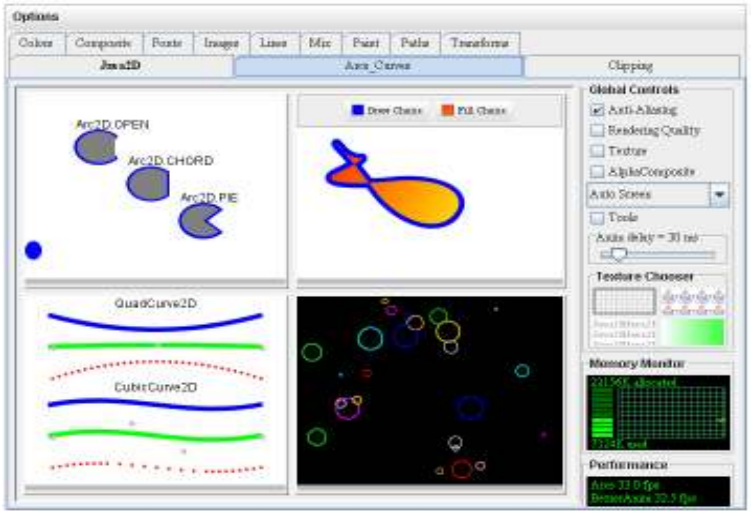
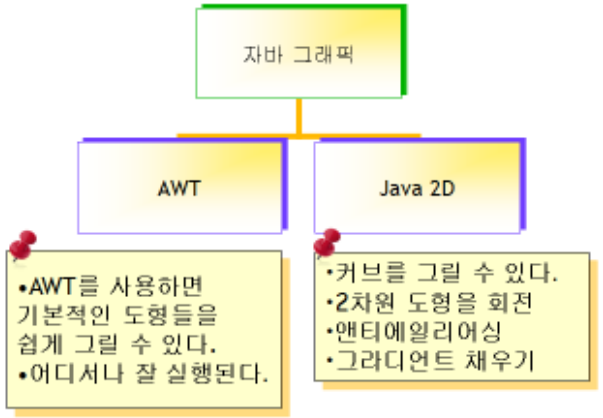
그래픽스, 이미지

514760-1
2016년 가을학기
11/10/2016
박경신

자바에서의 그래픽



자바 그래픽의 두가지 방법



간단한 그래픽 예제

```
// (1) 프레임 생성하기
public class BasicPaint {

  public static void main(String[] args) {
  JFrame f = new JFrame("그래픽 기초 프로그램");
  f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
  f.setSize(300, 200);
  f.setVisible(true);
  }
}
```

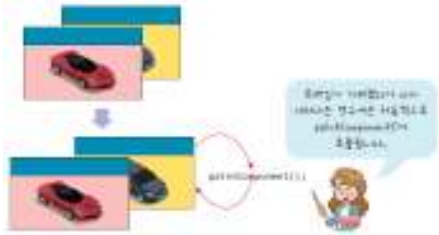
간단한 그래픽 예제

- JFrame을 생성하고 여기에 JPanel을 추가한 후에 JPanel 위에 그림을 그려보자.

```
// (2) Jframe을 생성하고 여기에 Jpanel을 추가한 후 Jpanel 위에 그림을 그린다.
public class BasicPaint {
  public static void main(String[] args) {
  JFrame f = new JFrame("그래픽 기초 프로그램");
  f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
  f.add(new MyPanel());
  f.setSize(300, 200);
  f.setVisible(true);
  }
}
class MyPanel extends JPanel {

  public MyPanel() {
  ...

  }
}
```

간단한 그래픽 예제

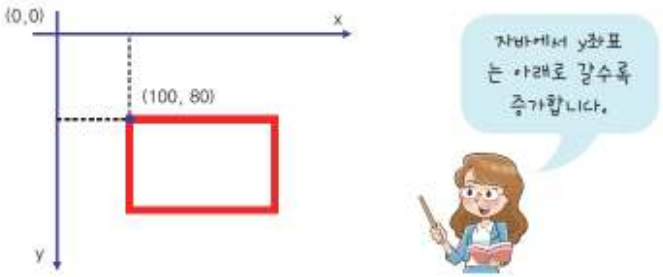
- 컴포넌트에 무언가를 그리려면 paintComponent() 메소드를 중복 정의한다.
- paintComponent() 메소드는 컴포넌트가 화면에 그려질 때 호출된다.



```
class MyPanel extends Jpanel {
        public void paintComponent(Graphics g)
        {
                super.paintComponent(g);
                // 여기에 그림을 그리는 코드를 넣는다.
        }
}
```

간단한 그래픽 예제

- 그래픽 좌표계는 x+ 오른쪽, y+ 아래쪽으로 갈수록 증가한다.



간단한 그래픽 예제

- 사각형을 그리려면 Graphics 객체가 가지고 있는 drawRect()을 호출하면 된다.

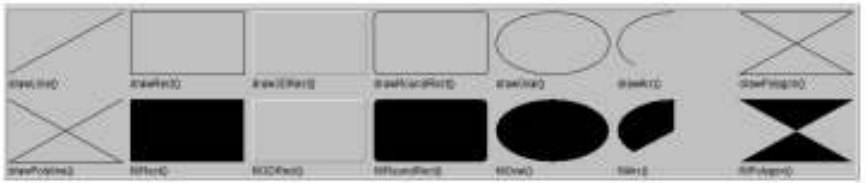
```
g.drawRect(50, 50, 50, 50);
g.drawOval(200, 50, 50, 50);
```

전체 소스

```
public class BasicPaint {
  public static void main(String[] args) {
            JFrame f = new JFrame("그래픽 기초 프로그램");
            f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
            f.add(new MyPanel());
            f.setSize(300, 200);
            f.setVisible(true);
  }
}
class MyPanel extends JPanel {
  public MyPanel() {
            setBorder(BorderFactory.createLineBorder(Color.black));
  }
  protected void paintComponent(Graphics g) {
            super.paintComponent(g);
            g.drawRect(50, 50, 50, 50);
            g.drawOval(200, 50, 50, 50);
  }
}
```

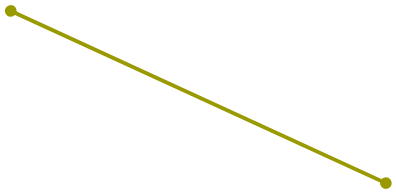
기초 도형 그리기

기초 도형	관련된 메소드
직선	drawLine(), drawPolyline()
사각형	drawRect(), fillRect(), clearRect()
3차원 사각형	draw3DRect(), fill3DRect()
둥근 사각형	drawRoundRect(), fillRoundRect()
타원	drawOval(), fillOval()
호	drawArc(), fillArc()
다각형	drawPolygon(), fillPolygon()



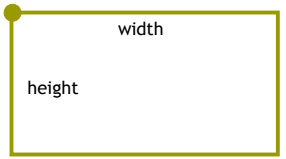
직선 그리기

메 소 드	설 명
drawLine(int x1, int y1, int x2, int y2)	좌표 (x1,y1)에서 좌표 (x2,y2) 까지 직선을 그린다.
drawPolyline(int[] xpoints, int[] ypoints, int numpoints)	배열 xpoints[]와 배열 ypoints[]을 가지고 여러 개의 직선을 그린다. polygon과 다른 점은 첫 번째 점과 마지막 점이 연결되지 않는다.



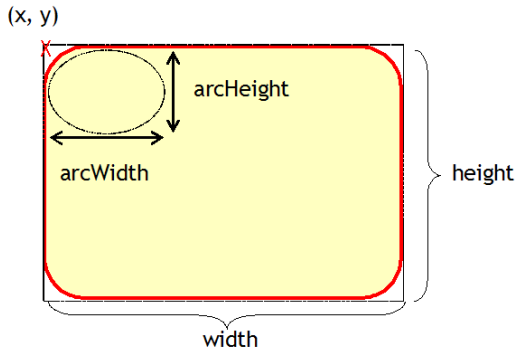
사각형 그리기

메소드 및 설명
drawRect(int x, int y, int width, int height) // 왼쪽 상단 좌표 (x, y)
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
drawRoundRect(int x, int y, int width, int height, int arcWidth, int arcHeight)
fillRoundRect(int x, int y, int width, int height, int arcWidth, int arcHeight)



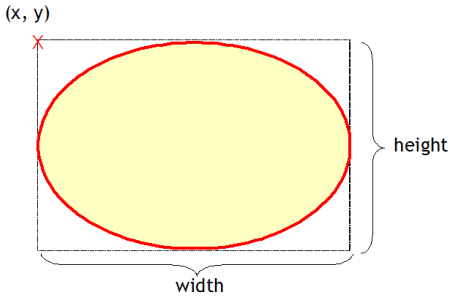
drawRoundRect()

- 시작점(x, y)과 넓이(width)와 높이(height)



타원 그리기

메 소 드	설 명
drawOval(int x, int y, int width, int height)	좌측 상단의 좌표가 x,y이며 폭 width, 높이 height의 사각형 안에 내접하는 타원을 그린다.
fillOval(int x, int y, int width, int height)	채워진 타원을 그린다.



호그리기

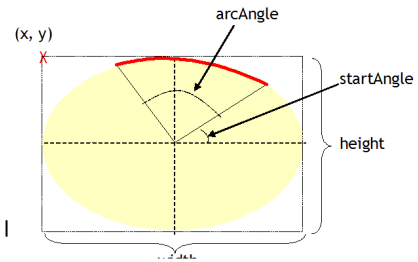


그림 23-6 drawOval() 매개 변수의 의미

메 소 드	설 명
drawArc(int x, int y, int width, int height, int startAngle, int arcAngle)	좌측 상단의 좌표가 x, y이며 폭 width, 높이 height의 사각형 안에 내접하는 타원을 startAngle을 시작 각도로 하여 arcAngle의 각도만큼의 호를 그린다.
fillArc(int x, int y, int width, int height, int startAngle, int arcAngle)	좌측 상단의 좌표가 x,y이며 폭 width, 높이 height의 사각형 안에 내접하는 타원을 startAngle을 시작각도로 하여 arcAngle의 각도만큼의 채워진 호를 그린다.

예제: 얼굴그리기

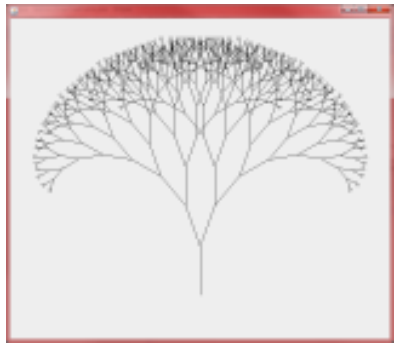
```
import javax.swing.*;
import java.awt.event.*;
import java.awt.*;

class MyPanel extends JPanel {

    public void paintComponent(Graphics g) {
        super.paintComponent(g);
        g.setColor(Color.YELLOW);
        g.fillOval(20, 30, 200, 200);
        g.setColor(Color.BLACK);
        // 왼쪽 눈을 그린다.
        g.drawArc(60, 80, 50, 50, 180, -180);
        // 오른쪽 눈을 그린다.
        g.drawArc(150, 80, 50, 50, 180, -180);
        // 입을 그린다.
        g.drawArc(70, 130, 100, 70, 180, 180);
    }
}
```

예제: 얼굴 그리기

```
public class SnowManFace extends JFrame {
    public SnowManFace() {
        setSize(280, 300);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setTitle("눈사람 얼굴");
        setVisible(true);
        add(new MyPanel());
    }

    public static void main(String[] args) {
        SnowManFace s=new SnowManFace();
    }
}
```

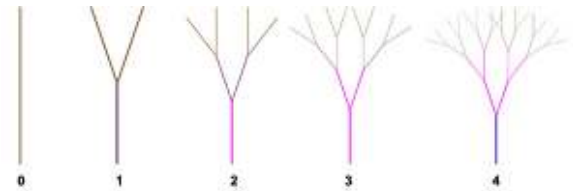
예제: 프랙탈로 나무 그리기

- 프랙탈(fractal)은 자기 유사성을 가지는 기하학적 구조를 프랙털 구조를 말한다.



프랙탈 트리를 그리는 알고리즘

① 나무 줄기를 그린다.
② 줄기의 끝에서 특정한 각도로 2개의 가지를 그린다.
③ 동일한 과정을 가지의 끝에서 반복한다. 충분한 가지가 생성될 때까지 이 과정을 반복한다.

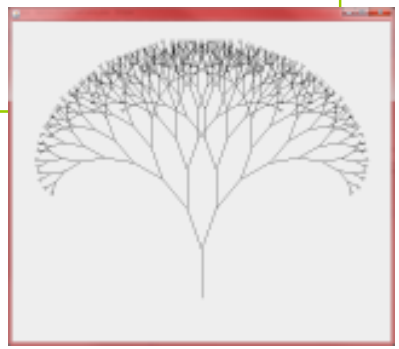


예제: 프랙탈로 나무 그리기

```
import java.awt.Color;
import java.awt.Graphics;
import javax.swing.JFrame;

public class DrawTreeFrame extends JFrame {

  public DrawTreeFrame() {
          setSize(800, 700);
          setDefaultCloseOperation(EXIT_ON_CLOSE);
          setVisible(true);
  }

  private void drawTree(Graphics g, int x1, int y1, double angle, int depth) {
          if (depth == 0)
                  return;
          int x2 = x1 + (int) (Math.cos(Math.toRadians(angle)) * depth * 10.0);
          int y2 = y1 + (int) (Math.sin(Math.toRadians(angle)) * depth * 10.0);
          g.drawLine(x1, y1, x2, y2);
          drawTree(g, x2, y2, angle - 20, depth - 1);
          drawTree(g, x2, y2, angle + 20, depth - 1);
  }
```

예제: 프랙탈로 나무 그리기

```
@Override
public void paint(Graphics g) {
        g.setColor(Color.BLACK);
        drawTree(g, 400, 600, -90, 10);
}

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



색상

- java.awt 패키지의 일부인 Color 클래스를 사용
- 빛의 3원색인 Red 성분, Green 성분, Blue 성분이 얼마나 함유되어 있는지를 0에서 255까지의 수를 사용하여 나타낸다.

색상

클래스 변수 이름	색 상	RGB 값
Color.black	black	(0,0,0)
Color.blue	blue	(0,0,255)
Color.cyan	cyan	(0,255,255)
Color.gray	gray	(128,128,128)
Color.darkGray	dark gray	(64,64,64)
Color.lightGray	light gray	(192,192,192)
Color.green	green	(0,255,0)
Color.magenta	magenta	(255,0,255)
Color.orange	orange	(255,200,0)
Color.pink	pink	(255,175,175)
Color.red	red	(255,0,0)
Color.white	white	(255,255,255)
Color.yellow	yellow	(255,255,0)

색상 설정

- 마젠타 색상을 얻는 방법

① Color c = Color.magenta;

② Color c = **new** Color (255,0,255);

- Color는 알파값(alpha)을 가질 수 있다. 알파값이란 색상의 투명도를 나타낸다.
- E.g. Color c = **new** Color (255, 0, 0, 128);

컴포넌트 색상 설정 메소드

생 성 자	설 명
setBackground(Color c)	컴포넌트 객체에서 배경색을 설정한다.
setColor(Color c)	전경색을 설정한다.
Color getColor()	현재의 전경색을 반환한다.

예제

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

class MyPanel extends JPanel implements ActionListener {
        JButton button;
        Color color = new Color(0, 0, 0);

        public MyPanel() {
                setLayout(new BorderLayout());
                button = new JButton("색상 변경");
                button.addActionListener(this);
                add(button, BorderLayout.SOUTH);
        }

        public void paintComponent(Graphics g) {
               super.paintComponent(g);
                g.setColor(color);
                g.fillRect(10, 10, 210, 220);
        }
```

예제

```
        public void actionPerformed(ActionEvent e) {
                color = new Color((int) (Math.random()*255.0),
                                (int) (Math.random()*255.0),(int) (Math.random()*255.0));
                repaintComponent();
        }
}

public class ColorTest extends JFrame {
        public ColorTest() {
                setSize(240, 300);
                setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                setTitle("Color Test");
                setVisible(true);
                JPanel panel = new MyPanel();
                add(panel);
        }

        public static void main(String[] args) {
                ColorTest s = new ColorTest();
        }
}
```

실행 결과

색상 선택기

예제

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

public class ColorChooserTest extends JFrame implements ChangeListener {

    protected JColorChooser color;

    public ColorChooserTest() {
        setTitle("색상 선택기  테스트");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        color = new JColorChooser();    // 생성자 호출
        color.getSelectionModel().addChangeListener(this);    // 리스너 등록
        color.setBorder(BorderFactory.createTitledBorder("색상 선택"));
```

예제

```
        JPanel panel = new JPanel();
        panel.add(color);
        add(panel);
        pack();
        this.setVisible(true);

    }

    public void stateChanged(ChangeEvent e) {
        Color newColor = color.getColor();
    }

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

문자열 출력과 폰트

- 문자열 출력 방법

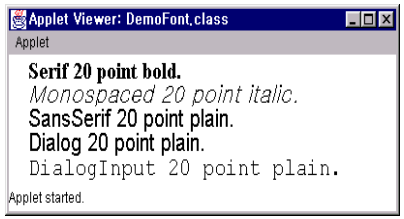
```
// (x, y) 위치에 문자열을 출력하려면
g.drawString("Hello World!", x, y);
```

- 폰트를 지정하기 위해서는 Font 클래스를 사용
- Font 객체는 폰트 이름(Courier, Helvetica,..) 과 스타일(plain, bold, italic,...), 크기(12포인트,...)의 3가지 속성

```
// plain 형식이고 크기는 10포인트
Font font = new Font("Courier", Font.PLAIN, 10);
```

폰트의 종류

논리적인 폰트	설 명
"Serif"	삐침(serif)를 갖는 가변폭 글꼴, 대표적으로 TimesRoman이 있다.
"SansSerif"	삐침(serif)를 갖지않는 가변폭 글꼴, 대표적으로 Helvetica가 있다.
"Monospaced"	고정폭을 가지는 글꼴, 대표적으로 Courier가 있다.
"Dialog"	대화상자에서 텍스트 출력을 위하여 사용되는 글꼴
"DialogInput"	대화상자에서 텍스트 입력을 위하여 사용되는 글꼴



폰트의 지정

```
public void paint(Graphics g)
{
Font f = new Font("Serif", Font.BOLD | Font.ITALIC, 12);
g.setFont(f);
...
}
```

```
JLabel myLabel = new JLabel("폰트 색상");
Font f = new Font("Dialog", Font.ITALIC, 10);          // ①
myLabel.setFont(f);
```

예제

```
...
class MyPanel extends JPanel {
        Font f1, f2, f3, f4, f5;
        public MyPanel() {
            f1 = new Font("Serif", Font.PLAIN, 20);
            f2 = new Font("San Serif", Font.BOLD, 20);
            f3 = new Font("Monospaced", Font.ITALIC, 20);
            f4 = new Font("Dialog", Font.BOLD | Font.ITALIC, 20);
            f5 = new Font("DialogInput", Font.BOLD, 20);
        }
        public void paintComponent(Graphics g) {
            super.paintComponent(g);
            g.setFont(f1);
            g.drawString("Serif 20 points PLAIN", 10, 50);
            g.setFont(f2);
            g.drawString("SanSerif 20 points BOLD", 10, 70);
            g.setFont(f3);
            g.drawString("Monospaced 20 points ITALIC", 10, 90);
            g.setFont(f4);
            g.drawString("Dialog 20 points BOLD + ITALIC", 10, 110);
            g.setFont(f5);
            g.drawString("DialogInput 20 points BOLD", 10, 130);
        }
}
```

예제

```
public class FontTest extends JFrame {
        public FontTest() {
            setSize(500, 200);
            setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
            setTitle("Font Test");
            setVisible(true);
            JPanel panel = new MyPanel();
            add(panel);
        }

        public static void main(String[] args) {
            FontTest s = new FontTest();
        }
}
```

이미지 출력

- 자바는 GIF, PNG JPEG 타입의 이미지를 화면에 그릴 수 있다.

```
BufferedImage img = null;
try {
    img = ImageIO.read(new File("strawberry.jpg"));
} catch (IOException e) {
    ...
}
```

예제

```
// 소스를 입력하고 Ctrl+Shift+O를 눌러서 필요한 파일을 포함한다.

public class LoadImageApp extends JPanel {

    BufferedImage img;

    public void paint(Graphics g) {
        g.drawImage(img, 0, 0, null);
    }

    public LoadImageApp() {
        try {
            img = ImageIO.read(new File("dog.png"));
        } catch (IOException e) {
        }

    }
```

아직 학습하지 않았지만 파일을 읽을 때 오류를 처리하는 코드이다.

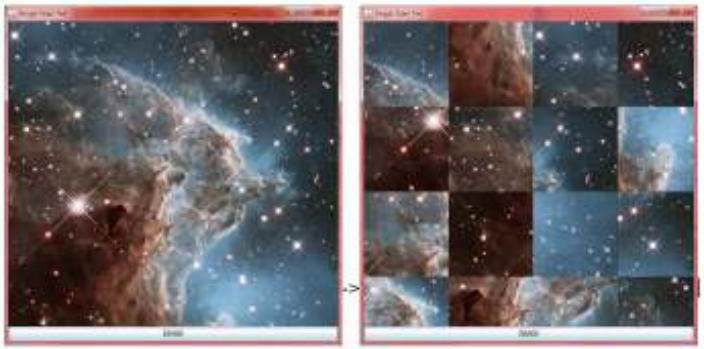
예제

```
    public Dimension getPreferredSize() {
        if (img == null) {
            return new Dimension(100, 100);
        } else {
            return new Dimension(img.getWidth(null),
img.getHeight(null));
        }
    }

    public static void main(String[] args) {

        JFrame f = new JFrame("이미지 표시 예제");

        f.add(new LoadImageApp());
        f.pack();
        f.setVisible(true);
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    }
}
```

실행 결과

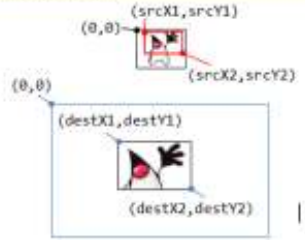
LAB: 이미지 나누어서 그리기

- drawImage()를 이용하여 이미지를 그릴 때, 일부만 그릴 수 있고 또 크기를 변경할 수 있다. 이러한 기능을 이용하여서 이미지를 16조각으로 나누어서 그리는 프로그램을 작성하여 보자.



drawImage() 메소드

```
boolean drawImage(Image img,
                  int dstx1, int dsty1, int dstx2,  int dsty2,
                  int srcx1, int srcy1, int srcx2, int srcy2,
                  ImageObserver observer);
```



예제

```
public class MyImageFrame extends JFrame implements ActionListener {

        private int pieces = 4;
        private int totalPieces = pieces * pieces;
        private int[] pieceNumber;
        private BufferedImage img;

        public MyImageFrame() {
                setTitle("Image Draw Test");
                try {
                        img = ImageIO.read(new File("hubble.jpg"));
                } catch (IOException e) {
                        System.out.println(e.getMessage());
                        System.exit(0);

                }
                pieceNumber = new int[totalPieces];
                for (int i = 0; i < totalPieces; i++) {
                        pieceNumber[i] = i;
                }
                add(new MyPanel(), BorderLayout.CENTER);
                JButton button = new JButton("DIVIDE");

                button.addActionListener(this);
                add(button, BorderLayout.SOUTH);
                setSize(img.getWidth(null), img.getHeight(null));
                setVisible(true);
        }
```

예제

```
        void divide() {
                Random rand = new Random();
                int ri;
                for (int i = 0; i < totalPieces; i++) {
                        ri = rand.nextInt(totalPieces);
                        int tmp = pieceNumber[i];
                        pieceNumber[i] = pieceNumber[ri];
                        pieceNumber[ri] = tmp;
                }
        }

        class MyPanel extends JPanel {
                public void paintComponent(Graphics g) {
                        super.paintComponent(g);
                        int pieceWidth = img.getWidth(null) / pieces;
                        int pieceHeight = img.getHeight(null) / pieces;
```

예제

```
            for (int x = 0; x < pieces; x++) {
                        int sx = x * pieceWidth;
                        for (int y = 0; y < pieces; y++) {
                                    int sy = y * pieceHeight;
                                    int number = pieceNumber[x * pieces + y];
                                    int dx = (number / pieces) * pieceWidth;
                                    int dy = (number % pieces) * pieceHeight;
            g.drawImage(img, dx, dy, dx + pieceWidth, dy + pieceHeight,
                        sx, sy, sx + pieceWidth, sy + pieceHeight, null);
                        }
            }
}
}
            public static void main(String[] args) {
                        new MyImageFrame();
            }
            public void actionPerformed(ActionEvent e) {
                        divide();
                        repaint();
            }
}
```

영상처리

- 영상 처리(image processing)은 이미지를 읽어서 여러 가지 처리를 하는 학문 분야이다. 예를 들어서 화질이 나쁜 이미지의 화질을 향상시키는 것도 영상 처리의 일종이다.

예제

```
public class GrayScaleImage extends JFrame {

            BufferedImage image;
            int width;
            int height;

            public GrayScaleImage() {
                        try {
                                    File input = new File("Lenna.png");
                                    image = ImageIO.read(input);
                                    width = image.getWidth();
                                    height = image.getHeight();

                                    for (int r = 0; r < height; r++) {
                                                for (int c = 0; c < width; c++) {

                                                            Color color = new
Color(image.getRGB(r, c));
```

예제

```
                                                int red = (int) (color.getRed());
                                                int green = (int) (color.getGreen());
                                                int blue = (int) (color.getBlue());
                                                int avg = (red + green + blue) / 3;
                                                Color newColor = new Color(avg,
avg, avg);

                                                image.setRGB(r, c,
newColor.getRGB());
                                    }
                        }

                        File ouptut = new File("output.png");
                        ImageIO.write(image, "png", ouptut);
                        add(new MyPanel());
                        pack();
                        setVisible(true);

            } catch (Exception e) {
                        System.out.println("이미지 읽기 실패!");
            }
}
```

예제

```
        class MyPanel extends JPanel {

                public void paintComponent(Graphics g) {
                        g.drawImage(image, 0, 0, null);
                }

                public Dimension getPreferredSize() {
                        if (image == null) {
                                return new Dimension(100, 100);
                        } else {
                                return new Dimension(image.getWidth(null),
                                                image.getHeight(null));
                        }
                }
        }

        static public void main(String args[]) throws Exception {
                GrayScaleImage obj = new GrayScaleImage();
        }
}
```

Java 2D

- 광범위한 그래픽 객체를 그릴 수 있다.
- 도형의 내부를 그라디언트(gradient)나 무늬로 채울 수 있다.
- 이미지를 그릴 수 있고 필터링 연산을 적용할 수 있다.
- 그래픽 객체들의 출력을 가지할 수 있는 메커니즘을 제공한

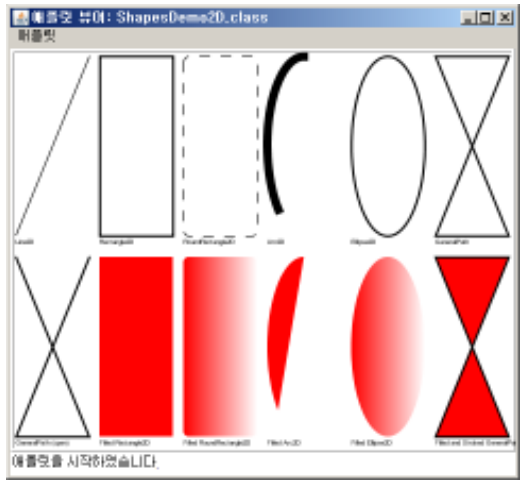
Using 2D Graphics API to display complex charts

Using image-filtering operations

Java 2D를 이용한 그리기

```
public void paintComponent(Graphics g)
{
    Graphics2D    g2 = (Graphics2D) g;
    g2.drawLine(100, 100, 300, 300);
    g2.drawRect(10, 10, 100, 100);
    ...
}
```

Java 2D를 이용한 그리기



사각형 그리기

- Shape r1 = **new** Rectangle2D.Float(10, 10, 50, 60);
- g2.draw(r1);

(a) Rectangle2D (b) RoundRectangle2D

타원 그리기

- // 타원 객체를 생성하고 타원을 그린다.
- g2.draw(new Ellipse2D.Double(x, y, rectwidth, rectheight));

원호생성

- Shape arc1 = **new** Arc2D.Float(10, 10, 90, 90, 90, 60, Arc2D.OPEN);

(a) Arc2D.OPEN (b) Arc2D.CHORD (c) Arc2D.PIE

예제

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

public class MoreShapes extends JFrame {

        public MoreShapes() {
                setSize(600, 130);
                setTitle("Java 2D Shapes");
                setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                JPanel panel = new MyPanel();
                add(panel);
                setVisible(true);
        }

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

예제

```
class MyPanel extends JPanel {
    ArrayList<Shape> shapeArray = new ArrayList<Shape>();

    public MyPanel() {
        Shape s;

        // 사각형
        s = new Rectangle2D.Float(10, 10, 70, 80);
        shapeArray.add(s);

        // 둥근 사각형
        s = new RoundRectangle2D.Float(110, 10, 70, 80, 20, 20);
        shapeArray.add(s);

        // 타원
        s = new Ellipse2D.Float(210, 10, 80, 80);
        shapeArray.add(s);

        // 원호: Arc2D.OPEN
        s = new Arc2D.Float(310, 10, 80, 80, 90, 90, Arc2D.OPEN);
        shapeArray.add(s);
```

예제

```
        // 원호 Arc2D.CHORD
        s = new Arc2D.Float(410, 10, 80, 80, 0, 180, Arc2D.CHORD);
        shapeArray.add(s);

        // 원호 Arc2D.PIE
        s = new Arc2D.Float(510, 10, 80, 80, 45, 90, Arc2D.PIE);
        shapeArray.add(s);
    }

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

        // 앤티 에일리어싱을 설정한다.
        g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
                RenderingHints.VALUE_ANTIALIAS_ON);

        g2.setColor(Color.BLACK);
        g2.setStroke(new BasicStroke(3));
        for (Shape s : shapeArray)
            g2.draw(s);
    }
}
```

예제

도형 채우기

- 단일색으로 채우기
  - g2.setColor(Color.*BLUE*);
  - g2.fill(ellipse);
- 투명하게 채우기
  - g2.setComposite(AlphaComposite.getInstance(AlphaComposite .SRC_OVER, 0.50F));
- 그라디언트로 채우기
  - GradientPaint gp = new GradientPaint(0, 0, Color.WHITE, 0, 100, Color.RED);

예제

```
class MyComponent extends JComponent {
        public void paint(Graphics g) {
                Graphics2D g2 = (Graphics2D) g;
                g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
                        RenderingHints.VALUE_ANTIALIAS_ON);
                g2.setStroke(new BasicStroke(3));
                GradientPaint gp = new GradientPaint(0, 10, Color.WHITE, 0,
                        70, Color.RED);
                // 사각형
                g2.setPaint(Color.RED);
                g2.fill(new Rectangle2D.Float(10, 10, 70, 80));          // 둥근 사각형
                g2.setPaint(gp);
                g2.fill(new RoundRectangle2D.Float(110, 10, 70, 80, 20, 20));
...
        }
}
```

실행 결과

LAB: 간단한 애니메이션

예제

```
class MyPanel extends JPanel implements ActionListener {

        private final int WIDTH = 500;
        private final int HEIGHT = 300;
        private final int START_X = 0;
        private final int START_Y = 250;
        private BufferedImage image;
        private Timer timer;
        private int x, y;

        public MyPanel() {
                setBackground(Color.BLACK);
                setPreferredSize(new Dimension(WIDTH, HEIGHT));
                setDoubleBuffered(true);

                File input = new File("ship.jpg");
                try {
                        image = ImageIO.read(input);
                } catch (IOException e) {
                        e.printStackTrace();
                }
```

예제

```
                x = START_X;
                y = START_Y;
                timer = new Timer(20, this);
                timer.start();
        }
        @Override
        public void paintComponent(Graphics g) {
                super.paintComponent(g);
                g.drawImage(image, x, y, this);
        }
        @Override
        public void actionPerformed(ActionEvent e) {
                x += 1;
                y -= 1;
                if (x > WIDTH) {
                        x = START_X;
                        y = START_Y;
                }
                repaint();
        }
}
```

예제

```
public class MyFrame extends JFrame {

        public MyFrame() {
                add(new MyPanel());
                setTitle("애니메이션 테스트");
                setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                setSize(500, 300);
                setVisible(true);
        }

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

Q & A