

# Chapter 12 자바 그래픽

## 12.1 그래픽 프로그래밍의 기초

### 어디에 그릴 것인가?

- 자바에서 Frame은 다른컴포넌트를 넣어두는 컨테이너의 역할로 설계었기 때문에 프레임에 직접 도형을 그리는 것은 좋은 방법이 아니다.
- 그림을 그릴 수 있는 화면을 가지고 있으며 컨테이너의 역할을 하는 JPanel에 그리는 것이 좋다.

### 어떻게 그리는가?

- 컴포넌트 위에 그림을 그리려면 모든 컴포넌트가 가지고 있는 paintComponent() 메소드를 재정의하여 그림을 그리는 코드를 작성해야 한다.
- paintComponent() : 컴포넌트가 자신의 모습을 화면에 그리는 메소드

### repaint() 메소드

- 사용자는 paintComponent() 메소드를 직접 호출하면 안 되고 자동으로 호출되도록 해야 한다.
- repaint() 메소드 : 사용자가 화면을 다시 그리고 싶은 경우에 paintComponent()를 호출한다.
- 그림이 그려지는 컴포넌트가 JPanel이나 JLabel처럼 그래픽 컴포넌트인 경우 paintComponent() 메소드에서 super.paintComponent(g);를 호출하는 것이 좋다.

### 완전한 예제

```
import javax.swing.JFrame;
import javax.swing.JPanel;

import java.awt.*;

public class MyFrame20 extends JFrame {
    class MyPanel extends JPanel {
        protected void paintComponent(Graphics g) {
            super.paintComponent(g);
            g.drawOval(60, 50, 60, 60);
            g.drawRect(120, 50, 60, 60);
            g.setColor(Color.BLUE);
            g.fillOval(180, 50, 60, 60);
            g.fillRect(240, 50, 60, 60);
        }
    }

    public MyFrame20() {
        setLocation(500, 500);
        setSize(600, 200);
        setTitle("Basic Painting");
        add(new MyPanel());
        setVisible(true);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }
}
```

```

    }

    public static void main(String [] args) {
        MyFrame20 mf = new MyFrame20();
    }
}

```

## 12.2 색상과 폰트 변경하기

- We will study how to change the color of shape.
- We will study how to define the color of shape using Color class.
- we will study how to make and chane the font of letter.

### 색상 객체 생성

- To define and manage color in JAVA, we use Color class in java.awt Package.
- Way 1 : We can represent color through each number from 0 to 255 according to Red, Green, and Blue in Color class.
- Way 2 : As popular colors are predefined as static constant in Color Class, we can use those colors through Class name such as Color.white, Color.black, and Color.yellow.

```

Color c = color.magenta; // Way 2

Color c = new Color(255, 0, 255); // Way 1

Color c = new Color(255, 0, 255, 128) // Transparency

```

### 색상 설정 방법

Constructor	Description
setBackgroundcolor	컴포넌트 객체에서 배경색을 설정한다.
setColor	전경색을 설정한다.
Colot getColor()	현재의 전경색을 반환한다.

### 윈도우에 문자열 출

- We can print Strings in window using drawString() method in Graphics class.

```

g.drawString("Hoeelo World", x, y) // (x, y) 위치에 문자열 "Helloworld"을 출력한
다.

```

### 폰트 객체 생성

- We can designate font of letter using constructor of Font class.

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

Logical Font	Description
"Serif"	배침(serif)를 갖는 가변폭 글꼴, 대표적으로 TimesRoman
"Sanserif"	배침을 갖지 않는 가변폭 글꼴, 대표적으로 Helvetica
"Monospaced"	고정폭을 가지는 글꼴, 대표적으로 Courier
"Dialog"	대화 상자에서 텍스트를 출력하기 위해 사용되는 글꼴
"DialogInput"	대화 상자에서 텍스트 입력을 위하여 사용되는 글꼴

Font Style	Description
Font.BOLD	볼드체
Font.ITALIC	이탤릭체
Font.PLAIN	표준체

## 폰트 설정 방법

- We can designate object of font using setFont method of Graphics class

```
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);
myLabel.setFont(f);
```

## 예제 12-1

```
import javax.swing.JPanel;
import javax.swing.JFrame;

import java.awt.*;

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("Setif 20 points PLAIN", 10, 20);
        g.setFont(f2);
        g.drawString("Sanerif 20 points BOLD", 10);
        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);
    }
}

public class FontTest extends JFrame {
    public FontTest() {
        setLocation(500, 400);
        setSize(500, 200);
        setTitle("Font Test");
        JPanel panel = new MyPanel();
        add(panel);
        setVisible(true);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }

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

```

## 12.3 기초 도형 그리기

### Graphics 클래스

Method type of Graphics class	Description
기초 도형	drawXxx(), fillXxx() 형식의 메소드를 통하여 직선, 사각형, 타원 등을 화면에 그릴 수 있다.
텍스트 문자열	drawString() 메소드를 통해 텍스트를 화면에 그릴 수 있다.
image	drawImage() 메소드를 통해 화면에 이미지를 그릴 수 있다.

### 그래픽 좌표계

- 자바에서 원점 (0, 0)은 왼쪽 상단 구석에 있다.
- getWidth(), getHeight() 메소드를 호출하여 컴포넌트의 폭과 높이를 알 수 있다.

## 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(), drawPolygon()

## 12.4 이미지 출력 및 처리

- ImageIcon 객체에서 Image 객체를 추출하면 화면에 이미지를 출력할 수 있다.

```
ImageIcon icon = new ImageIcon("d://car.png");
img = icon.getImage();
```

- 이미지를 화면에 출력하는 것은 Graphics 클래스의 drawImage()를 사용한다.

```
public void paintComponent(Graphics g) {
    super.paintComponent(g);
    g.drawImage(img, x, y, null);
}
```

## 12.5 그래픽과 이벤트의 결합