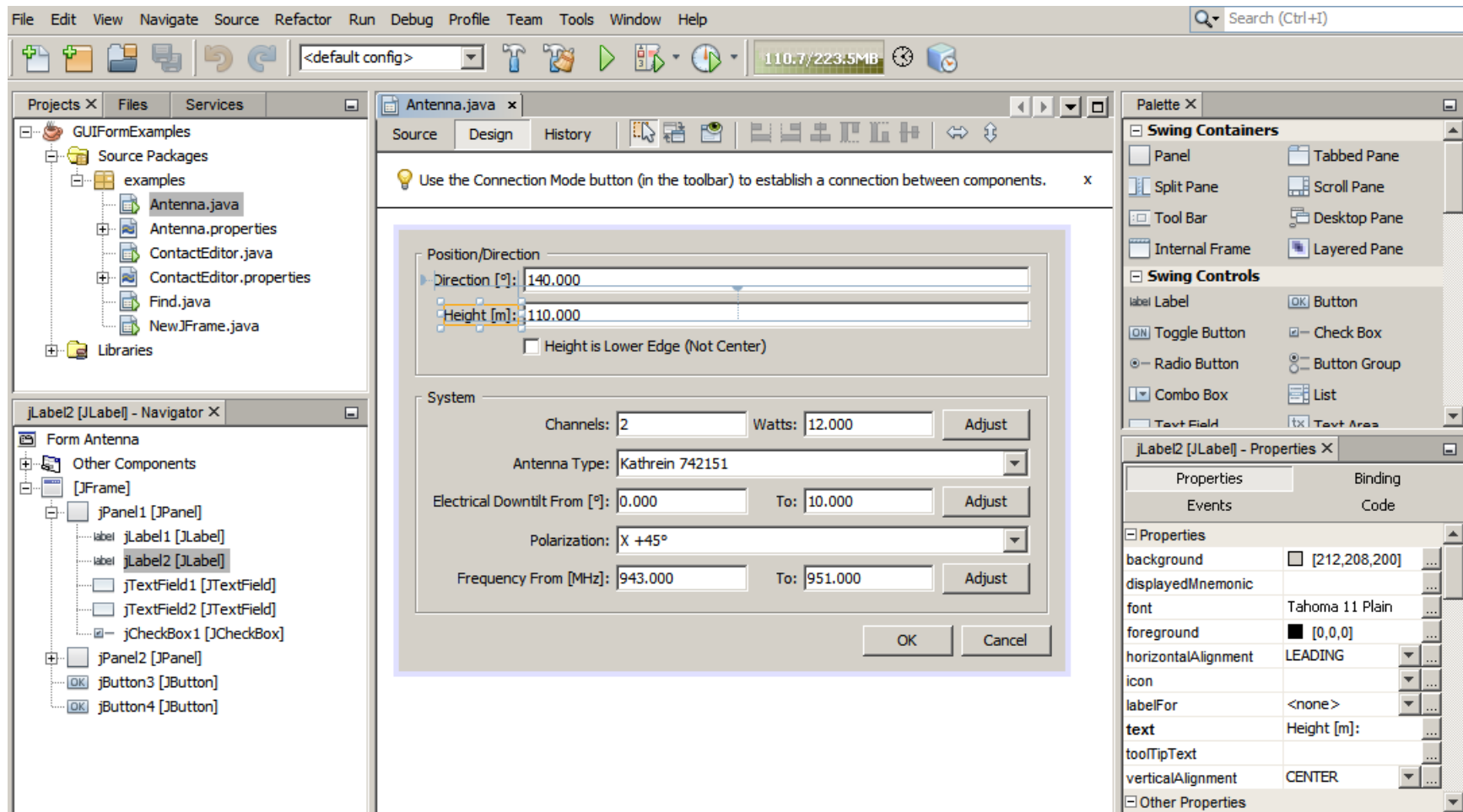


# 프로젝트 안내

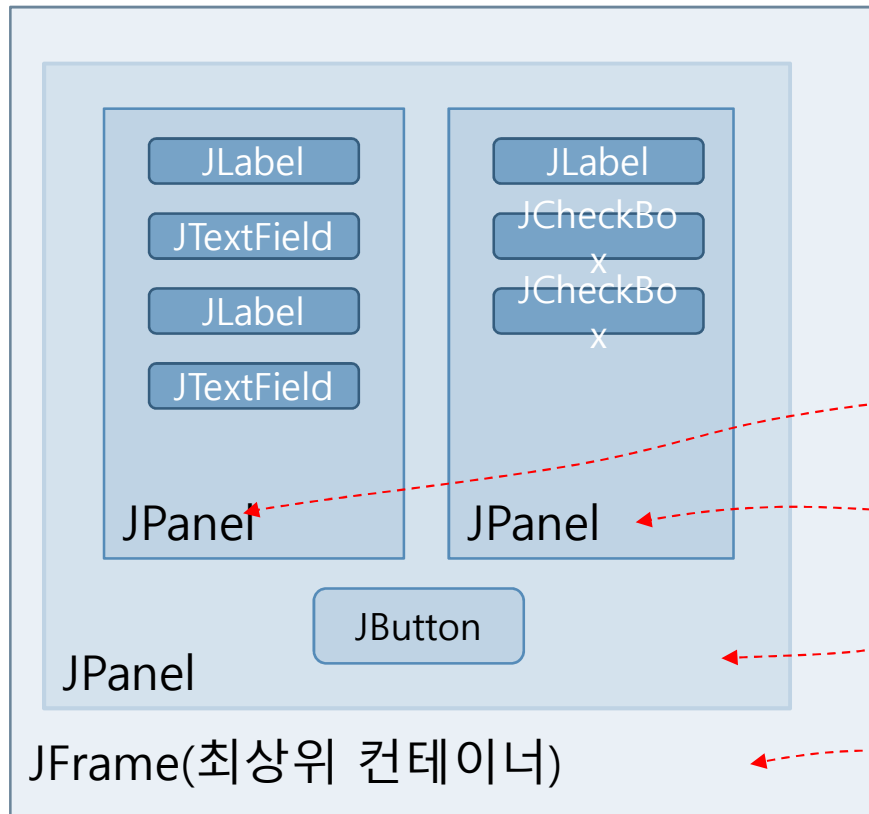
객체지향 프로그래밍 및 실습

# Simple GUI Builder

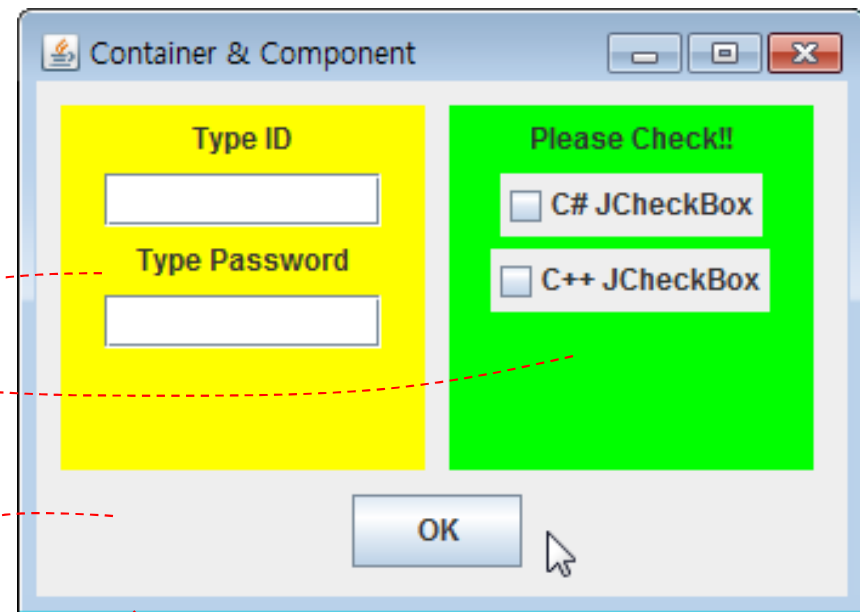


# 컨테이너와 컴포넌트의 포함관계

3



스윙의 컨테이너와 컴포넌트의 포함 관계



스윙 GUI 프로그램

# 컴포넌트를 부착한 프레임 예

4

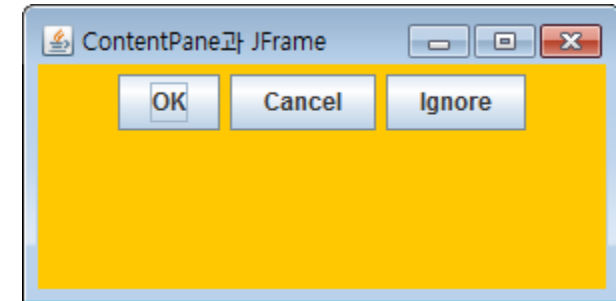
```
import javax.swing.*;
import java.awt.*;

public class ContentPaneEx extends JFrame {
    ContentPaneEx() {
        setTitle("ContentPane과 JFrame");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

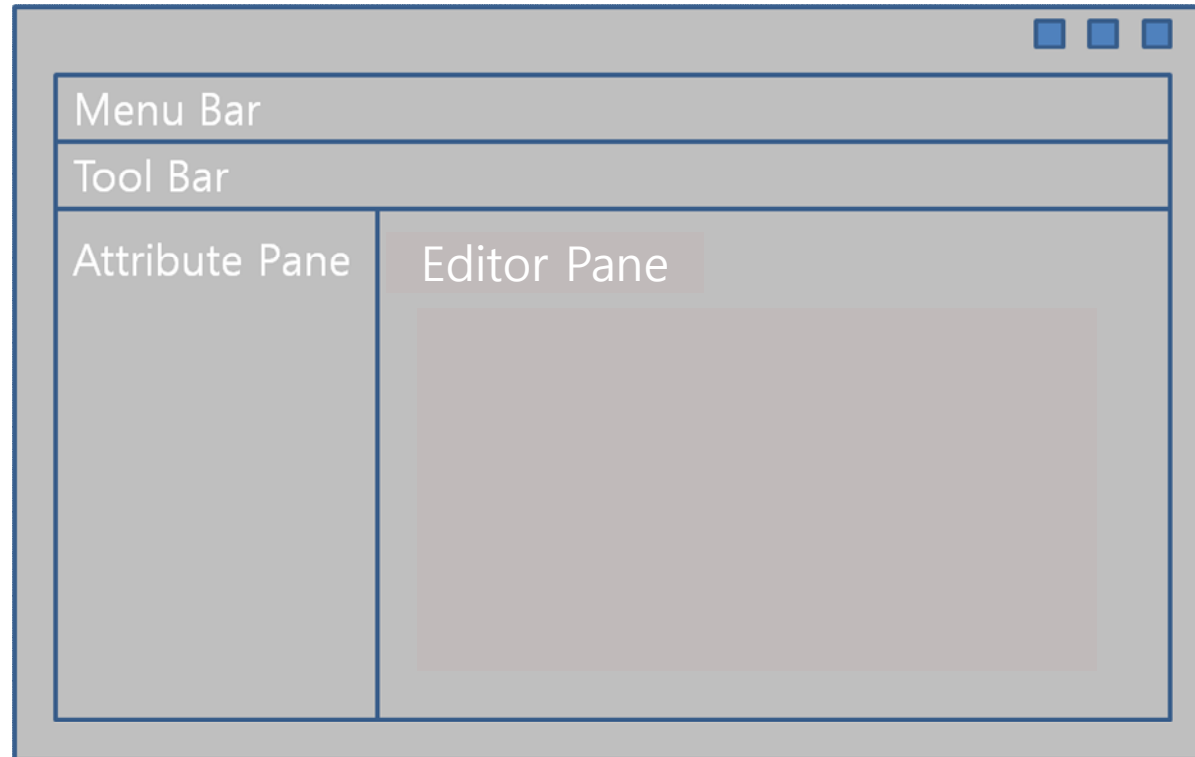
        Container contentPane = getContentPane();
        contentPane.setBackground(Color.ORANGE);
        contentPane.setLayout(new FlowLayout());
        contentPane.add(new JButton("OK"));
        contentPane.add(new JButton("Cancel"));
        contentPane.add(new JButton("Ignore"));

        setSize(350, 150);
        setVisible(true);
    }

    public static void main(String[] args) {
        new ContentPaneEx();
    }
}
```



# 사용자 인터페이스 요구 사항



GUI 구성 요소	SWING 컴포넌트
윈도우	JFrame
메뉴바	JMenuBar
툴바	JToolBar
속성 페인, 에디터 페인	JPanel
컴포넌트	JLabel

# 속성 페인의 속성

속성	SWING 컴포넌트
시작 x, y 좌표	JLabel, JTextField
너비, 높이	JLabel, JTextField
컴포넌트의 텍스트 속성값	JLabel, JTextField
컴포넌트 타입	JLabel, JComboBox
컴포넌트 변수명	JLabel, JTextField

그 외에 필요한 속성을 추가하라.

속성을 표시 및 변경하는 Swing 컴포넌트는 JTable 등으로 변경해도 됨.

또한, 속성 페인에 속성 이외에 필요한 버튼(JButton)등을 추가해도 됨.

예) [변경] 버튼: 사용자에게 의한 변경된 속성 값을 Editor Pane에 반영하는 역할

# 에디터 페인의 요구 사항

이벤트	액션
컴포넌트의 생성	사각형을 그리면 초기 위치와 크기가 결정됩니다.
컴포넌트의 선택	선택된 컴포넌트의 속성은 속성 페인에 표시되며, 컴포넌트의 배경색을 반전시킵니다.
컴포넌트의 위치 이동	선택된 컴포넌트에서 마우스를 프레스 한 후, 마우스 드래그를 통하여 마우스 포인트를 이동시킨 상태에서 마우스를 릴리즈 하게되면 그 위치로 컴포넌트가 이동되도록 합니다.
컴포넌트의 사이즈 변경	선택된 컴포넌트의 너비, 높이를 마우스를 이용하여 변경시킨다.
컴포넌트의 삭제	선택된 컴포넌트를 삭제한다. 삭제 방법은 자유롭게 선택한다.

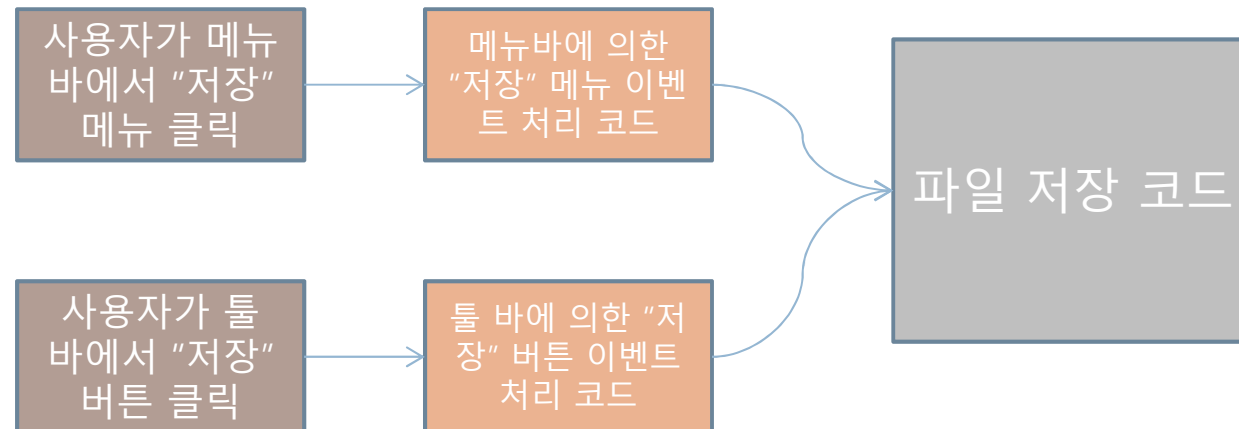
# 메뉴 바/툴 바의 요구 사항

이벤트	액션
새로 만들기	1. 새로운 에디터 페인을 시작합니다. 2. 이미 편집 혹은 불러낸 내용이 있다면, 프로그램이 처음 시작 했을 때의 상태로 되돌립니다.
열기	파일 로드
저장	파일 저장
다른 이름으로 저장	새로운 파일 생성
.java 파일 생성	편집내용을 기반으로 java 파일을 생성한다.
닫기	프로그램 종료



# 메뉴 바/툴 바의 요구 사항

- 모든 이벤트는 메뉴 바와 툴 바 모두에서 발동 가능하도록 구현합니다. 그러나, 각 이벤트에 해당하는 액션은 한 벌만 구현하여 메뉴 바에서 발동 되었건 툴 바에서 발동 되었건 동일한 액션 구현부가 실행 되도록 합니다. [그림]을 참고해서 이해하기 바랍니다. [그림]은 "저장" 이벤트를 예로 삼은 것입니다.



# 파일 생성에 대한 요구 사항

- 파일 저장은 1. 객체 저장 방식, 2. 여러분이 고안한 파일 스펙에 의거하여 바이너리 혹은 문자 방식 스트림을 사용하여 저장 하는 방식, 3. json 타입으로 저장하는 방식의 세 가지 중 하나를 선택합니다.
- 배점은  $3 > 2 > 1$  순으로 높은 점수를 받게 될 것입니다.

# 그 외의 요구 사항

- “MVC 모델”을 사용하기 바랍니다.

# 그 외의 요구 사항

- 지금까지 소개된 최소 요구 사항을 기반으로 자유롭게 완성도 있게 구현하기 바랍니다.
- 프로젝트 검사 시, 자신이 구현한 결과물의 구조를 효과적으로 전달할 수 있도록 보고서를 작성해 오세요.
  - ▣ 형식적으로 작성하지 말고, 검사 시 설명을 도울 수 있는 다이어그램
  - ▣ 결과 요약 표
    - 기능 구현 목록
    - 항목별 구현 완성도 표시 (O, Δ, X)
    - 미완성 기능에 대한 사유, 증상 등 특이사항설명