22        add(jp,BorderLayout.CENTER);
23    }
24 }
```

선 Chart

```
1 import javax.swing.*;
2 import java.awt.*;
3 import java.awt.event.*;
4 import java.awt.geom.*;
5 import java.util.*;
6
7 2 usages
8 public class lineChartPanel extends JPanel implements ActionListener {
9     ArrayList<Shape> shapeArrayList = new ArrayList<>();
10
11     3 usages
12     JButton button;
13     2 usages
14     Color color1 = new Color( r: 0, g: 0, b: 0);
15     2 usages
16     Color color2 = new Color( r: 0, g: 0, b: 0);
17     2 usages
18     public lineChartPanel(){
19         setLayout(new BorderLayout());
20         //버튼 생성
21         button = new JButton( text: "Color Change");
22         button.addActionListener( l: this); //이벤트 발생
23         add(button, BorderLayout.SOUTH);
24     }
25     public void paintComponent(Graphics g)
26     {
27         super.paintComponent(g);
28
29         Graphics2D g2 = (Graphics2D) g;
30
31         g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING, RenderingHints.VALUE_ANTIALIAS_ON);
32         g2.setColor(Color.BLACK);
33         g2.setStroke(new BasicStroke( width: 5)); //선두께
34
35         g.drawLine( x1: 90, y1: 10, x2: 90, y2: 300); //세로
36         g.drawLine( x1: 70, y1: 60, x2: 110, y2: 60); //세로 숫자 줄 4
37         g.drawLine( x1: 70, y1: 120, x2: 110, y2: 120); //세로 숫자 줄 3
38         g.drawLine( x1: 70, y1: 180, x2: 110, y2: 180); //세로 숫자 줄 2
39         g.drawLine( x1: 70, y1: 240, x2: 110, y2: 240); //세로 숫자 줄 1
40         g.drawLine( x1: 90, y1: 300, x2: 400, y2: 300); //가로
41         g.setFont(new Font( name: "굴림체", Font.PLAIN, size: 20)); //폰트 지정
42         g.drawString( str: "4", x: 45, y: 60); //범례(legend)
43         g.drawString( str: "3", x: 45, y: 120); //범례(legend)
44         g.drawString( str: "2", x: 45, y: 180); //범례(legend)
45         g.drawString( str: "1", x: 45, y: 240); //범례(legend)
46
47         g.drawString( str: "이과", x: 102, y: 320); //이과
48         g.drawString( str: "문과", x: 182, y: 320); //문과
49         g.drawString( str: "예체능", x: 260, y: 320); //예체능
50         g.drawString( str: "기타", x: 342, y: 320); //기타
51     }
52 }
```

```
50 //2022현재
51 g2.setPaint(color1);
52 g.drawString( str: "2022", x: 380, y: 30); //년도
53 g2.fill(new Ellipse2D.Float( x: 105, y: 115, w: 15, h: 15)); //원 이과
54 g.drawLine( x1: 110, y1: 120, x2: 190, y2: 180); //이과 -> 문과
55 g2.fill(new Ellipse2D.Float( x: 185, y: 175, w: 15, h: 15)); //원 문과
56 g.drawLine( x1: 190, y1: 180, x2: 270, y2: 60); //문과 -> 예체능
57 g2.fill(new Ellipse2D.Float( x: 265, y: 55, w: 15, h: 15)); //원 예체능
58 g.drawLine( x1: 270, y1: 60, x2: 350, y2: 240); //예체능 -> 기타
59 g2.fill(new Ellipse2D.Float( x: 345, y: 235, w: 15, h: 15)); //원 기타
60
61 //2021과거
62 g2.setPaint(color2);
63 g.drawString( str: "2021", x: 380, y: 45); //년도
64 g2.fill(new Ellipse2D.Float( x: 105, y: 55, w: 15, h: 15)); //원 이과
65 g.drawLine( x1: 110, y1: 60, x2: 190, y2: 180); //이과4 -> 문과2
66 g2.fill(new Ellipse2D.Float( x: 185, y: 175, w: 15, h: 15)); //원 문과
67 g.drawLine( x1: 190, y1: 180, x2: 270, y2: 240); //문과2 -> 예체능1
68 g2.fill(new Ellipse2D.Float( x: 265, y: 235, w: 15, h: 15)); //원 예체능
69 g.drawLine( x1: 270, y1: 240, x2: 350, y2: 120); //예체능1 -> 기타3
70 g2.fill(new Ellipse2D.Float( x: 345, y: 115, w: 15, h: 15)); //원 기타
71 }
72 public void actionPerformed(ActionEvent e){
73     color1 = new Color((int)(Math.random()*255.0),
74         (int)(Math.random()*255.0),
75         (int)(Math.random()*255.0));
76     color2 = new Color((int)(Math.random()*255.0),
77         (int)(Math.random()*255.0),
78         (int)(Math.random()*255.0));
79     repaint(); //event 에 대한 paint 함수 다시 호출함
```

```
1 import javax.swing.*;
2 import java.awt.*;
3 import java.awt.event.*;
4 import java.awt.geom.*;
5 import java.util.*;
6
7 2 usages
8 public class pchartPanel extends JPanel implements ActionListener{
9     ArrayList<Shape> shapeArrayList = new ArrayList<>();
10    3 usages
11    JButton button;
12    2 usages
13    Color color1 = new Color( r: 0, g: 0, b: 0);
14    2 usages
15    Color color2 = new Color( r: 0, g: 0, b: 0);
16    2 usages
17    Color color3 = new Color( r: 0, g: 0, b: 0);
18    2 usages
19    Color color4 = new Color( r: 0, g: 0, b: 0);
20    2 usages
21    public pchartPanel(){
22        setLayout(new BorderLayout());
23        //버튼 생성
24        button = new JButton( text: "Color Change");
25        button.addActionListener( l: this); //이벤트 발생
26        add(button, BorderLayout.SOUTH);
27    }
28 }
29
30 2 usages
31 public void paintComponent(Graphics g)
32 {
33     super.paintComponent(g);
34
35     Graphics2D g2 = (Graphics2D) g;
36
37     g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING, RenderingHints.VALUE_ANTIALIAS_ON);
38     g2.setColor(Color.BLACK);
39     g2.setStroke(new BasicStroke( width: 5)); //선두께
40
41     g.drawLine( x1: 90, y1: 10, x2: 90, y2: 300); //세로
42     g.drawLine( x1: 70, y1: 60, x2: 110, y2: 60); //세로 숫자 줄
43     g.drawLine( x1: 70, y1: 120, x2: 110, y2: 120); //세로 숫자 줄
44     g.drawLine( x1: 70, y1: 180, x2: 110, y2: 180); //세로 숫자 줄
45     g.drawLine( x1: 70, y1: 240, x2: 110, y2: 240); //세로 숫자 줄
46     g.drawLine( x1: 90, y1: 300, x2: 400, y2: 300); //가로
47     g.setFont(new Font( name: "굴림체", Font.PLAIN, size: 20)); //폰트 지정
48     g.drawString( str: "4", x: 45, y: 60); //범례(legend)
49     g.drawString( str: "3", x: 45, y: 120); //범례(legend)
50     g.drawString( str: "2", x: 45, y: 180); //범례(legend)
51     g.drawString( str: "1", x: 45, y: 240); //범례(legend)
52
53     g.drawString( str: "이과", x: 102, y: 320); //이과
54     g.drawString( str: "문과", x: 182, y: 320); //문과
55     g.drawString( str: "예체능", x: 260, y: 320); //예체능
56     g.drawString( str: "기타", x: 342, y: 320); //기타
57 }
```

막대

```
22 public void paintComponent(Graphics g)
23 {
24     super.paintComponent(g);
25
26     Graphics2D g2 = (Graphics2D) g;
27
28     g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING, RenderingHints.VALUE_ANTIALIAS_ON);
29     g2.setColor(Color.BLACK);
30     g2.setStroke(new BasicStroke( width: 5)); //선두께
31
32     g.drawLine( x1: 90, y1: 10, x2: 90, y2: 300); //세로
33     g.drawLine( x1: 70, y1: 60, x2: 110, y2: 60); //세로 숫자 줄
34     g.drawLine( x1: 70, y1: 120, x2: 110, y2: 120); //세로 숫자 줄
35     g.drawLine( x1: 70, y1: 180, x2: 110, y2: 180); //세로 숫자 줄
36     g.drawLine( x1: 70, y1: 240, x2: 110, y2: 240); //세로 숫자 줄
37     g.drawLine( x1: 90, y1: 300, x2: 400, y2: 300); //가로
38     g.setFont(new Font( name: "굴림체", Font.PLAIN, size: 20)); //폰트 지정
39     g.drawString( str: "4", x: 45, y: 60); //범례(legend)
40     g.drawString( str: "3", x: 45, y: 120); //범례(legend)
41     g.drawString( str: "2", x: 45, y: 180); //범례(legend)
42     g.drawString( str: "1", x: 45, y: 240); //범례(legend)
43
44     g.drawString( str: "이과", x: 102, y: 320); //이과
45     g.drawString( str: "문과", x: 182, y: 320); //문과
46     g.drawString( str: "예체능", x: 260, y: 320); //예체능
47     g.drawString( str: "기타", x: 342, y: 320); //기타
48 }
```

```
51
52 g2.setPaint(color1);
53 g2.fill(new Rectangle2D.Float( x: 110, y: 120, w: 30, h: 180)); //사각형
54
55 g2.setPaint(color2);
56 g2.fill(new Rectangle2D.Float( x: 190, y: 180, w: 30, h: 120)); //사각형
57
58 g2.setPaint(color3);
59 g2.fill(new Rectangle2D.Float( x: 270, y: 60, w: 30, h: 240)); //사각형
60
61 g2.setPaint(color4);
62 g2.fill(new Rectangle2D.Float( x: 350, y: 240, w: 30, h: 60)); //사각형
63 }
64 public void actionPerformed(ActionEvent e){
65     color1 = new Color((int)(Math.random()*255.0),
66         (int)(Math.random()*255.0),
67         (int)(Math.random()*255.0));
68     color2 = new Color((int)(Math.random()*255.0),
69         (int)(Math.random()*255.0),
70         (int)(Math.random()*255.0));
71     color3 = new Color((int)(Math.random()*255.0),
72         (int)(Math.random()*255.0),
73         (int)(Math.random()*255.0));
74     color4 = new Color((int)(Math.random()*255.0),
75         (int)(Math.random()*255.0),
76         (int)(Math.random()*255.0));
77     repaint(); //event 에 대한 paint 함수 다시 호출함
78 }
79 }
```

원 chart

```
1 import javax.swing.*;
2 import java.awt.*;
3 import java.awt.event.*;
4 import java.awt.geom.*;
5 import java.util.*;
6
7 2 usages
8 public class circlecartPanel extends JPanel implements ActionListener {
9     2 usages
10     int num1 =3; // 이과의 수
11     2 usages
12     int num2 =2; // 문과의 수
13     2 usages
14     int num3 =4; // 예체능의 수
15     1 usage
16     int num4 =1; // 기타의 수
17
18     3 usages
19     JButton button;
20     3 usages
21     Color color1 = new Color( r:0, g:0, b:0);
22     3 usages
23     Color color2 = new Color( r:0, g:0, b:0);
24     3 usages
25     Color color3 = new Color( r:0, g:0, b:0);
26     3 usages
27     Color color4 = new Color( r:0, g:0, b:0);
28     2 usages
29     z usages
30     public circlecartPanel(){
31         setLayout(new BorderLayout());
32         //버튼 생성
33         button = new JButton( text: "Color Change");
34         button.addActionListener( this); //이벤트 발생
35         add(button,BorderLayout.SOUTH);
36     }
37
38     public void paintComponent (Graphics g){
39         super.paintComponent(g);
40
41         Graphics2D g2 = (Graphics2D) g;
42
43         g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING, RenderingHints.VALUE_ANTIALIAS_ON);
44         GradientPaint gp = new GradientPaint( x1:0, y1:10,Color.white, x2:0, y2:70,Color.RED);
45
46         //전체 합을 구한다.
47         int total = num1 + num2 + num3 + num4;
48         if (total == 0)
49             return;
50         // 전체에서의 비중을 구함.
51         //arc4 = 전체 - (arc1+arc2+arc3)로 구함
52
53         int arc1 = (int) 360.0 * num1 / total;
54         int arc2 = (int) 360.0 * num2 / total;
55         int arc3 = (int) 360.0 * num3 / total;
56         g.setColor(color1); //색상지정
57         g.fillArc( x:50, y:20, width:200, height:200, startAngle:0, arc1,arc2); //(x축,y축,반지름,반지름,시작각,끝각) - 원호를 그림
58         g.setColor(color2); //색상지정
59         g.fillArc( x:50, y:20, width:200, height:200, startAngle: arc1 + arc2, arc3); //(x축,y축,반지름,반지름,시작각,끝각) - 원호를 그림
60         g.setColor(color4); //색상지정
61         g.fillArc( x:50, y:20, width:200, height:200, startAngle: arc1 + arc2 + arc3, arcAngle: 360 - (arc1 + arc2 + arc3)); //(x축,y축,반지름,반지름,시작각,끝각) - 원호를 그림
62         g.setColor(Color.BLACK); //색상지정
63         g.setFont(new Font( name: "굴림체", Font.PLAIN, size:12)); //폰트 지정
64         g.drawString( str: " 이과 비율", x:300, y:150); //범례(legend)
65         g.drawString( str: " 문과 비율", x:300, y:170); //범례(legend)
66         g.drawString( str: " 예체능 비율", x:300, y:190); //범례(legend)
67         g.drawString( str: " 기타 비율", x:300, y:210); //범례(legend)
68         g2.setPaint(color1);
69         g2.fill(new Ellipse2D.Float( x:290, y:140, w:10, h:10)); //원
70         g2.setPaint(color2);
71         g2.fill(new Ellipse2D.Float( x:290, y:160, w:10, h:10)); //원
72         g2.setPaint(color3);
73         g2.fill(new Ellipse2D.Float( x:290, y:180, w:10, h:10)); //원
74         g2.setPaint(color4);
75         g2.fill(new Ellipse2D.Float( x:290, y:200, w:10, h:10)); //원
76     }
77 }
```

```
g.fillArc( x:50, y:20, width:200, height:200, startAngle:0, arc1,arc2); //(x축,y축,반지름,반지름,시작각,끝각) - 원호를 그림
g.setColor(color2); //색상지정
g.fillArc( x:50, y:20, width:200, height:200, startAngle: arc1 + arc2, arc3); //(x축,y축,반지름,반지름,시작각,끝각) - 원호를 그림
g.setColor(color4); //색상지정
g.fillArc( x:50, y:20, width:200, height:200, startAngle: arc1 + arc2 + arc3, arcAngle: 360 - (arc1 + arc2 + arc3)); //(x축,y축,반지름,반지름,시작각,끝각) - 원호를 그림
g.setColor(Color.BLACK); //색상지정
g.setFont(new Font( name: "굴림체", Font.PLAIN, size:12)); //폰트 지정
g.drawString( str: " 이과 비율", x:300, y:150); //범례(legend)
g.drawString( str: " 문과 비율", x:300, y:170); //범례(legend)
g.drawString( str: " 예체능 비율", x:300, y:190); //범례(legend)
g.drawString( str: " 기타 비율", x:300, y:210); //범례(legend)
g2.setPaint(color1);
g2.fill(new Ellipse2D.Float( x:290, y:140, w:10, h:10)); //원
g2.setPaint(color2);
g2.fill(new Ellipse2D.Float( x:290, y:160, w:10, h:10)); //원
g2.setPaint(color3);
g2.fill(new Ellipse2D.Float( x:290, y:180, w:10, h:10)); //원
g2.setPaint(color4);
g2.fill(new Ellipse2D.Float( x:290, y:200, w:10, h:10)); //원
}
```

```
67 }
68 public void actionPerformed(ActionEvent e){
69     color1 = new Color((int)(Math.random()*255.0),
70         (int)(Math.random()*255.0),
71         (int)(Math.random()*255.0));
72     color2 = new Color((int)(Math.random()*255.0),
73         (int)(Math.random()*255.0),
74         (int)(Math.random()*255.0));
75     color3 = new Color((int)(Math.random()*255.0),
76         (int)(Math.random()*255.0),
77         (int)(Math.random()*255.0));
78     color4 = new Color((int)(Math.random()*255.0),
79         (int)(Math.random()*255.0),
80         (int)(Math.random()*255.0));
81     repaint(); //event 에 대한 paint 함수 다시 호출함
82 }
83 }
84 }
```



```

Main.java × lineChartPanel.java × pchartPanel.java × circlecartPanel.java × MyChartPanel.java
22주 객체지향 11week javax.swing.*;
2 import java.awt.*;
3 import java.awt.event.*;
4 import java.awt.geom.*;
5 import java.util.*;
6
7 1 usage
8 public class MyChartPanel extends JPanel implements ActionListener {
9     ArrayList<Shape> shapeArrayList = new ArrayList<>();
10
11     3 usages
12 JButton button;
13
14 4 usages
15 Color color1 = new Color(0, 0, 0);
16 1 usage
17 Color color2 = new Color(0, 0, 0);
18 1 usage
19 public MyChartPanel(){
20     setLayout(new BorderLayout());
21     //버튼 생성
22     button = new JButton("Color Change");
23     button.addActionListener(this); //이벤트 발생
24     add(button, BorderLayout.SOUTH);
25 }

```

```

22 public void paintComponent(Graphics graphics) {
23
24     super.paintComponent(graphics);
25
26     Graphics2D g2 = (Graphics2D) graphics;
27
28     g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING, RenderingHints.VALUE_ANTIALIAS_ON);
29     g2.setColor(Color.BLACK);
30     g2.setStroke(new BasicStroke(5)); //선두께
31
32
33     graphics.setColor(Color.white);
34     graphics.fillRect(0, 0, 370, 450);
35     Color blue = new Color(0, 100, 255);
36     graphics.setColor(blue);
37     graphics.fillRect(65, 60, 240, 230);
38     graphics.fillRect(109, 290, 150, 120);
39
40     int xValues[] = {109, 70, 63, 109};
41     int yValues[] = {290, 278, 308, 321};
42
43     graphics.fillPolygon(xValues, yValues, 4);
44
45     int xValues2[] = {259, 305, 295, 259};
46     int yValues2[] = {290, 308, 332, 321};
47
48     graphics.fillPolygon(xValues2, yValues2, 4);
49 }

```

나만의 Chart

```

50 graphics.setColor(Color.white);
51 graphics.fillOval( x: 41, y: 271, width: 40, height: 40);
52 graphics.fillOval( x: 292, y: 305, width: 40, height: 40);
53 graphics.fillOval( x: 73, y: 402, width: 20, height: 20);
54 graphics.fillOval( x: 90, y: 405, width: 90, height: 25);
55 graphics.fillOval( x: 186, y: 405, width: 90, height: 25);
56 graphics.setColor(Color.black);
57 graphics.drawOval( x: 41, y: 271, width: 40, height: 40);
58 graphics.drawOval( x: 292, y: 305, width: 40, height: 40);
59 graphics.drawOval( x: 90, y: 405, width: 90, height: 35);
60 graphics.drawOval( x: 186, y: 405, width: 90, height: 35);
61
62
63 graphics.setColor(Color.white);
64 graphics.fillOval( x: 85, y: 100, width: 200, height: 180);
65 graphics.fillOval( x: 123, y: 280, width: 120, height: 112);
66 graphics.setColor(Color.red);
67 graphics.fillRoundRect( x: 103, y: 275, width: 166, height: 112);
68 graphics.setColor(new Color( r: 204, g: 204, b: 0));
69 graphics.fillOval( x: 169, y: 278, width: 30, height: 30);
70 graphics.setColor(Color.black);
71 graphics.drawLine( x1: 171, y1: 285, x2: 197, y2: 285);
72 graphics.drawLine( x1: 169, y1: 290, x2: 200, y2: 290);
73 graphics.fillOval( x: 179, y: 293, width: 10, height: 10);
74 graphics.drawLine( x1: 184, y1: 300, x2: 184, y2: 307);
75

```

```

76
77 graphics.drawArc( x: 148, y: 290, width: 70, height: 70, startAngle: 0, arcAngle: -180);
78 graphics.drawLine( x1: 148, y1: 325, x2: 218, y2: 325);
79 graphics.setColor(Color.white);
80 graphics.fillOval( x: 123, y: 80, width: 60, height: 70);
81 graphics.fillOval( x: 183, y: 80, width: 60, height: 70);
82 graphics.setColor(color1);
83 graphics.drawOval( x: 123, y: 80, width: 60, height: 70);
84 graphics.drawOval( x: 183, y: 80, width: 60, height: 70);
85 graphics.fillOval( x: 160, y: 108, width: 20, height: 20);
86 graphics.fillOval( x: 186, y: 108, width: 20, height: 20);
87 graphics.setColor(Color.white);
88 graphics.fillOval( x: 167, y: 114, width: 6, height: 6);
89 graphics.fillOval( x: 190, y: 114, width: 6, height: 6);
90 graphics.setColor(Color.red);
91 graphics.fillOval( x: 169, y: 158, width: 27, height: 27);
92 graphics.setColor(color1);
93 graphics.drawOval( x: 169, y: 158, width: 27, height: 27);
94 graphics.setColor(Color.white);
95 graphics.fillOval( x: 184, y: 161, width: 10, height: 13);
96
97

```



```
98
99
100 graphics.setColor(color1);
101 graphics.drawArc( x: 90, y: 60, width: 190, height: 190, startAngle: -45, arcAngle: -90);
102 graphics.drawLine( x1: 183, y1: 185, x2: 183, y2: 250);
103 graphics.drawLine( x1: 100, y1: 182, x2: 148, y2: 189);
104 graphics.drawLine( x1: 93, y1: 200, x2: 148, y2: 200);
105 graphics.drawLine( x1: 98, y1: 217, x2: 148, y2: 211);
106
107 graphics.drawLine( x1: 219, y1: 189, x2: 268, y2: 182);
108 graphics.drawLine( x1: 219, y1: 200, x2: 272, y2: 200);
109 graphics.drawLine( x1: 219, y1: 211, x2: 269, y2: 218);
110
111 graphics.setFont(new Font( name: "kai body",Font.PLAIN, size: 24));
112
113
114
115
116 }
117 public void actionPerformed(ActionEvent e){
118     color1 = new Color((int)(Math.random()*255.0),
119         (int)(Math.random()*255.0),
120         (int)(Math.random()*255.0));
121     color2 = new Color((int)(Math.random()*255.0),
122         (int)(Math.random()*255.0),
123         (int)(Math.random()*255.0));
124
125     repaint(); //event 에 대한 paint 함수 다시 호출함
126 }
127 }
```

결과 화면

