# **Java 실습**

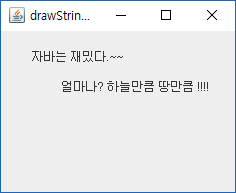
## **주제** : **그래픽 처리**

예제)

1. JPanel을 상속받아 paintComponent()를 오버라이딩하고 drawString() 메소드를사용하여 패널 내의(30, 30)과 (60, 60)에 각각 문자열을 출력하는 프로그램을 살펴본다.

|  |
| --- |
| import javax.swing.\*;  import java.awt.\*;  public class GraphicsDrawStringEx extends JFrame {  Container contentPane;  GraphicsDrawStringEx() {  setTitle("drawString 사용 예제");  setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  contentPane = getContentPane();  MyPanel panel = new MyPanel();  contentPane.add(panel , BorderLayout.CENTER);  setSize(250,200);  setVisible(true);  }  class MyPanel extends JPanel {  public void paintComponent(Graphics g) {  super.paintComponent(g);  g.drawString("자바는 재밌다.~~", 30,30);  g.drawString("얼마나? 하늘만큼 땅만큼 !!!!", 60, 60);  }  }  public static void main(String [] args) {  new GraphicsDrawStringEx();  }  } |

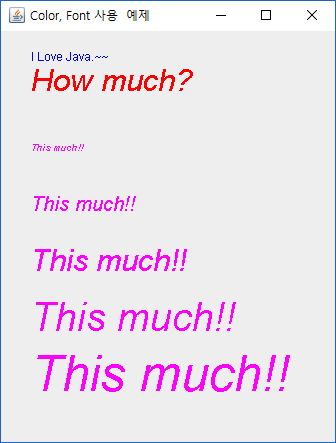
결과 :



1. Color와 Font 클래스를 이용하여 문자를 출력시키는 프로그램을 살펴본다.

|  |
| --- |
| import javax.swing.\*;  import java.awt.\*;  public class GraphicsColorFontEx extends JFrame {  Container contentPane;  GraphicsColorFontEx() {  setTitle("Color, Font 사용 예제");  setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  contentPane = getContentPane();  MyPanel panel = new MyPanel();  contentPane.add(panel , BorderLayout.CENTER);  setSize(350, 450);  setVisible(true);  }  class MyPanel extends JPanel {  public void paintComponent(Graphics g) {  super.paintComponent(g);  g.setColor(Color.BLUE);  g.drawString("I Love Java.~~", 30,30);  g.setColor(new Color(255, 0, 0));  g.setFont(new Font("Arial", Font.ITALIC, 30));  g.drawString("How much?", 30, 60);  g.setColor(new Color(0x00ff00ff));  for(int i=1; i<=5; i++) {  g.setFont(new Font("Dialog", Font.ITALIC, i\*10));  g.drawString("This much!!", 30, 60+i\*60);  }  }  }  public static void main(String [] args) {  new GraphicsColorFontEx();  }  } |

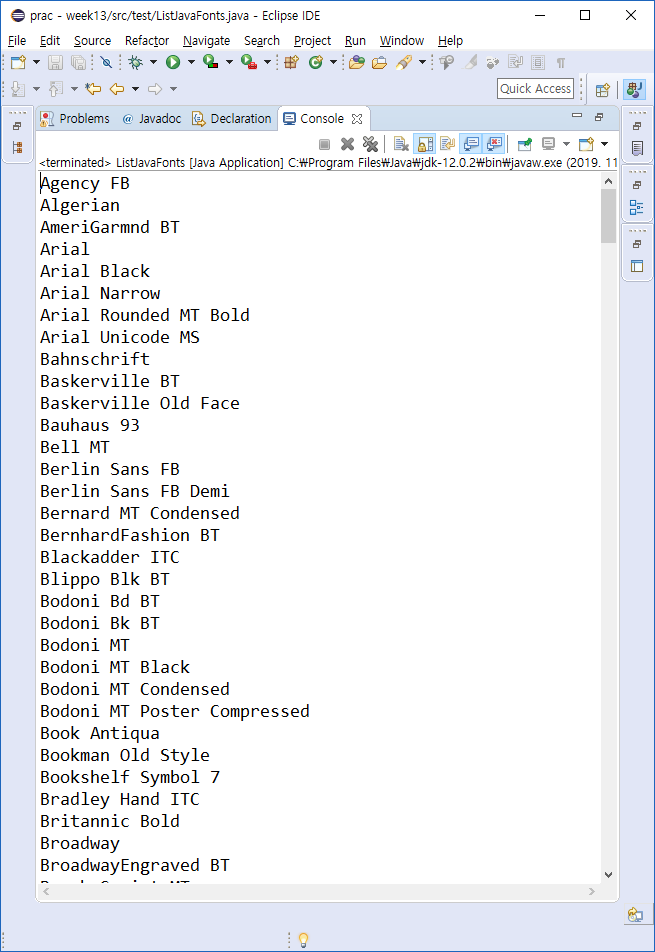
결과 :



1. 사용가능 폰트 종류 알아보기 : getAvailableFontFamilyNames() 를 이용한다.

|  |
| --- |
| import java.awt.GraphicsEnvironment;  public class ListJavaFonts  {  public static void main(String[] args)  {  String fonts[] =  GraphicsEnvironment.getLocalGraphicsEnvironment().getAvailableFontFamilyNames();  for ( int i = 0; i < fonts.length; i++ )  {  System.out.println(fonts[i]);  }  }  } |

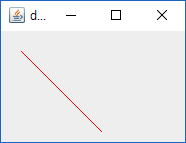
결과 :



1. Graphics 객체를 이용한 여러종류의 도형을 살펴본다.
2. 선 그리기

|  |
| --- |
| import javax.swing.\*;  import java.awt.\*;  public class GraphicsDrawLineEx extends JFrame {  Container contentPane;  GraphicsDrawLineEx() {  setTitle("drawLine 사용 예제");  setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  contentPane = getContentPane();  MyPanel panel = new MyPanel();  contentPane.add(panel, BorderLayout.CENTER);  setSize(200, 150);  setVisible(true);  }  class MyPanel extends JPanel {  public void paintComponent(Graphics g) {  super.paintComponent(g);  g.setColor(Color.RED);  g.drawLine(20,20, 100, 100);  }  }  public static void main(String [] args) {  new GraphicsDrawLineEx();  }  } |

결과:



1. 여러종류 도형 그리기

|  |
| --- |
| package week13;  import javax.swing.\*;  import java.awt.\*;  public class GraphicsDrawLineEx extends JFrame {  Container contentPane;  GraphicsDrawLineEx() {  setTitle("drawLine 사용 예제");  setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  contentPane = getContentPane();  MyPanel panel = new MyPanel();  contentPane.add(panel, BorderLayout.CENTER);  setSize(300,600);  setVisible(true);  }  class MyPanel extends JPanel {  public void paintComponent(Graphics g) {  super.paintComponent(g);  g.setColor(Color.RED);  g.drawLine(20,20, 100, 100);  g.drawOval(20,150,80,80);  g.drawRect(20,270,80,80);  g.drawArc(20,400,80,80,90,270);    }  }  public static void main(String [] args) {  new GraphicsDrawLineEx();  }  } |

결과 :

|  |  |
| --- | --- |
|  | 선 굵기 조정 :  Graphics2D g2 = (Graphics2D) g;  g2.setStroke(new BasicStroke(10)); |

1. 이미지를 출력하는 프로그램을 살펴본다.

|  |
| --- |
| import javax.swing.\*;  import java.awt.\*;  public class GraphicsDrawImageEx1 extends JFrame {  Container contentPane;  GraphicsDrawImageEx1() {  setTitle("drawImage 사용 예제 1");  setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  contentPane = getContentPane();  MyPanel panel = new MyPanel();  contentPane.add(panel, BorderLayout.CENTER);  setSize(300, 400);  setVisible(true);  }  class MyPanel extends JPanel {  ImageIcon imageIcon = new ImageIcon("images/image0.jpg");  Image image = imageIcon.getImage();  public void paintComponent(Graphics g) {  super.paintComponent(g);  g.drawImage(image, 20,20, this);  }  }  public static void main(String [] args) {  new GraphicsDrawImageEx1();  }  } |

결과 :



참고 : “image/image0.jpg”이미지를 출력

1. 이미지 loading :

|  |
| --- |
| ImageIcon icon = new ImageIcon(“image/image0.jpg”);  **Image img = icon.getImage();** |

1. 원본 이미지를 (20,20) 위치에 원본 크기로 그리기 :

|  |
| --- |
| public void paintComponent(Graphics g) {  super.paintComponent(g);  g.drawImage(img, 20, 20, this);  } |

1. 원본 이미지를 100x100 크기로 조절하여 그리기

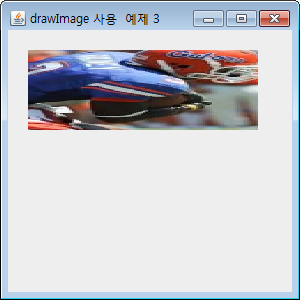
|  |
| --- |
| public void paintComponent(Graphics g) {  super.paintComponent(g);  g.drawImage(img, 20, 20, 100, 100, this);  } |

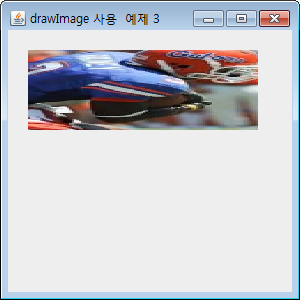
1. 원본 이미지를 패널에 꽉 차도록 그리기 : JPanel의 크기가 변할 때마다 이미지의 크기도 따라서 변함

|  |
| --- |
| public void paintComponent(Graphics g) {  super.paintComponent(g);  g.drawImage(img, 0, 0, getWidth(), getHeight(), this);  } |

1. 원본 이미지의 (50, 0)에서 (150,150) 사각형 부분을 JPanel의 (20,20)에서 (250,100) 영역에 그리기

|  |
| --- |
| public void paintComponent(Graphics g) {  super.paintComponent(g);  g.drawImage(img, 20,20,250,100,50,0,150,150, this);  } |

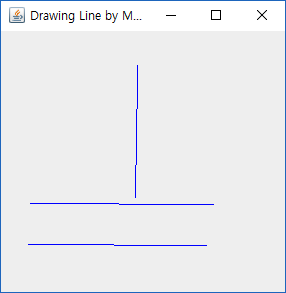




1. 마우스를 이용한 선을 그려본다.(repaint() 사용)

|  |
| --- |
| import javax.swing.\*;  import java.awt.\*;  import java.util.\*;  import java.awt.event.\*;  public class GraphicsDrawLineMouseEx extends JFrame {  Container contentPane;  GraphicsDrawLineMouseEx() {  setTitle("Drawing Line by Mouse 예제");  setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  contentPane = getContentPane();  MyPanel panel = new MyPanel();  contentPane.add(panel, BorderLayout.CENTER);  setSize(300, 300);  setVisible(true);  }  public static void main(String [] args) {  new GraphicsDrawLineMouseEx();  }  class MyPanel extends JPanel {  Vector<Point> vs = new Vector<Point>();  Vector<Point> ve = new Vector<Point>();  Point startP = null;  Point endP = null;  public MyPanel() {  addMouseListener(new MouseAdapter(){  public void mousePressed(MouseEvent e) {  startP = e.getPoint();  }  public void mouseReleased(MouseEvent e) {  endP = e.getPoint();  vs.add(startP);  ve.add(endP);  repaint();  }  });  }  public void paintComponent(Graphics g) {  super.paintComponent(g);  g.setColor(Color.BLUE);  for(int i=0; i<vs.size(); i++) {  Point s = vs.elementAt(i);  Point e = ve.elementAt(i);  g.drawLine((int)s.getX(), (int)s.getY(),  (int)e.getX(), (int)e.getY());  }  }  }  } |

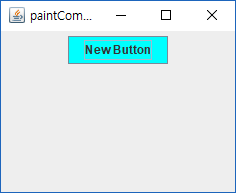
결과 :



8-1 : JButton을 상속받아 새로운 버튼 생성을 한다.

|  |
| --- |
| import javax.swing.\*;  import java.awt.\*;  public class paintComponentEx extends JFrame {  Container contentPane;  paintComponentEx() {  setTitle("paintComponent 사용 예제");  setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  contentPane = getContentPane();  contentPane.setLayout(new FlowLayout());  MyButton b = new MyButton("New Button");  b.setOpaque(true);  b.setBackground(Color.CYAN);  contentPane.add(b);  setSize(250,200);  setVisible(true);  }  class MyButton extends JButton {  MyButton(String s) {  super(s);  }  }  public static void main(String [] args) {  new paintComponentEx();  }  } |

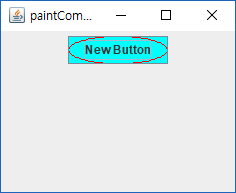
결과 :



8-2 : JButton을 상속받아 새로운 버튼 생성을 한다. 이때 paintComponent()메소드를 추가해본다.

|  |
| --- |
| class MyButton extends JButton {  MyButton(String s) {  super(s);  }    public void paintComponent(Graphics g) {  super.paintComponent(g);  g.setColor(Color.RED);  g.drawOval(0,0,this.getWidth()-1, this.getHeight()-1);  }    } |

결과 :



연습문제)

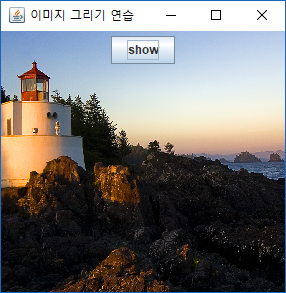
1. 다음과 같이 출력하는 프로그램을 작성하시오.

|  |  |
| --- | --- |
|  |  |
|  |  |

선의 굵기를 조정하는 방법 : Graphics2D 객체를 이용한다.

|  |
| --- |
| Graphics2D g2 = (Graphics2D) g;  g2.setStroke(new BasicStroke(5)); |

1. 이미지와 컴포넌트를 동시에 화면에 출력하는 프로그램을 작성해보시오.
   1. 바탕화면에 images/back.jpg가 그려지고 그 위에 show 버튼을 보이는 프로그램을 작성하시오.



* 1. 이번에는 버튼을 클릭할 때 마다 한번 누르면 배경이 나타나고 다시 누르면 보이지 않도록 이벤트 처리를 해보시오.

|  |  |
| --- | --- |
| 버튼을 누른다 | 다시 버튼을 누른다 |

* 1. 이번에는 배경화면 위에서 반지름이 20픽셀인 원 이미지를 그리고 마우스를 드래깅하면 원이 이동하는 프로그램을 작성하시오.

|  |  |
| --- | --- |
|  |  |

1. Images/apple.jpg 이미지를 그리고 +키를 누르면 10%확대하고 – 키를 누르면 10%축소하는 프로그램을 작성하시오.

|  |  |
| --- | --- |
| 원본 | 확대 |

1. 예제 5번을 활용해 이미지를 4등분하고 각각 10 픽셀씩 사이를 띄고 출력하는 프로그램을 작성하시오.

