

Chapter 09 자바 GUI 기초

9.1 자바 GUI 소개

- Console(명령 프롬프트와 같이 텍스트만을 사용하여 대화하는 것) <-> Graphical User Interface(GUI)

JWT, 스윙, JavaFX

현재 자바에서 사용할 수 있는 GUI

- Abstract Window Toolkit(AWT): 초기 자바 버전에서 제공하던 GUI.
- Swing: 순수 자바로 작성되어 어떤 플랫폼에서도 일관된 화면을 보여줄 수 있다.

AWT	Swing
운영체계가 제공하는 자원으로 작성되어 플랫폼에 따라 다르게 보인다.	자바로 작성되어 플랫폼에 독립적이다.
무겁다.	가볍다.
룩앤필 지원 x	룩앤필 지원
컴포넌트가 적다.	컴포넌트가 많다.
모델 뷰 컨트롤러(MVC) 모델을 따르지 않는다.	MVC 모델을 따른다.

스윙 패키지

Swing Package	Contents
java.awt	GUI 컴포넌트를 위한 부모 클래스 / Color나 Point와 같은 유틸리티 타입의 클래스
java.awt.event	GUI 컴포넌트로부터 발생하는 버튼 클릭과 같은 이벤트를 처리하기 위한 클래스와 인터페이스

9.2 자바 GUI 기초

컨테이너 컴포넌트 / 최상위 컴포넌트

Component 종류	설명	예
단순 컴포넌트	단순한 컴포넌트	JButton, JLabel, JCheckbox, JChoice, JList, JMenu, JPasswordField, JScrollbar, JTextArea, JCanvas
컨테이너 컴포넌트	다른 컴포넌트를 안에 포함할 수 있는 컴포넌트	JFrame, JDialog, JPanel, JScrollPane
최상위 컨테이너	다른 컨테이너 안에 포함될 수 없는 컨테이너	JFrame, JDialog, JApplet

GUI 프로그램을 만드는 절차

1. Container를 생성한다.
2. Component를 추가한다.

프레임을 생성하는 방법 #1 JFrame 객체를 이용하는 방법

```
import javax.swing.JFrame; // 1. swing을 사용하기 위해 javax.swing 패키지 안의
JFrame Class를 import한다.

public class FrameTest {
    public static void main(String [] args) {

        JFrame f = new JFrame("FrameTest"); // 2. new 연산자를 이용하여서 JFrame
객체를 생성한다.

        f.setTitle("MyFrame"); // 3. 프레임 객체의 setTitle() 메소드를 호출하여 프
레이ムの 타이틀을 설정한다.
        f.setSize(300, 200); // 4. 프레임 객체의 setSize() 메소드를 호출하여 프레임의
크기를 변경한다.
        f.setVisible(true); // 5. 프레임 객체의 setVisible() 메소드를 호출하여 프레
임을 화면에 나타낸다.
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); // 6. 프레임 객체의
setDefaultCloseOperation() 메소드를 호출하여 프레임의 닫힘 버튼을 누르면 전체 프로그램이
종료되게 만든다.
    }
}
```

프레임을 생성하는 방법 #2 JFrame을 상속한 클래스의 객체를 이용하는 방법

```
import javax.swing.JFrame; // 1. swing을 사용하기 위해 javax.swing 패키지 안의
JFrame Class를 import한다.

public class MyFrame extends JFrame { // 2. JFrame을 상속하여 MyFrame 클래스를
정의한다.
    public MyFrame() { // 3. MyFrame 클래스의 생성자에서 MyFrame 객체를 초기화한다.
        setSize(300, 200);
        setTitle("MyFrame");
        setVisible(true);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }

    public static void main(String[] args) {
        MyFrame f = new MyFrame(); // 4. main()에서 MyFrame 객체를 생성한다.
        MyFrame 생성자가 호출되면서 모든 작업이 시작된다.
    }
}
```

프레임에 버튼 추가하기

```

import javax.swing.JFrame;
import javax.swing.JButton;
import java.awt.FlowLayout;

public class MyFrame2 extends JFrame {
    public MyFrame2() {
        setSize(300, 200);
        setTitle("MyFrame");

        setLayout(new FlowLayout()); // 1. MyFrame 클래스의 생성자에서
        setLayout(new FlowLayout()); 문장을 실행하여 배치 관리자(컨테이너 안에서 자식 컴포넌트들의
        배치를 담당하는 객체)를 FlowLayout으로 변경한다. 배치 관리자도 객체이므로 new로 생성하였다.
        FlowLayout은 자식 컴포넌트들을 순차적으로 배치하는 배치 관리자다.
        JButton button = new JButton("버튼"); // 2. new 연산자를 이용하여 버튼 객
        체를 생성한다.
        add(button); // 3. add(button); 문장을 실행하여 버튼을 프레임에 추가한다.

        setVisible(true);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }

    public static void main(String [] args) {
        MyFrame2 f = new MyFrame2();
    }
}

```

9.3 컨테이너 살펴보기

- Container 중에서 우리가 가장 많이 사용하는 컨테이너는 JFrame과 JPanel이다.

JFrame Class

- JFrame 은 수많은 조상 클래스들을 가지고 있으므로 조상 클래스가 제공하는 속성과 메소드를 사용할 수 있다.
- JFrame은 Frame(프레임), Manubar(메뉴바), Content Pane(컨텐츠 페인)으로 구성되어 있다.

JFrame 구성	Description
Frame	AWT에서 제공하는 클래스(java.awt.Frame)로서 윈도우 자체를 나타낸다.
Manubar	메뉴를 부착할 수 있는 공간
Content Pane	Componet들을 부착 할 수 있는 판유리(pane) 같은 것
프레임이 제공하는 Method	Contents
setLocation(x, y)	프레임의 위치를 설정한다.
setSize(width, heigh)	프레임의 크기를 설정한다.
setTitle	제목 줄의 제목을 변경한다.
setLayout()	프레임의 배치 관리자를 설정한다.

프레임이 제공하는 Method	Contents
add(components)	프레임의 콘텐츠 페인에 컴포넌트를 추가한다.
setIconImage(iconImage)	윈도우의 제목 줄에 표시할 아이콘을 설정한다.
setResizable(boolean)	사용자가 윈도우의 크기를 조절하도록 허용하는지 여부를 설정한다.
getContentPane()	프레임 안의 콘텐츠 페인을 가져온다.

예제 9-1 JFrame 사용 예제

```
import java.awt.Color;
import java.awt.FlowLayout;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JButton;

public class MyFrame0 extends JFrame {
    public MyFrame0() {
        setLocation(400, 300);
        setSize(400, 150);
        setTitle("Example 9-1");
        setLayout(new FlowLayout());
        getContentPane().setBackground(Color.yellow);

        JButton button1 = new JButton("확인");
        JButton button2 = new JButton("취소");

        add(button1);
        add(button2);

        setVisible(true);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }

    public static void main(String [] args) {
        MyFrame0 mf = new MyFrame0();
    }
}
```

JPanel Class

- Component들을 부착할 수 있도록 설계된 Container
- Button을 Frame에 직접 추가하기보다 Panel에 추가하고 이 Panel을 Frame에 추가하는 것이 일반적이다. 유지 보수 및 배치 관리에 유리하기 때문이다.

JPanel이 제공하는 Method	Contents
add(aComponent)	패널에 컴포넌트를 추가한다.
remove(aComponent)	패널에 컴포넌트를 삭제한다.

JPanel이 제공하는 Method Contents

setBackground(Color c) 패널에 배경색을 변경한다.

```
import java.awt.Color;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JButton;

public class MyPanel extends JFrame {
    JPanel panel;
    JButton button1, button2;

    public MyPanel() {
        setLocation(600, 400);
        setSize(400, 150);
        setTitle("Example 9-2");

        panel = new JPanel();
        panel.setBackground(Color.orange);

        button1 = new JButton("확인");
        button1.setBackground(Color.yellow);

        button2 = new JButton("취소");
        button2.setBackground(Color.GREEN);

        panel.add(button1);
        panel.add(button2);

        add(panel);

        setVisible(true);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }

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

9.4 배치 관리자

- Container 안에 존재하는 Component들의 크기와 위치를 관리하는 객체
- 배치 관리자는 Container 마다 하나씩만 존재해야 한다.

배치 관리자의 종류

java.awt 패키지가 포함하는 배치 관리자	Description
FlowLayout	컨테이너에 추가되는 순서대로 컴포넌트를 부착한다. 위쪽에서 아래쪽으로 왼쪽에서 오른쪽으로 배치한다. 패널의 기본 배치 관리자다.
BorderLayout	컨테이너의 영역을 동서남북, 중앙의 5개 영역으로 구분하여 이 영역에 컴포넌트를 배치한다. 프레임의 기본 배치 관리자다.
GridLayout	컨테이너의 공간을 동일한 크기의 격자로 나누고 이격자에 컴포넌트를 배치한다.
CardLayout	컨테이너에 컴포넌트를 카드처럼 겹치게 쌓아서 배치한다.

배치 관리자 설정하기

```
panel.setLayout(new BorderLayout); // 1. new 연산자를 이용하여 배치 관리자 객체를 만든다.
2. setLayout() 메소드로 배치 관리자를 설정한다.
button.setMaximumSize(new Dimension(300, 200));
button.setAligement(JComponent.CENTER_ALIGNMENT);
```

FlowLayout

FlowLayout의 생성자

Constructor of FlowLayout	Description
FlowLayout()	
FowLayout(int align)	align은 정렬 방법을 지정한다.
FlowLayout(int align, int hGap, int vGap)	간격을 지정한다.

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

public class MyFrame5 extends JFrame {
    public MyFrame5() {
        setTitle("Flow Layout Test");
        setSize(500, 150);
        setLocation(300, 500);

        setLayout(new FlowLayout());

        add(new JButton("Button1"));
        add(new JButton("Button2"));
        add(new JButton("Button3"));
        add(new JButton("Button4"));
        add(new JButton("Button5"));

        setVisible(true);
    }
}
```

```

        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }

    public static void main(String[] args) {
        MyFrame5 f = new MyFrame5();
    }
}

```

BorderLayout

- Container를 5개의 영역(상, 하, 좌, 우, 중앙)으로 구분하고 각각의 영역에 컴포넌트를 배치할 수 있는 배치 관리자
- 영역 표시: "North", "South", "West", "East", "Center"
- 만약 영역을 지정하지 않으면 컴포넌트는 중앙에 놓인다.
- 여러 개의 컴포넌트를 같은 영역에 추가하는 경우, 마지막으로 추가된 컴포넌트만 표시된다.
- 중앙에 컴포넌트를 추가할 때 다른 컴포넌트가 없으면 컨테이너 전체 영역을 차지하며, 다른 컴포넌트가 추가되면 중앙 영역은 줄어든다.

GridLayout

- Container 공간을 격자(Grid)로 나눈 후에 각 셀에 하나씩 Component들을 배치한다.
- Component들이 추가되는 순서대로 격자의 셀을 채우게 된다.

GridLayout 생성자	설명
GridLayout()	1행과 1열의 격자
GridLayout(int rows, int cols)	rows 행과 cols 열
GridLayout(int rows, int cols, int hGap, int vGap)	간격 지정

```

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

public class MyFrame7 extends JFrame {
    public MyFrame7() {
        setTitle("GridLayoutTest");
        setSize(300, 150);
        setLayout(new GridLayout(2, 3));

        add(new JButton("B1"));
        add(new JButton("B2"));
        add(new JButton("B3"));
        add(new JButton("B4"));
        add(new JButton("B5"));
        add(new JButton("ll"));

        setVisible(true);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }

    public static void main(String [] args) {

```

```

        MyFrame7 f = new MyFrame7();
    }
}

```

CardLayout

- 한 번에 하나의 컴포넌트만 볼 수 있게 배치하는 관리자

CardLayout Class의 Method	설명
next(container)	주어진 Container의 다음 카드로 이동한다.
previous(container)	주어진 Container의 이전 카드로 이동한다.
first(container)	주어진 Container의 첫 번째 카드로 이동한다.
last(container)	주어진 Container의 마지막 카드로 이동한다.

```

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

public class MyFrame8 extends JFrame {
    JButton b1, b2, b3;
    Container cPane;
    CardLayout layoutm;

    public MyFrame8() {
        setTitle("CardLayoutTest");
        setSize(300, 150);
        setLocation(400, 400);

        cPane = getContentPane();
        layoutm = new CardLayout();
        setLayout(layoutm);

        JButton b1 = new JButton("Card #1");
        JButton b2 = new JButton("Card #2");
        JButton b3 = new JButton("Card #3");

        add(b1);
        add(b2);
        add(b3);

        b1.addActionListener(e->layoutm.next(cPane));
        b2.addActionListener(e->layoutm.next(cPane));
        b3.addActionListener(e->layoutm.next(cPane));

        setVisible(true);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }

    public static void main(String [] args) {

```



```

        MyFrame8 f = new MyFrame8();
    }
}

```

절대 위치로 배치하기

- Component의 크기와 위치가 외부의 영향을 받지 않도록 해야 하거나 특수한 효과를 주고자 하는 경우에 의도적으로 컴포넌트와 컴포넌트가 겹치도록 하기 위해서 절대 위치를 사용한다.
- 1. 배치 관리자를 null로 설정한다.
- 2. add() 메소드를 사용하여 컴포넌트를 컨테이너에 추가한다.
- 3. setSize(w, h)와 setLocation(x, y) 또는 setBounds(x, y, w, h)를 사용하여 위치와 크기를 지정한다.

9.5 스윙 비주얼 디자인어: WindowBuilder

9.6 기초 컴포넌트들

- 레이블(JLabel): 텍스트를 표시할 수 있는 공간
- 텍스트 필드(JTextField): 사용자가 한 줄의 텍스트를 입력할 수 있는 공간
- 버튼(JButton): 클릭되면 어떤 동작을 실행하는 버튼

레이블

- 레이블: 텍스트를 표시하기 위한 컴포넌트
- 정보 또는 계산의 결과를 표시할 수 있다.
- 레이블은 텍스트와 이미지를 동시에 표시할 수 있다.
- 텍스트를 표시하는데 사용하는 폰트의 종류, 크기, 색상 등을 변경할 수 있다.

```

// 1. 일반적으로 레이블을 생성할 때 표시할 텍스트를 넘긴다.
JLabel label = new JLabel("안녕하세요.");

// 2. 레이블 객체를 먼저 생성하고 나중에 레이블의 텍스트를 설정할 수도 있다.
JLabel label = new JLabel();
label.setText("안녕하세요.");

```

예제 9-3

```

import java.awt.*;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JLabel;

public class LabelTest extends JFrame {
    private JPanel panel;
    private JLabel label1, label2;

    public LabelTest() {
        setTitle("레이블 테스트");
    }
}

```

```

        setSize(400, 150);

        panel = new JPanel();

        label1 = new JLabel("Color Label");
        label1.setForeground(Color.BLUE);
        label2 = new JLabel("Font Label");
        label2.setFont(new Font("Arial", Font.ITALIC, 30));
        label2.setForeground(Color.ORANGE);

        panel.add(label1);
        panel.add(label2);

        add(panel);
        setVisible(true);
    }

    public static void main(String [] args) {
        LabelTest t = new LabelTest();
    }
}

```

레이블에 이미지 표시하기

- ImageIcon : JPEG, GIF, PNG 이미지 파일을 읽을 수 있다.

```

ImageIcon image = new ImageIcon("d://dog.png"); // 1. ImageIcon 인스턴스를 생성한다.
JLabel label = new JLabel("Dog");
label.setIcon(image); // 2. setIcon() 메소드를 사용하여 레이블에 이미지를 지정한다.

```

예제 9-4

```

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JLabel;
import javax.swing.ImageIcon;

public class ImageLabelTest extends JFrame {
    JPanel panel;
    JLabel label;

    public ImageLabelTest() {
        setLocation(600, 400);
        setSize(400, 400);
        setTitle("Example 9-4");

        panel = new JPanel();
        label = new JLabel();
    }
}

```

```

        ImageIcon image = new ImageIcon("Retriever.jpeg");
        label = new JLabel("Retriever");
        label.setIcon(image);

        panel.add(label);

        add(panel);

        setVisible(true);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }

    public static void main(String [] args) {
        ImageLabelTest i = new ImageLabelTest();
    }
}

```

텍스트 필드

- Text Field: 입력이 가능한 한 줄의 텍스트 상자
- 텍스트 필드 안에서 문자열을 선택할 수 있고 선택한 문자열을 복사하여 붙이는 것(Cut and Paste)도 가능하다.
- JTextField 클래스는 사용자가 입력하는 문자를 보여주지 않는 JPasswordField Class와 사용자가 입력할 수 있는 문자 집합을 제한하는 JFormattedTextField Class를 갖는다.

```

JTextField tf = new JTextField(30); // 30자 크기의 텍스트 필드를 만든다.
tf.setText("아이디를 입력하십시오"); // 텍스트 필드의 텍스트를 설정한다.
System.out.println(tf.getText()); // 텍스트 필드의 텍스트를 가져온다.
tf.requestFocus(); // 텍스트 필드가 사용자로부터 입력을 받을 수 있으려면 키보드 포커스를 가
지고 있어야 한다.

```

예제 9-5

```

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JLabel;
import javax.swing.JTextField;
import javax.swing.JPasswordField;
import javax.swing.JButton;

public class LoginWindow2 extends JFrame {
    JPanel panel;
    JLabel l1, l2;
    JTextField tf;
    JPasswordField pf;
    JButton b1, b2;

    public LoginWindow2() {
        setLocation(600, 400);
    }
}

```

```

        setSize(400, 150);
        setTitle("Login Window");

        panel = new JPanel();
        l1 = new JLabel("ID: ");
        l2 = new JLabel("PW: ");
        tf = new JTextField(15);
        pf = new JPasswordField(15);
        b1 = new JButton("Login");
        b2 = new JButton("Cancel");

        panel.add(l1);
        panel.add(tf);
        panel.add(l2);
        panel.add(pf);
        panel.add(b1);
        panel.add(b2);
        add(panel);

        setVisible(true);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }

    public static void main(String [] args) {
        LoginWindow l = new LoginWindow();
    }
}

```

버튼

- Button : 사용자가 클릭했을 때 이벤트를 발생하여 원하는 동작을 하게 한다.
- 버튼에서 우리가 변결할 수 있는 것: 버튼 안의 텍스트, 버튼 텍스트의 폰트, 버튼의 전경색과 배경색, 버튼의 상태(활성, 비활성)

Button provided in swing package	Description
JButton	가장 일반적인 버튼
JCheckBox	체크박스 버튼
JRadioButton	라디오 버튼으로 그룹 중의 하나의 버튼만 체크할 수 있다.
JToggleButton	2가지 상태를 가지고 토글이 가능한 버튼

예제 9-6

```

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JLabel;
import javax.swing.JTextField;
import javax.swing.JButton;

```

```
public class LoginWindow2 extends JFrame {
    JPanel panel;
    JLabel label1, label2;
    JTextField tf1, tf2;
    JButton button;

    public LoginWindow2() {
        setLocation(600, 400);
        setSize(400, 150);
        setTitle("Example 9-6");

        panel = new JPanel();

        label1 = new JLabel("화씨 온도");
        label2 = new JLabel("섭씨 온도");
        tf1 = new JTextField(15);
        tf2 = new JTextField(15);
        button = new JButton("확인");

        panel.add(label1);
        panel.add(tf1);
        panel.add(label2);
        panel.add(tf2);
        panel.add(button);
        add(panel);

        setVisible(true);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }

    public static void main(String [] args) {
        LoginWindow2 l = new LoginWindow2();
    }
}
```

예제 9-7