



Power Java

제23장 그래픽 프로그래밍



© 2009 인피니티북스 All rights reserved



이번 장에서 학습할 내용

- 자바에서의 그래픽
- 기초 사항
- 기초 도형 그리기
- 색상
- 폰트
- Java 2D
- Java 2D를 이용한 그리기
- Java 2D를 이용한 채우기
- 도형 회전과 평행 이동

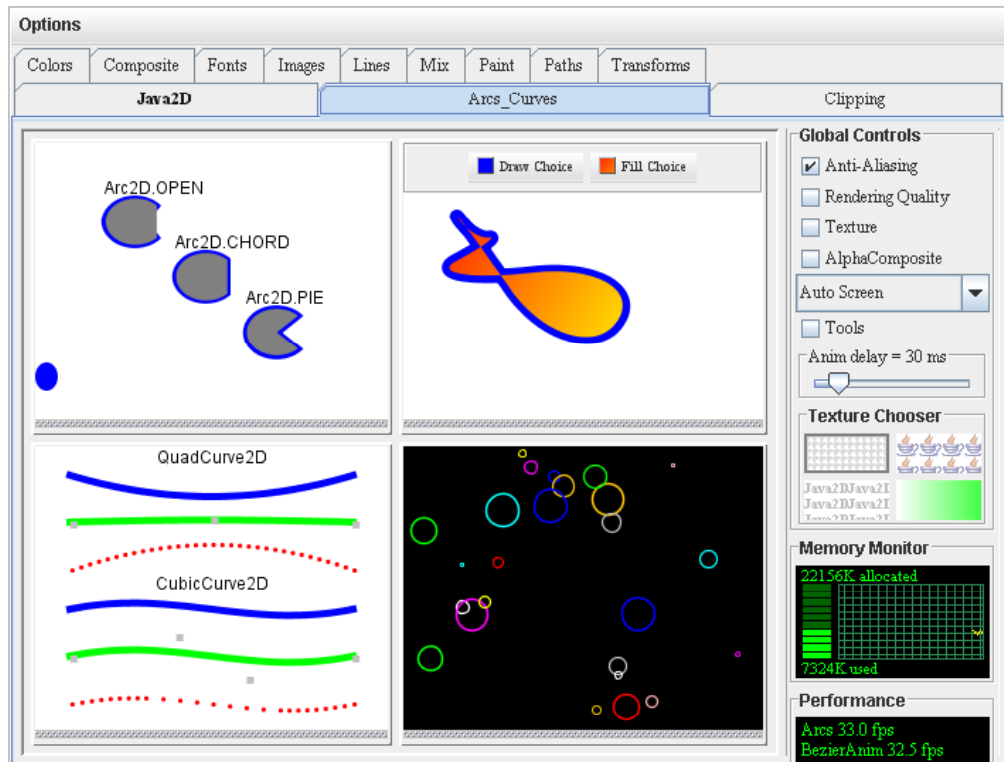
자바를
이용하여서
화면에 그림을
그려봅시다.



© 2009 인피니티북스 All rights reserved



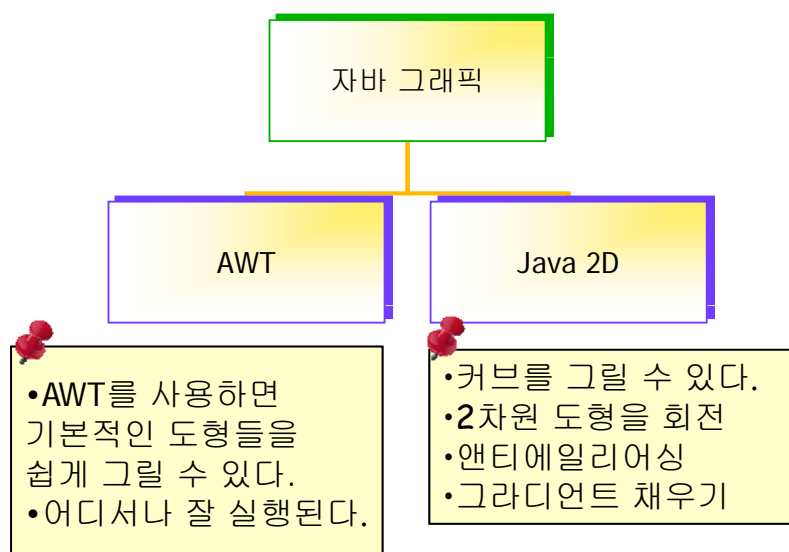
자바 그래픽 데모



© 2009 인피니티박스 All rights reserved



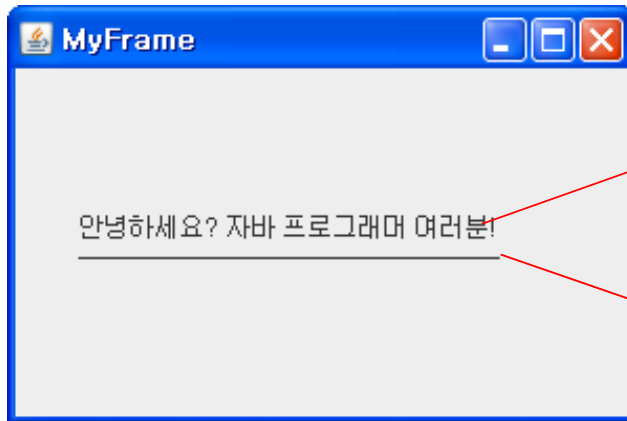
자바 그래픽의 두가지 방법



© 2009 인피니티박스 All rights reserved



간단한 예제



문자열 출력

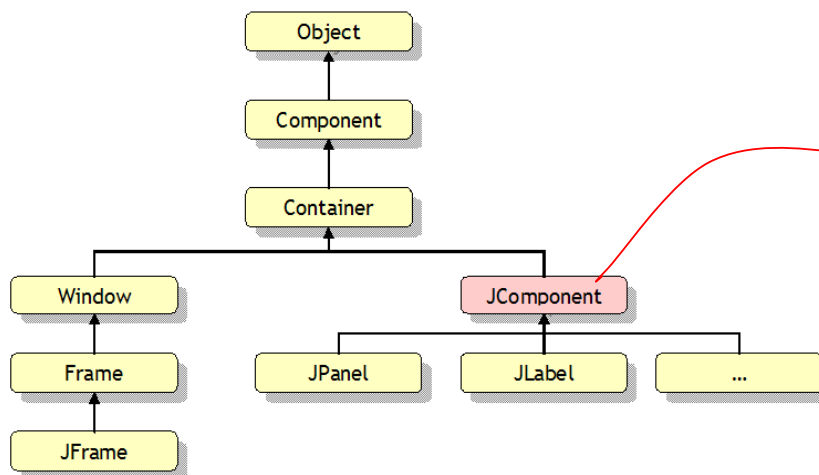
직선

© 2009 인피니티박스 All rights reserved



어디에 그릴 것인가?

- JFrame이나 JPanel에도 그릴 수 있지만 우리 나름대로의 컴포넌트를 정의하여 보자.



이것을 상속받아서
우리 나름대로의
컴포넌트를 정의한다.

그림 23-2 JComponent 클래스

© 2009 인피니티박스 All rights reserved



어떻게 그릴 것인가?

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

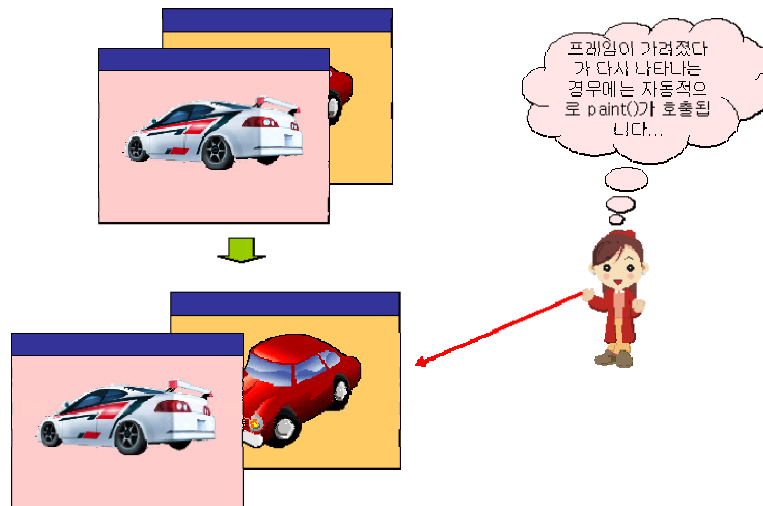


그림 23-3 `paint()` 메소드

© 2009 인피니티북스 All rights reserved



일반적인 코드의 형태

```
class MyComponent extends JComponent
{
    public void paint(Graphics g)
    {
        // 여기에 그림을 그리는 코드를 넣는다.
    }
}
```

© 2009 인피니티북스 All rights reserved



구체적인 코드

```
class MyComponent extends JComponent
{
    public static final int x = 30;
    public static final int y = 80;

    public void paint(Graphics g) {
        g.drawString("안녕하세요? 자바 프로그래머 여러분!", x, y);
        g.drawLine(x, y + 10, x + 200, y + 10);
    }
}
```

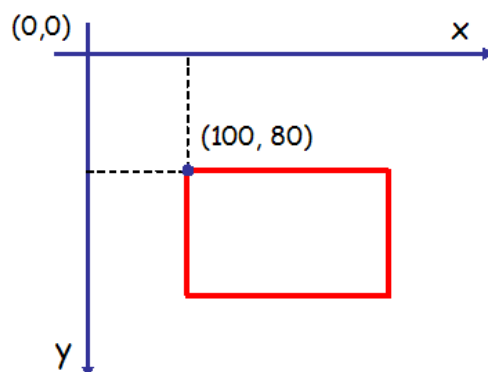
문자열 출력

직선 출력

© 2009 인피니티박스 All rights reserved



그래픽 좌표계



y좌표는 아래로 갈수록 증가합니다.



그림 17-3 자바의 좌표계

© 2009 인피니티박스 All rights reserved



예제

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

// 그림이 그려지는 컴포넌트를 정의
class MyComponent extends JComponent {

    public static final int x = 30;

    public static final int y = 80;

    public void paint(Graphics g) {
        g.drawString("안녕하세요? 자바 프로그래머 여러분!", x, y);
        g.drawLine(x, y + 10, x + 200, y + 10);
    }
}
```



예제

```
// 프레임 컴포넌트를 상속받아서 정의
public class MyFrame extends JFrame {
    public static final int WIDTH = 300;
    public static final int HEIGHT = 200;

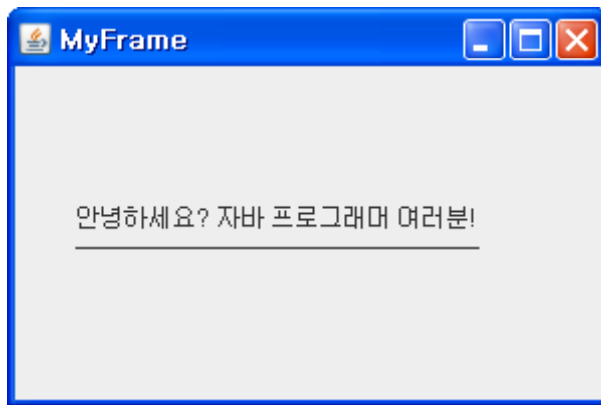
    public MyFrame() {
        setTitle("MyFrame");
        setSize(WIDTH, HEIGHT);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setVisible(true);

        // MyComponent 객체 생성하여 프레임에 추가
        MyComponent c = new MyComponent();
        add(c);
    }

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



실행 결과



© 2009 인피니티박스 All rights reserved



기초 도형 그리기

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

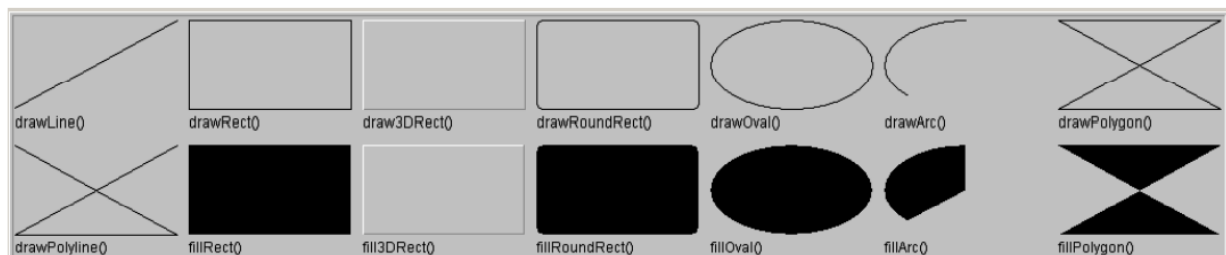


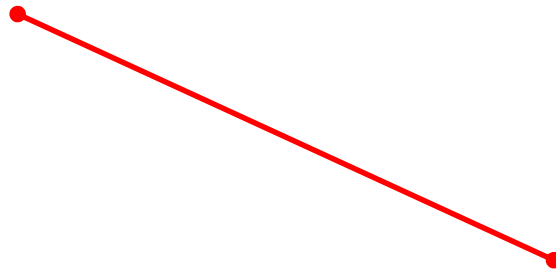
그림 23-7 그리기 메소드

© 2009 인피니티박스 All rights reserved



직선 그리기

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



© 2009 인피니티박스 All rights reserved



사각형 그리기

메소드 및 설명

`drawRect(int x, int y, int width, int height)` // 왼쪽 상단 좌표 (x, y)

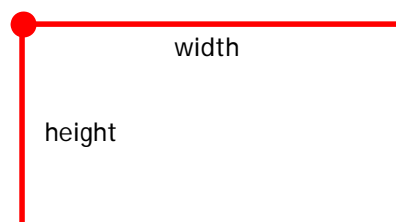
`fillRect(int x, int y, int width, int height)` // 채워진 사각형

`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)`



© 2009 인피니티박스 All rights reserved



drawRoundRect()

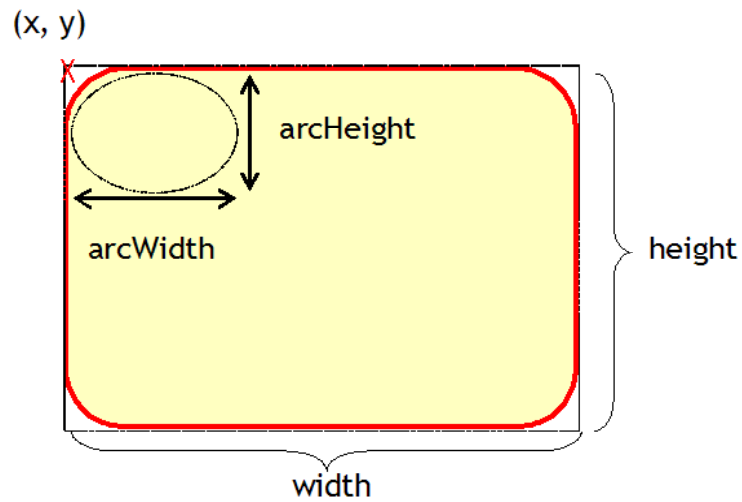


그림 23-6 drawRoundRect()의 매개 변수의 의미

© 2009 인피니티박스 All rights reserved



타원 그리기

메 소 드	설 명
<code>drawOval(int x, int y, int width, int height)</code>	좌측 상단의 좌표기 <code>x,y</code> 이며 폭 <code>width</code> , 높이 <code>height</code> 의 사각형 안에 내접하는 타원을 그린다.
<code>fillOval(int x, int y, int width, int height)</code>	채워진 타원을 그린다.

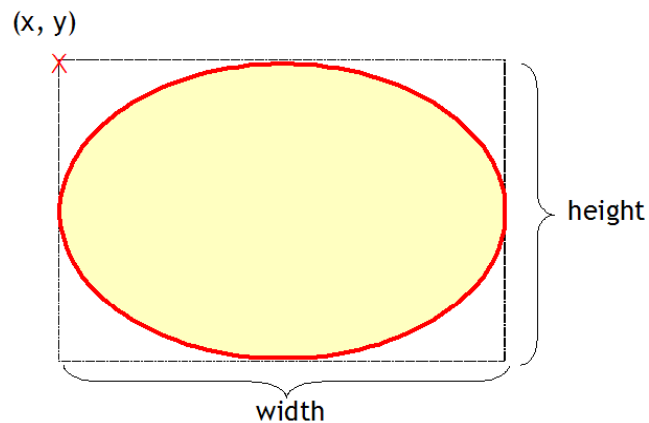


그림 23-6 drawOval() 매개 변수의 의미

© 2009 인피니티박스 All rights reserved



호그리기

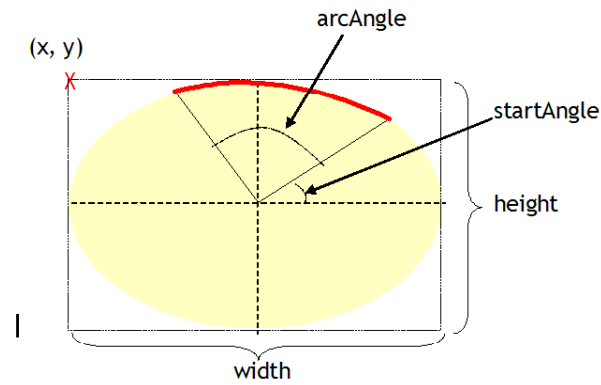


그림 23-6 `drawOval()` 매개 변수의 의미

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

© 2009 인피니티박스 All rights reserved



예 제

- 아래 그림과 비슷한 얼굴을 그려보자



© 2009 인피니티박스 All rights reserved



예제

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

class MyComponent extends JComponent {

    public void paint(Graphics 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);
    }
}
```

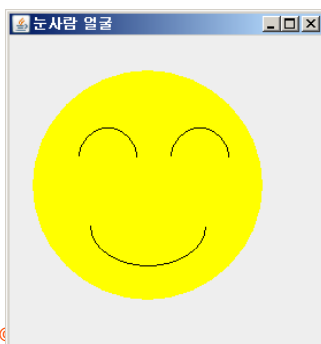
© 2009 인피니티박스 All rights reserved



예제

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

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





색상

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

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

© 2009 인피니티박스 All rights reserved



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 paint(Graphics g) {
        g.setColor(color);
        g.fillRect(10, 10, 210, 220);
    }
}
```

© 2009 인피니티박스 All rights reserved



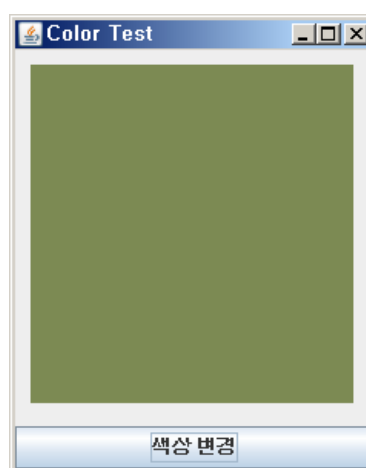
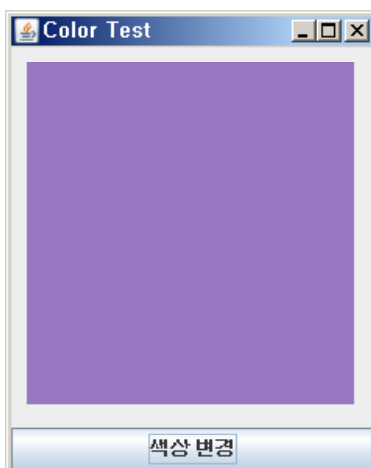
예제

```
public void actionPerformed(ActionEvent e) {  
    color = new Color((int) (Math.random()*255.0),  
                      (int) (Math.random()*255.0), (int) (Math.random()*255.0));  
    repaint();  
}  
}  
  
public class ColorTest extends JFrame {  
    public ColorTest() {  
        setSize(240, 300);  
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        setTitle("Color Test");  
        setVisible(true);  
        JPanel panel = new JPanel();  
        add(panel);  
    }  
  
    public static void main(String[] args) {  
        ColorTest s = new ColorTest();  
    }  
}
```

© 2009 인피니티박스 All rights reserved



실행 결과



© 2009 인피니티박스 All rights reserved



색상 선택기



© 2009 인피니티박스 All rights reserved



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"));
    }
}

```

© 2009 인피니티박스 All rights reserved



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();
}
}
```

© 2009 인피니티박스 All rights reserved



폰트

- 폰트를 지정하기 위해서는 Font 클래스를 사용
- Font 객체는 폰트 이름(Courier, Helvetica,..)과 스타일(plain, bold, italic,...), 크기(12포인트,...)의 3가지 속성
- Font font = new Font("Courier", Font.PLAIN, 10); // plain 형식이고 크기는 10포인트

© 2009 인피니티박스 All rights reserved

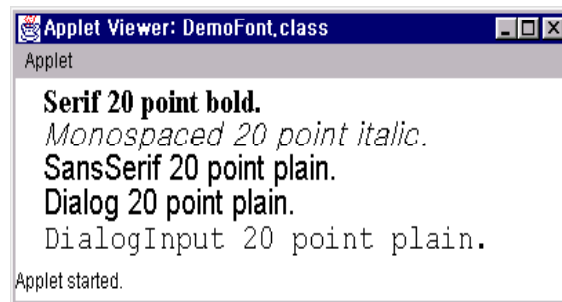


폰트의 종류

- 자바가 제공하는 논리적인 폰트 중에서 하나를 사용한다.
- 사용가능한 폰트의 리스트에서 사용자가 선택하도록 한다.

논리적인 폰트는 대표적인 폰트를 나타낸다. 다음은 논리적인 폰트 이름이다.

논리적인 폰트	설 명
"Serif"	삐침 (serif)을 갖는 가변폭 글꼴, 대표적으로 TimesRoman이 있다.
"SansSerif"	삐침 (serif)을 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



© 2009 인피니티박스 All rights reserved



예제

```
import javax.swing.*;
import java.awt.event.*;
import java.awt.*;
class MyComponent extends JComponent {
    Font f1, f2, f3, f4, f5;
    public MyComponent() {
        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 paint(Graphics 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 JPanel();
        add(panel);
    }

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

```



© 2009 인피니티박스 All rights reserved



Java 2D

- 광범위한 그래픽 객체를 그릴 수 있다.
- 도형의 내부를 그라디언트(gradient)나 무늬로 채울 수 있다.
- 문자열을 출력할 때 폰트와 렌더링 과정을 세밀하게 조정할 수 있다.
- 이미지를 그릴 수 있고 필터링 연산을 적용할 수 있다.
- 그래픽 객체들의 충돌을 감지할 수 있는 메커니즘을 제공한다.
- 렌더링 중간에 객체들을 조합하거나 변형할 수 있다.
- 화면과 프린터에 같은 방법으로 그릴 수 있다.



Using 2D Graphics API to display complex charts



Using image-filtering operations

그림 17-3 Java 2D를 이용한 그래픽의 예(출처: java.sun.com)



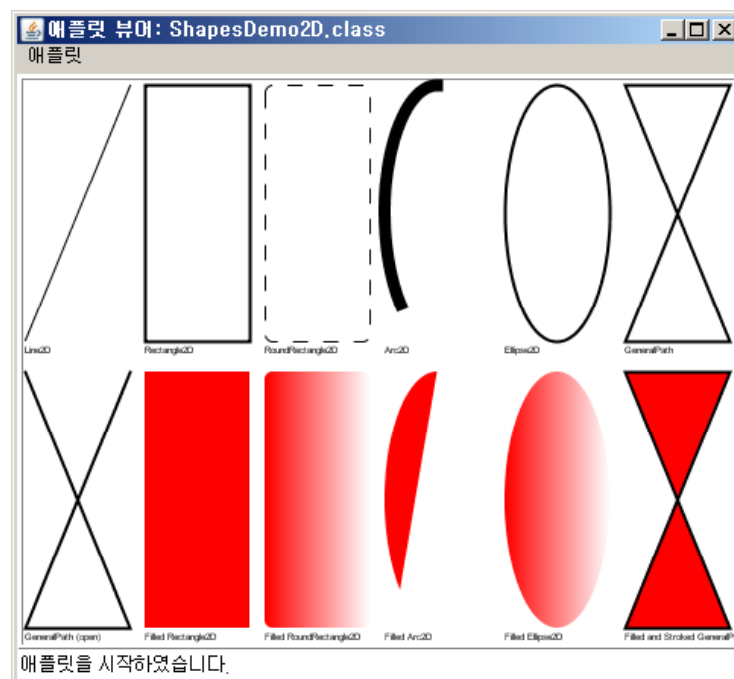
Java 2D를 이용한 그리기

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

© 2009 인피니티박스 All rights reserved



Java 2D를 이용한 그리기



© 2009 인피니티박스 All rights reserved



사각형 그리기

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



(a) Rectangle2D (b) RoundRectangle2D

그림 23-1 사각형의 종류



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 JPanel();
        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);
    }
}
```

© 2009 인피니티박스 All rights reserved



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 paint(Graphics g) {
    Graphics2D g2 = (Graphics2D) g;

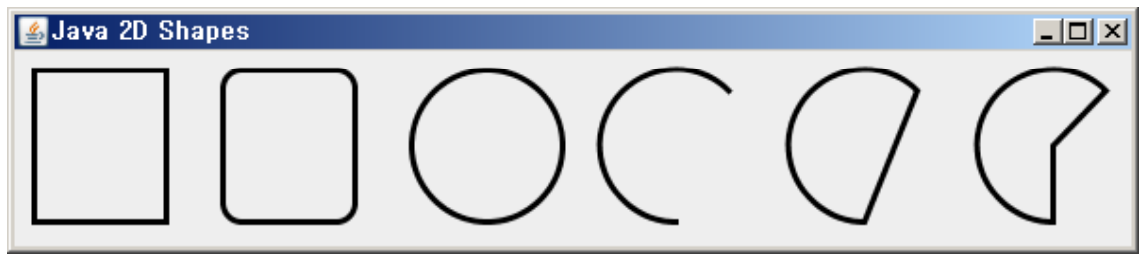
    // 앤티 에일리어싱을 설정한다.
    g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
        RenderingHints.VALUE_ANTIALIAS_ON);

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

© 2009 인피니티박스 All rights reserved



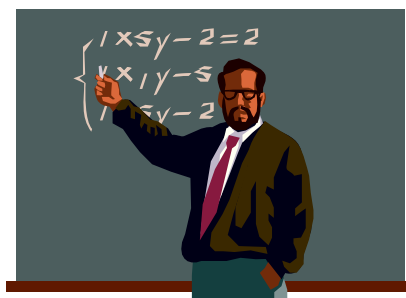
예제



© 2009 인피니티박스 All rights reserved



Q & A



© 2009 인피니티박스 All rights reserved