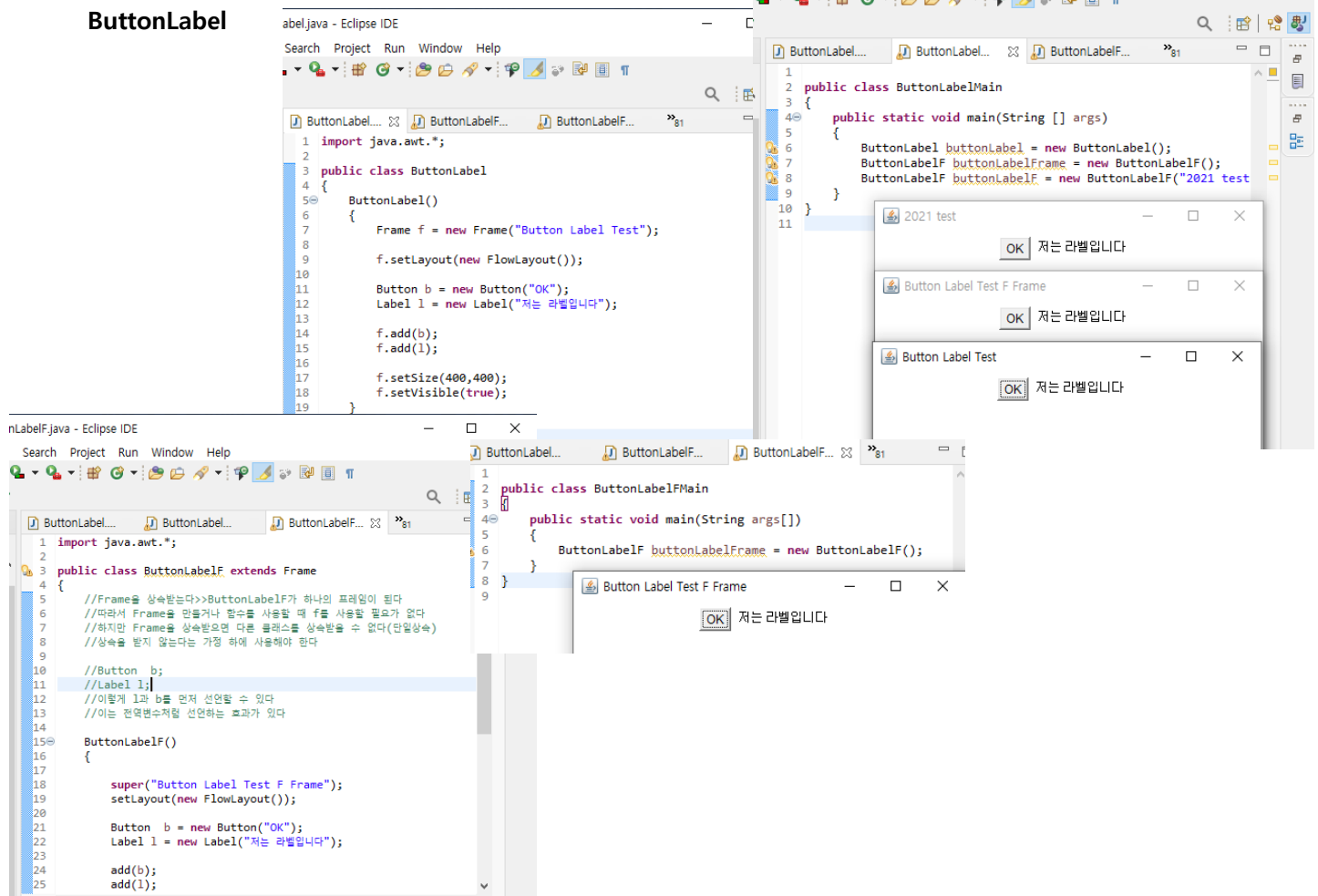


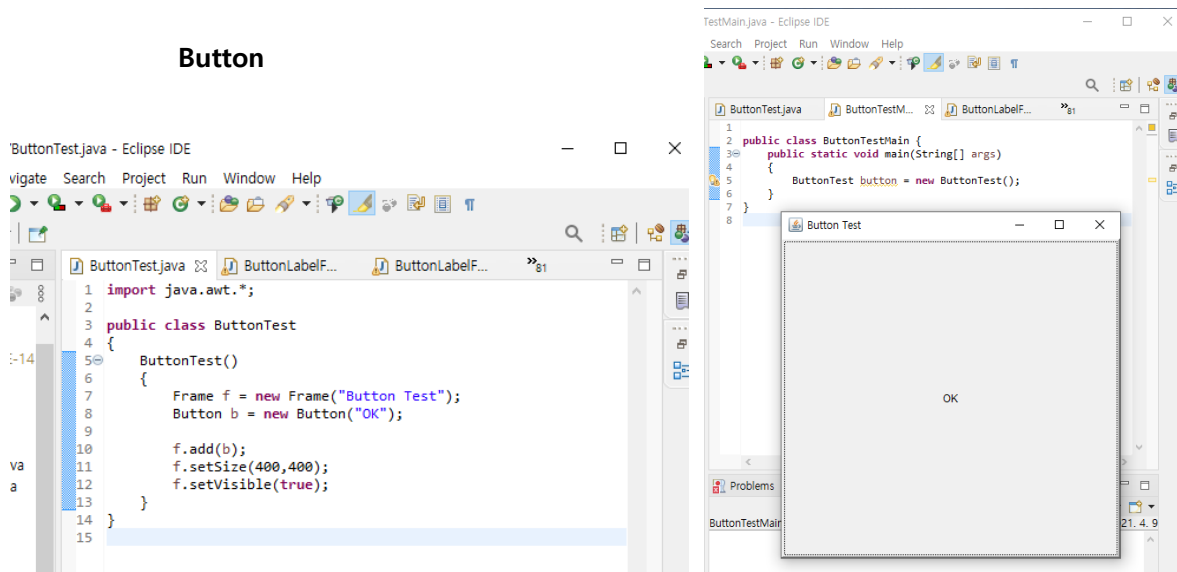
## 객체지향프로그래밍 6 주차 정리

### 프로그래밍

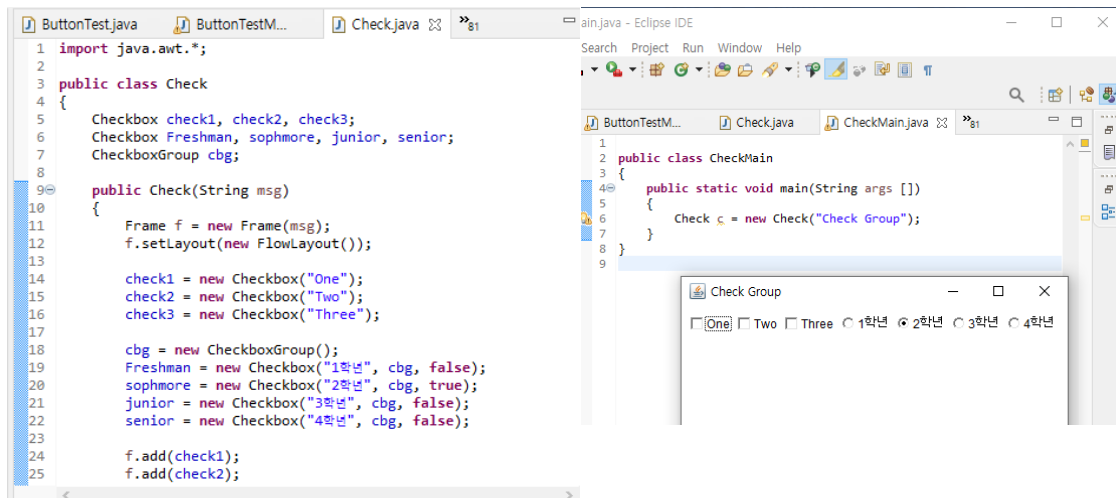
#### ButtonLabel



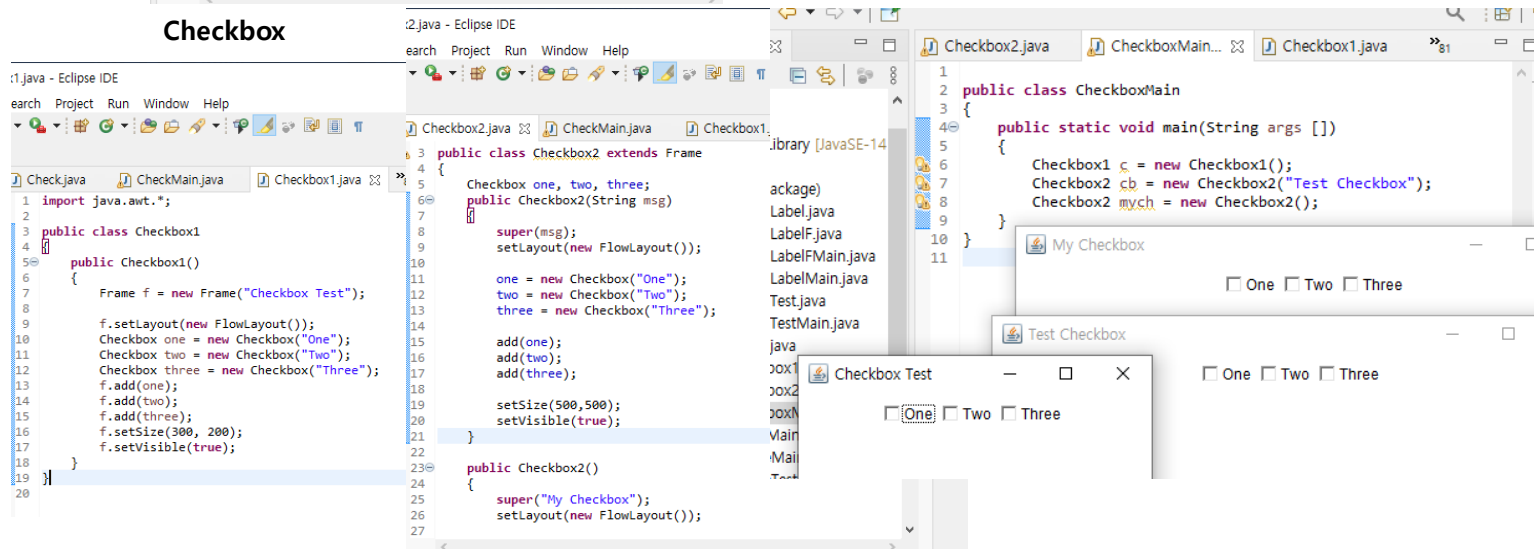
#### Button



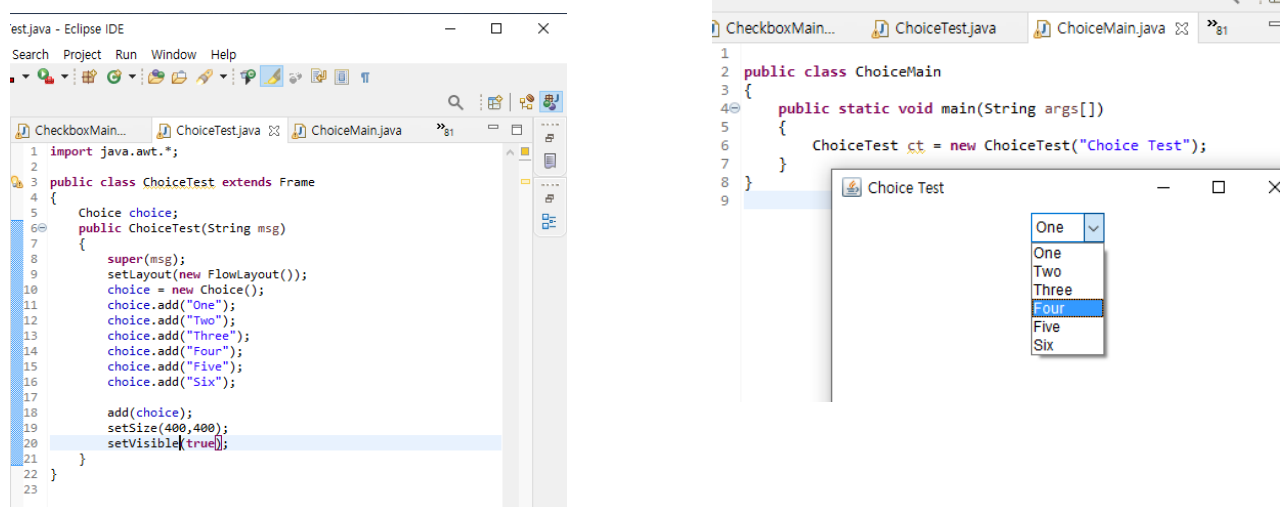
## Check



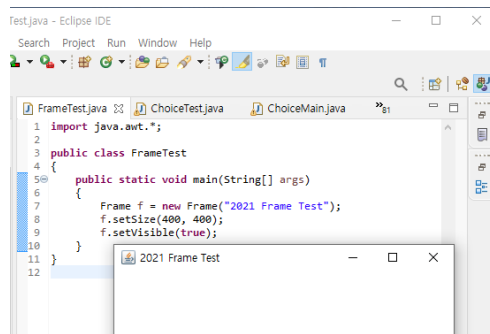
## Checkbox



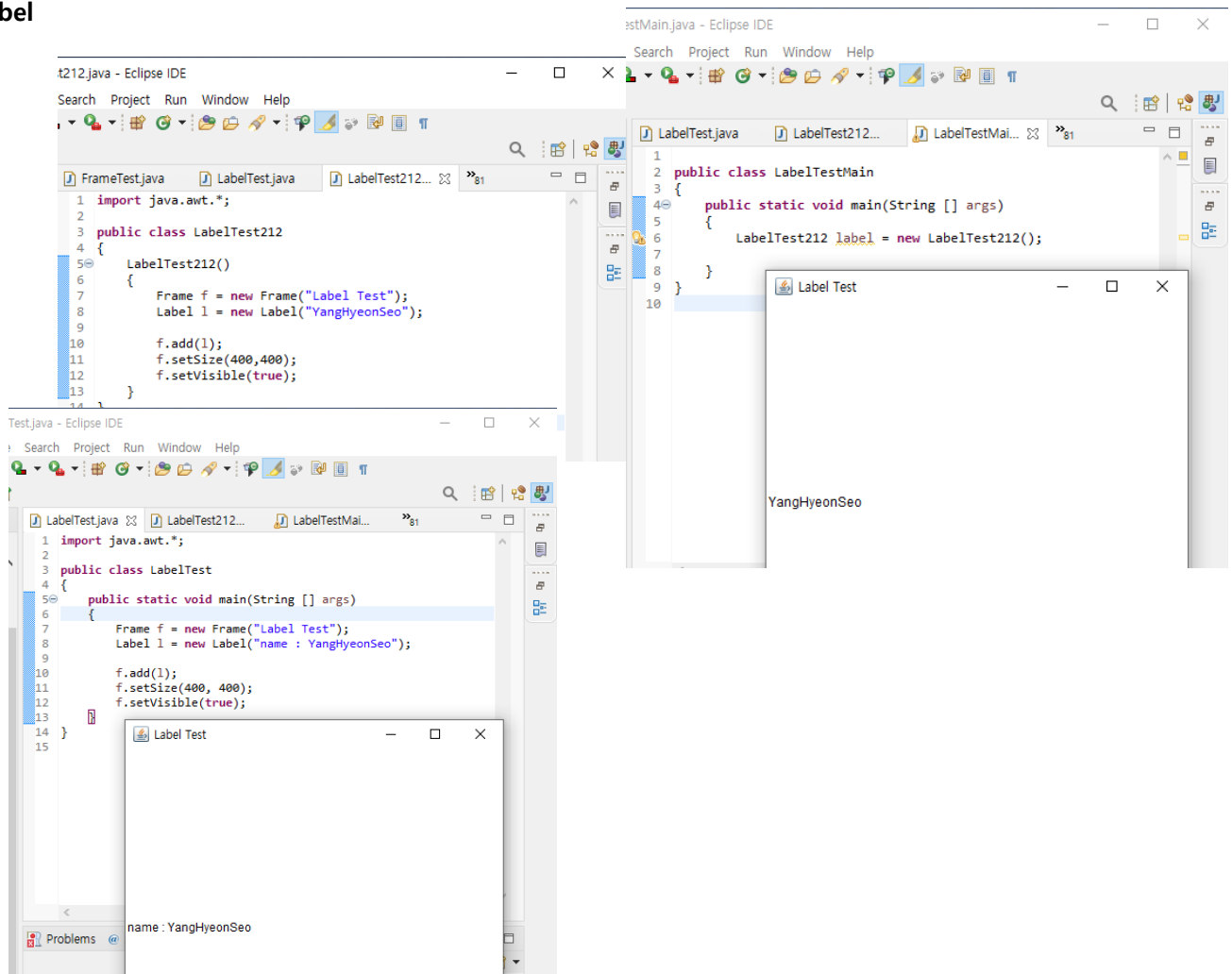
## Choice



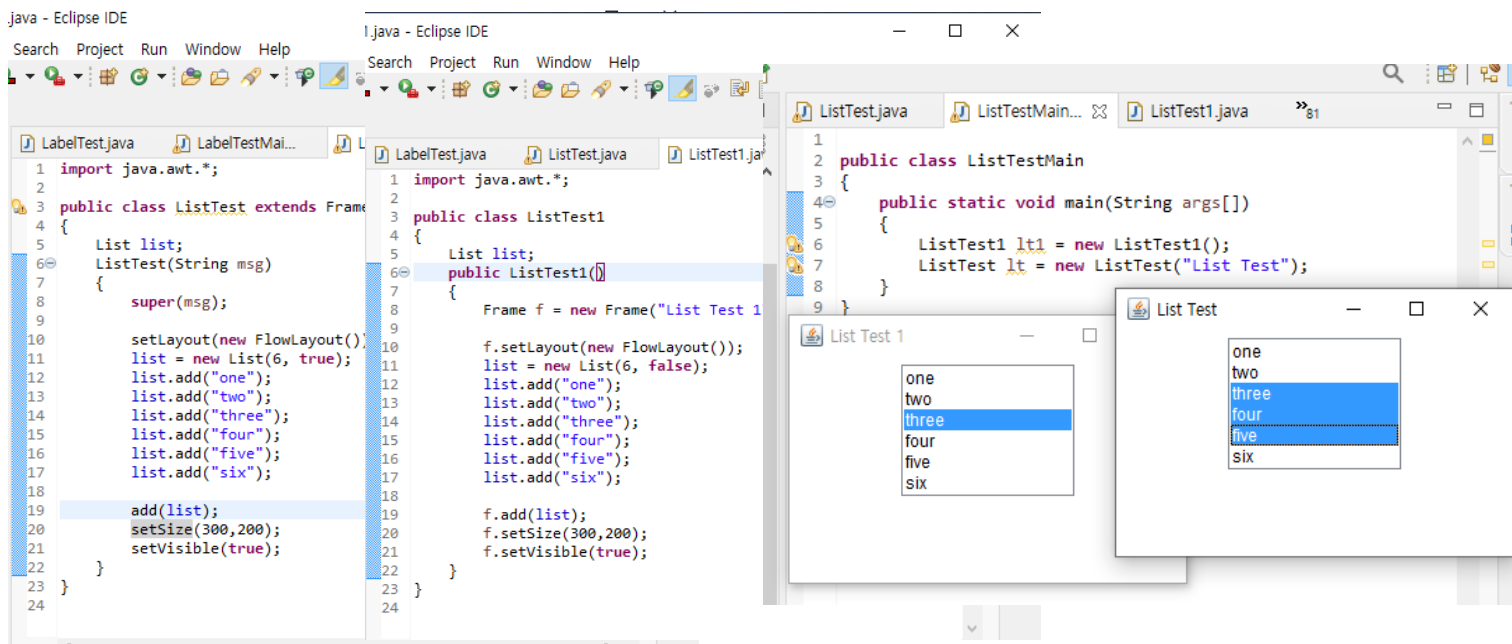
## Frame



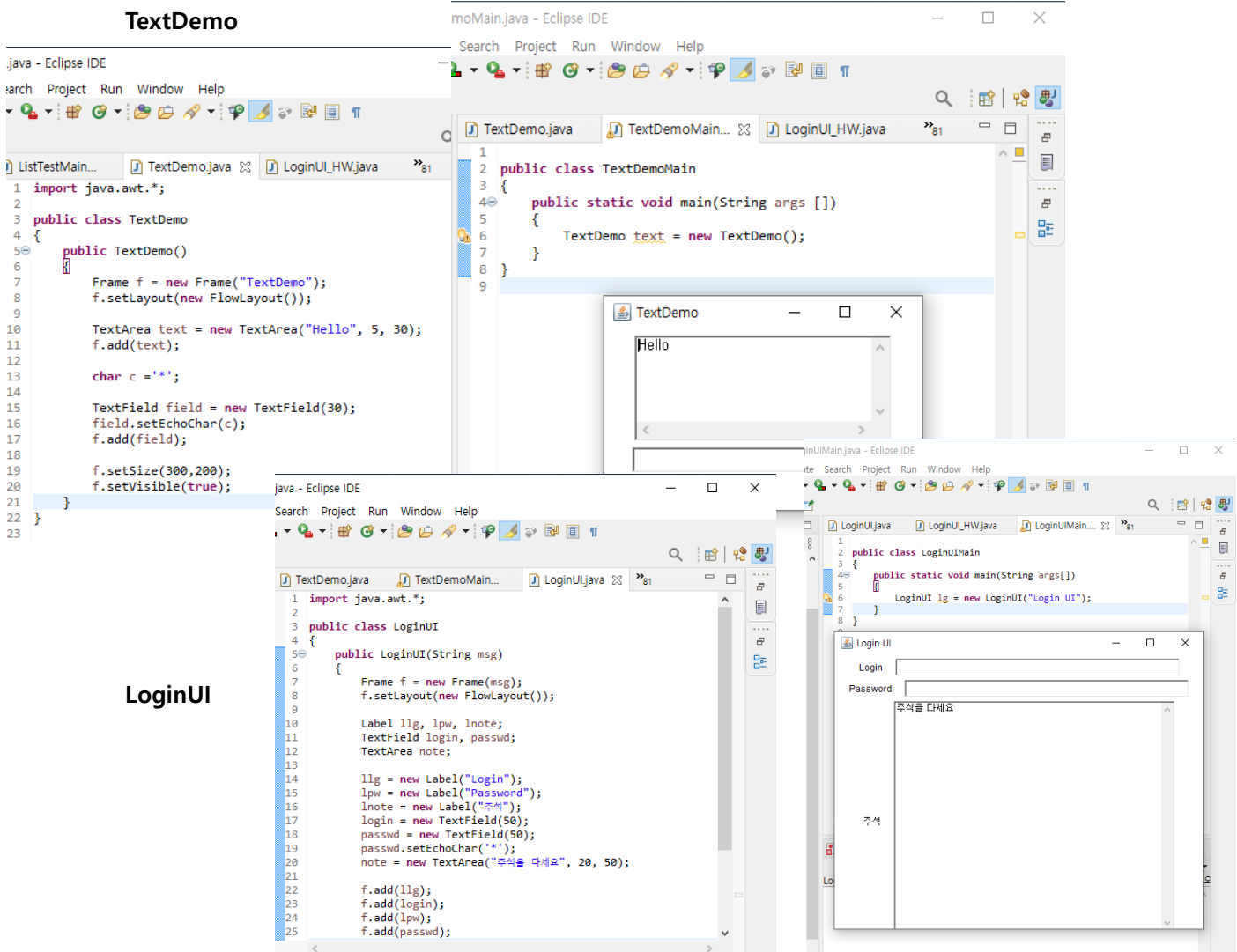
## Label



## List



## TextDemo



## LoginUI HW

```
HW.java - Eclipse IDE
Search Project Run Window Help

LoginUI.java LoginUI_HW.java LoginUIMain.... »81

1 import java.awt.*;
2
3 public class LoginUI_HW
4 {
5     public LoginUI_HW(String msg)
6     {
7         Frame f = new Frame(msg);
8         f.setLayout(new BorderLayout());
9
10        Panel p1 = new Panel();
11        Panel p2 = new Panel();
12        Panel p3 = new Panel();
13        // Frame f1 = new Frame();
14
15        Label llg, lpw, press;
16        TextField login, passwd;
17        Button ok;
18
19        llg = new Label("Login");
20        lpw = new Label("Password");
21
22        login = new TextField(50);
23        passwd = new TextField(50);
24        passwd.setEchoChar('*');
25    }
}
```

```
c/LoginUI_HWMMain.java - Eclipse IDE
Navigate Search Project Run Window Help

LoginUI_HW.java LoginUIMain.... »81

1 public class LoginUI_HWMMain
2 {
3     public static void main(String args[])
4     {
5         LoginUI_HW hw = new LoginUI_HW("Login UI Homework");
6     }
7 }
8
9
```

Login UI HomeWork

Login

Password

Press

Login UI HomeWork

Login

Password

Press

## AWT

:GUI 를 제공하는 자바 인터페이스 컴포넌트의 집합

GUI 를 사용하기 위해 импорт 하는 패키지 이름

자바 초기의 GUI 의 도구, 운영체제의 리소스를 사용한다

GUI(그래픽 유저 인터페이스)

: 그래픽 환경

## SWING

:AWT 보다 확장된 GUI 를 제공한다

순수 자바 컴포넌트 패키지

운영체제와 독립되어 자체 리소스를 사용한다

JButton 과 같이 대문자 J 가 앞에 붙는다

컨테이너: 컴포넌트를 포함한 클래스

컴포넌트: 응용프로그램에서 사용자의 입력을 받거나 출력해주기 위한 기능

버튼, 이미지, 리스트 등의 요소들

컴포넌트는 반드시 컨테이너에 연결하여 사용한다

Object -(상속)-> Component -(상속)-> Container, Button, Canvas...(AWT 패키지)

Object -(상속)-> MenuComponent -(상속)-> MenuItem, Menubar

## Component Class

AWT 가 가지고 있는 공통된 메소드와 기능을 가지고 있다

Add, getSize, setBackground, setEnabled, setFont, setForeground, setLocation...

이 중에서 getSize 와 isVisible 함수가 제일 많이 사용된다(->중요하다)

## Container 컴포넌트

다른 컴포넌트가 add 함수를 통해 Container 에 포함될 수 있다.

### Panel

눈에 보이지 않지만 컴포넌트들을 묶어준다

### Frame

자바 운용을 위한 main Component

Title 과 Layout 을 갖는다

생성자

Frame()

Frame(String) : String 으로 title 지정

주요 메소드

setSize: Frame 의 사이즈

setVisible(true or false) : Frame 을 보이게 할지 말지

FrameTitle(String) : title 지정

### Label

한 줄의 text 를 나타낸다

생성자

Label()

Label(String) : label 내용 지정

Label(String, int) : label 내용 지정, 정렬 지정

주요 메소드

getAlignment()

getText()

setAlignment(int)

setText(String)

### Button

마우스 클릭으로 이벤트를 발생시키게 해준다

생성자

`Button()`

`Button(String)`

주요 메소드

`addActionListener()` : 이벤트를 실행시켜주기 위한 설정

`getLabel()`

`setLabel()`

`getActionCommand()`

`setActionCommand()`

**Checkbox, CheckoxGroup**

**Checkbox** : 요소들을 여러 개 선택 가능

**CheckboxGroup** : 한 그룹에서 하나의 요소만을 선택 가능

생성자

`Checkbox()`

`Checkbox(String)`

`Checkbox(String, Boolean)`

`Checkbox(String, CheckboxGroup, Boolean)`

주요 메소드

`addItemListener()` : 이벤트

`getStatus(), setStatus(Boolean)`

`getLabel(), setLabel(String)`

`getSelectedCheck()`

**Choice**

주어진 리스트에서 하나의 아이템을 선택하는 것

List 와 비슷하지만 하나만 선택한다는 점에서 차이가 있다



생성자

Choice()

주요 메소드

addItemListener()

add(String item)

remove(int position), remove(String item)

getItem(int), getItemCount(), getSelectedIndex, getSelectedItem()

select(int position) select(String item)

List

주어진 리스트에서 여러 개의 아이템을 선택할 수 있다

생성자

List()

List(int rows)

List(int rows, Boolean) : Boolean 이 true 인 경우 > 여러 개 동시에 선택 가능

false 인 경우 > 한 개씩만 선택 가능

주요 메소드

addActionListener()

addItemListener()

add(String item)

removeAll(), remove(int position), remove(item)

getItem(int index), getItemCount(), getSelectedIndex(), getSelectedIndex()

select(int position), select(String item)

Text Components

TextField

한 줄의 text input 이 가능한 컴포넌트

생성자

**TextField()**

**TextField(int)**

**TextField(String text)**

**TextField(String, int)**

#### **주요 메소드**

**addActionListener()**

**setEchoChar(char)** : 비밀번호를 입력할 때 그대로 노출되지 않고 다른 모양으로 나오게 하는 함수

**getSelectedText(), getText()**

**setEditable(Boolean), setText(String)**

### **TextArea**

여러 줄의 텍스트를 입력하는 공간을 만드는 컴포넌트

#### **생성자**

**TextArea()**

**TextArea (int, int)**

**TextArea (String text)**

**TextArea (String, int, int)**

#### **주요 메소드**

**addActionListener()**

**append(String), insert(String, int position)** : 차이점 append 는 뒤에 추가하는 것이고, inset 는 원하는 위치에 추가하는 것

**replaceRange(String, int, int)**

**getSelectedText(), getText(), select(int, int)**

**setEditable(Boolean), setText(String)**

## 주요 메소드

`setEditable(Boolean)`

`getSelectedText()`, `getText()`

`setText(String)`

## setLayout()메소드값

### BorderLayout()

BorderLayout.NORTH		
BorderLayout.WEST	BorderLayout.CENTER	BorderLayout.EAST
BorderLayout.SOUTH		

컨테이너를 5 개의  
영역으로 분할하여  
컴포넌트를 배치한다

### GridLayout()

1	2
3	4
5	6

컨테이너 공간을 격자 모양으로 나누는 방법

### FlowLayout()

컴포넌트를 왼쪽부터 오른쪽으로 순차적으로 위치시키는 방법

### CardLayout()

컴포넌트가 겹쳐서 나열되는 방법

### Null

기존 레이아웃을 제거하고 컴포넌트의 위치와 크기를 변경하는 방법

**void setLocation(int, int)**

**void setSize(int, int)**