

그래픽 프로그래밍

예제 소스 코드는 파일과 연결되어 있습니다.

editplus(유료), notepad++(무료)와 같은 편집 도구를 미리 설치하여 PPT를 슬라이드 쇼로 진행할 때 소스 파일과 연결하여 보면 강의하실 때 편리합니다.

그래픽 컨텍스트

■ 그래픽 요소



- 그래픽 주체 : 그림을 그리도록 프로그래밍하는 개발자 혹은 JVM
- 그래픽 도구 : 펜, 붓, 팔레트, 폰트 등을 의미. 자바는 Graphics 클래스로 그리기, 칠하기, 이미지 출력하기, 클리핑 등 프로그래밍에 필요한 모든 필드와 메서드를 제공
- 그래픽 대상 : 그림을 그릴 수 있는 도화지 등을 의미한다. 자바에서는 AWT나 스윙의 모든 컴포넌트, 이미지가 그래픽 대상

그래픽 컨텍스트

■ 컴포넌트의 렌더링

- 그래픽을 지원하는 대부분의 프로그래밍에서 컴포넌트는 GUI 시스템이 자신의 모양을 렌더링
- 자바에서도 모든 스윙 컴포넌트는 JVM이 다음 메서드를 호출해 자신의 모양을 렌더링한다. 이 메서드는 javax.swing.JComponent 클래스가 제공

```
protected void paintComponent(Graphics g)
```

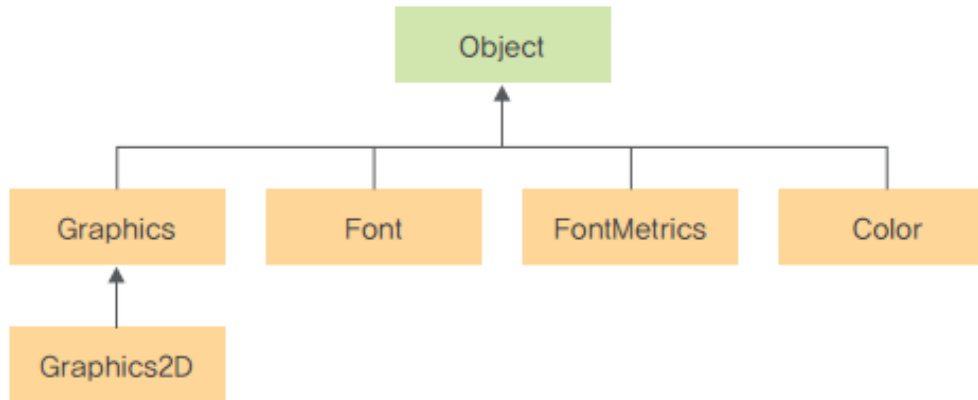
■ 커스텀 컴포넌트

```
class 커스텀컴포넌트 extends 스윙컴포넌트 {  
    ...  
  
    public void paintComponent(Graphics g) {  
        // 커스텀 컴포넌트에 필요한 코드  
    }  
}
```

그래픽 컨텍스트

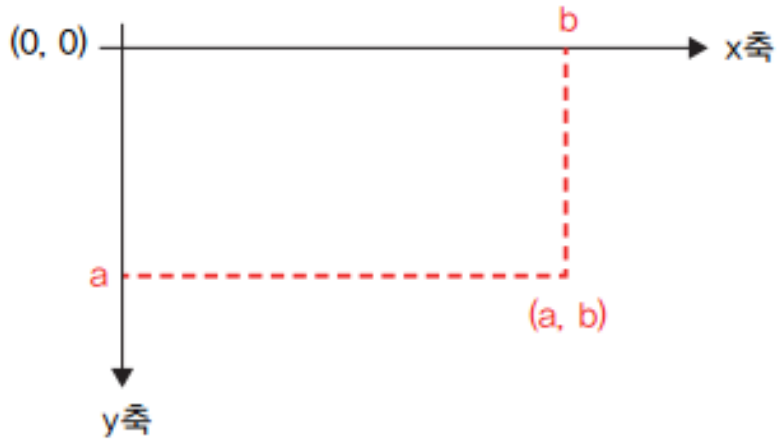
■ 그래픽 컨텍스트 의미

- 그래픽 객체를 그리는 데 필요한 정보
- Graphics 클래스는 자바에서 그리기 작업을 할 때 필요한 도구로 AWT뿐만 아니라 스윙에서도 사용. 자바에서는 색상 선택, 문자열 그리기, 도형 그리기, 도형 칠하기, 이미지 그리기, 클리핑 등을 지원하는 각종 메서드와 상수를 Graphics 클래스로 제공
- 그래픽 프로그래밍과 관련된 클래스의 계층구조



그래픽 컨텍스트

- Graphics 클래스가 사용하는 좌표 체계

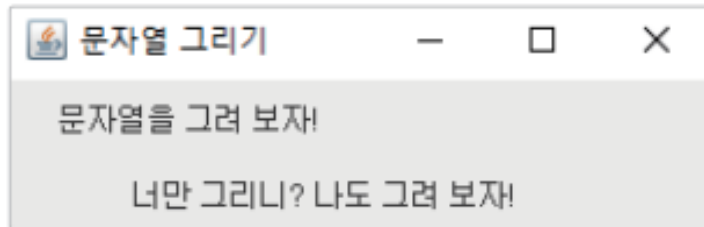


그래픽 그리기

■ 문자열 그리기

```
void drawString(String str, int x, int y)
```

- 예제 : [sec02/StringDemo](#)



그래픽 그리기

■ 색상 지정 : Color 클래스

● 생성자

```
Color(float r, float g, float b)
Color(float r, float g, float b, float a)
Color(int r, int g, int b)
Color(int r, int g, int b, int a)
Color(int rgb)
Color(int rgba, boolean hasalpha)
```

● Color 클래스가 제공하는 대표적인 상수

상수	RGB 값
static Color black(또는 BLACK)	(0, 0, 0)
static Color blue(또는 BLUE)	(0, 0, 255)
static Color darkgray(또는 DARK_GRAY)	(64, 64, 64)
static Color lightGray(또는 LIGHT_GRAY)	(192, 192, 192)
static Color green(또는 GREEN)	(0, 255, 0)
static Color red(또는 RED)	(255, 0, 0)
static Color white(또는 WHITE)	(255, 255, 255)

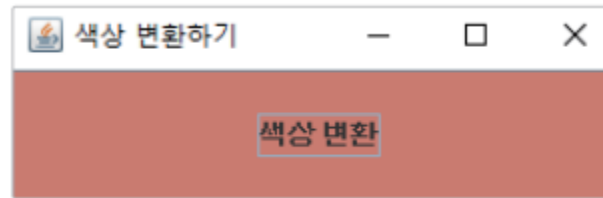
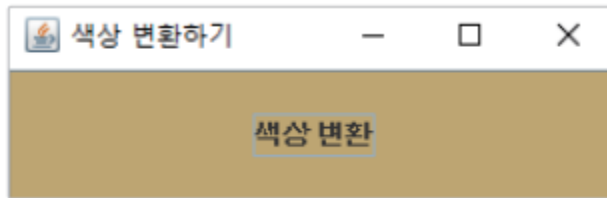
그래픽 그리기

■ 색상 지정 : Color 클래스

```
void setBackground(Color c)  
void setForeground(Color c)
```

```
void setColor(Color c)
```

- 예제 : [sec02/ColorDemo](#)



그래픽 그리기

■ 폰트 지정 : Font 클래스

● 생성자

// 지정한 폰트 이름, 폰트 스타일, 폰트 크기를 사용해 폰트 객체를 생성한다.

```
Font(String name, int style, int size)
```

● Font 설정

```
void setFont(Font font)
```

● Font 클래스가 제공하는 상수

상수		설명
폰트 이름	String DIALOG	대화상자에서 주로 사용하는 폰트이다.
	String MONOSPACED	고정 폭을 가지는 폰트이다.
	String SERIF	빠침이 있는 가변 폭의 폰트이다.
	String SANS_SERIF	빠침이 없는 가변 폭의 폰트이다.
폰트	int BOLD	굵은체이다.
스타일	int ITALIC	이탤릭체이다.
	int PLAIN	일반 폰트이다.

그래픽 그리기

■ 폰트 지정 : Font 클래스

- 예제 : [sec02/Font1Demo](#)



- 예제 : [sec02/Font2Demo](#)

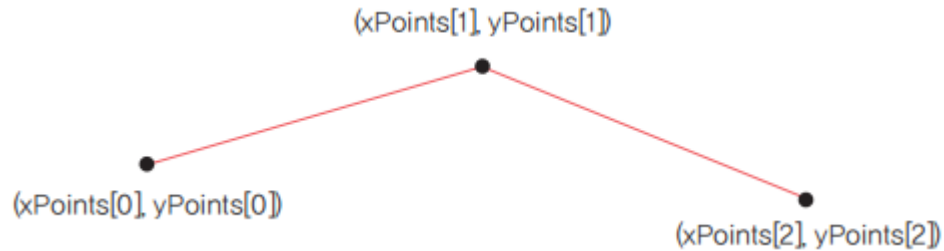


그래픽 그리기

■ 도형 그리기

● 직선

```
void drawLine(int x1, int y1, int x2, int y2)  
void drawPolyline(int[] xPoints, int[] yPoints, int nPoints)
```



● 예제 : [sec02/LineDemo](#)



그래픽 그리기

■ 도형 그리기

● 사각형

```
void drawRect(int x, int y, int width, int height)
```

```
void drawRoundRect(int x, int y, int width, int height, int arcWidth,  
                  int arcHeight)
```

```
void draw3DRect(int x, int y, int width, int height, boolean raised)
```

```
void fillRect(int x, int y, int width, int height)
```

```
void fillRoundRect(int x, int y, int width, int height, int arcWidth,  
                  int arcHeight)
```

```
void fill3DRect(int x, int y, int width, int height, boolean raised)
```



그래픽 그리기

■ 도형 그리기

● 사각형

- 예제 : [sec02/RectDemo](#)



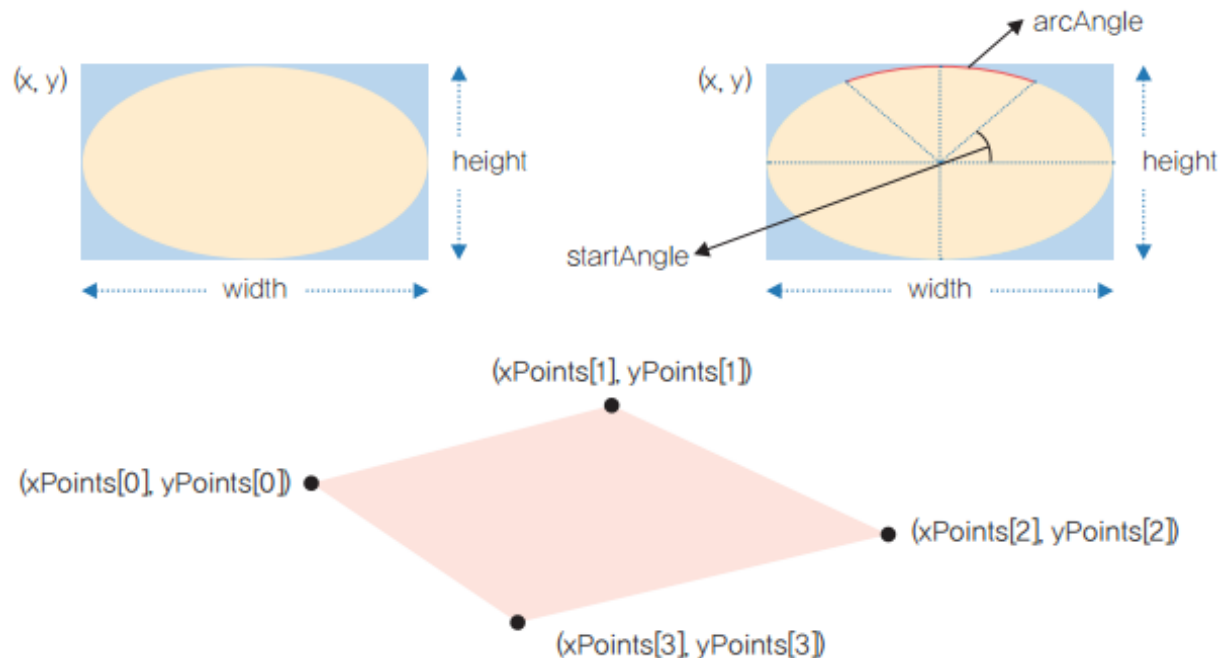
그래픽 그리기

■ 도형 그리기

- 타원, 호,
다각형

```
void drawOval(int x, int y, int width, int height)
void drawArc(int x, int y, int width, int height, int startAngle, int arcAngle)
void drawPolygon(int[] xPoints, int[] yPoints, int nPoints)

void fillOval(int x, int y, int width, int height)
void fillArc(int x, int y, int width, int height, int startAngle, int arcAngle)
void fillPolygon(int[] xPoints, int[] yPoints, int nPoints)
```



그래픽 그리기

■ 도형 그리기

- 타원, 호, 다각형
 - 예제 : [sec02/StarDemo](#)



그래픽 그리기

■ 이미지 그리기

● 주요 클래스

- java.awt.image 클래스: 그래픽 이미지를 사각형 픽셀 이미지로 표현.
- java.awt.image.BufferedImage 클래스 : Image의 자식 클래스로 애플리케이션이 직접 BufferedImage 객체를 생성할 수 있고, 이미지 데이터도 조작 가능

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

Image의 자식 클래스이다.

그래픽 그리기

■ 이미지 그리기

- Graphics 클래스가 제공하는 이미지 그리기 메서드

```
// 원본 이미지와 동일한 크기로 그리기
boolean drawImage(Image img, int x, int y, Color bgcolor,
                  ImageObserver observer)
boolean drawImage(Image img, int x, int y, ImageObserver observer)

// 원본 이미지 크기를 조절해 그리기
boolean drawImage(Image img, int x, int y, int width, int height,
                  Color bgcolor, ImageObserver observer)
boolean drawImage(Image img, int x, int y, int width, int height,
                  ImageObserver observer)
```

그래픽 그리기

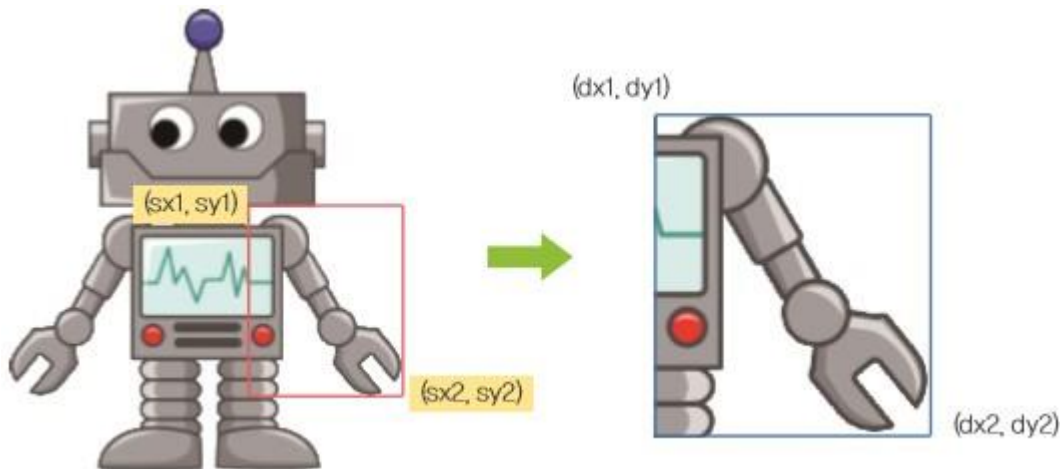
■ 이미지 그리기

- Graphics 클래스가 제공하는 이미지 그리기 메서드 --- 계속

// 원본 이미지 일부를 크기 조절해 그리기

```
boolean drawImage(Image img, int dx1, int dy1, int dx2, int dy2, int sx1,  
                  int sy1, int sx2, int sy2, Color bgcolor,  
                  ImageObserver observer)
```

```
boolean drawImage(Image img, int dx1, int dy1, int dx2, int dy2, int sx1,  
                  int sy1, int sx2, int sy2, ImageObserver observer)
```



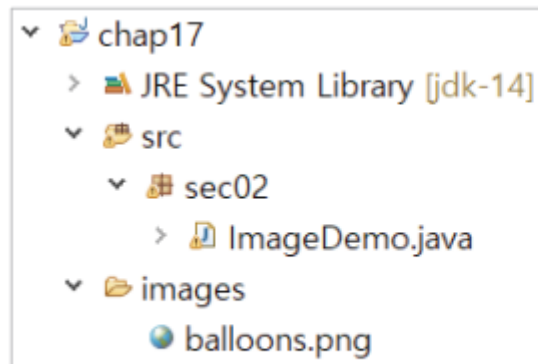
그래픽 그리기

■ 이미지 그리기

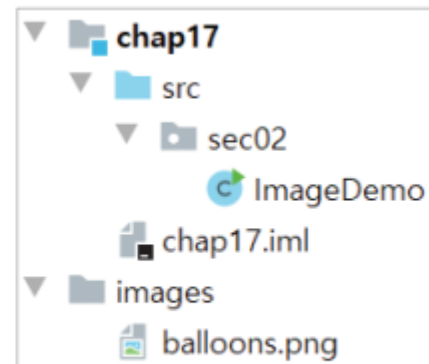
- 예제 : [sec02/ImageDemo](#)



- 이미지의 위치



(a) 이클립스



(b) 인텔리J 아이디어

그래픽 그리기

■ 클리핑

- 실제 이미지의 필요한 부분을 꺼내기 위해 잘라내는 것
- 클리핑은 Graphics 클래스로 그리기를 할 때 지정된 특정 부분만 보이도록 하는 기능
- 클리핑을 위한 사각형 영역을 클리핑 영역이라고 부름
- 영역을 따로 설정하지 않으면 컴포넌트의 전체 영역이 클리핑 영역

메서드	설명
<code>void clipRect(int x, int y, int w, int h)</code>	기존 클리핑 영역과 (x, y) 좌표에서 $w \times h$ 사각형 영역의 교집합을 새로운 클리핑 영역으로 지정한다.
<code>Shape getClip()</code>	클리핑 영역을 반환한다.
<code>void setClip(int x, int y, int w, int h)</code>	(x, y) 좌표에서 $w \times h$ 사각형 영역을 클리핑 영역으로 지정한다.

그래픽 그리기

■ 클리핑

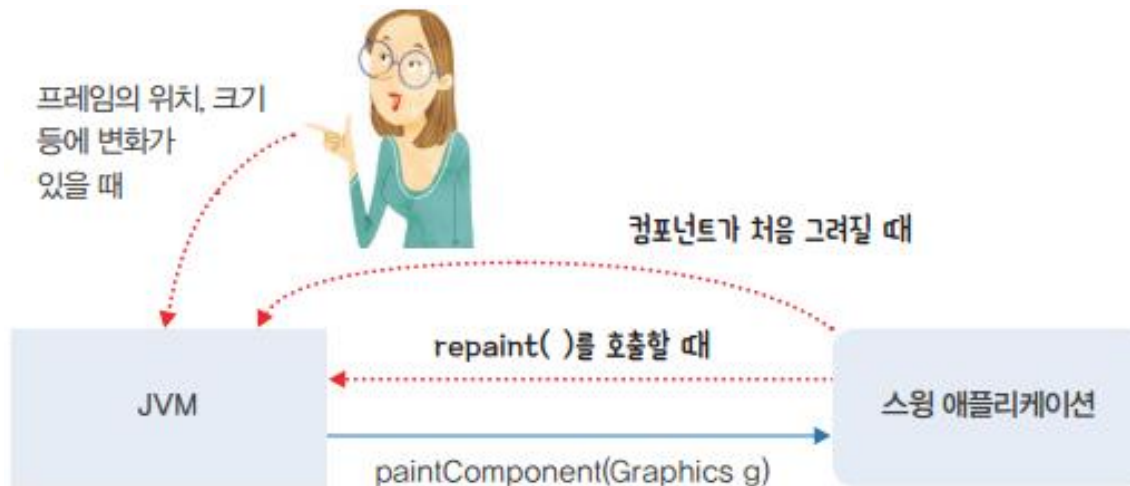
- 예제 : [sec02/ClipDemo](#)



그래픽 그리기

■ 스윙의 그리기 과정

- 스윙 컴포넌트는 모두 JComponent 클래스의 자식 객체로 JComponent 클래스의 paintComponent() 메서드를 호출해서 그린다.
- paintComponent() 메서드는 직접 호출할 수 없고, JVM이 호출하는 콜백 메서드
- paintComponent() 메서드 요청



그래픽 그리기

■ 스윙의 그리기 과정

- 예제: [sec02/RepaintDemo](#)

