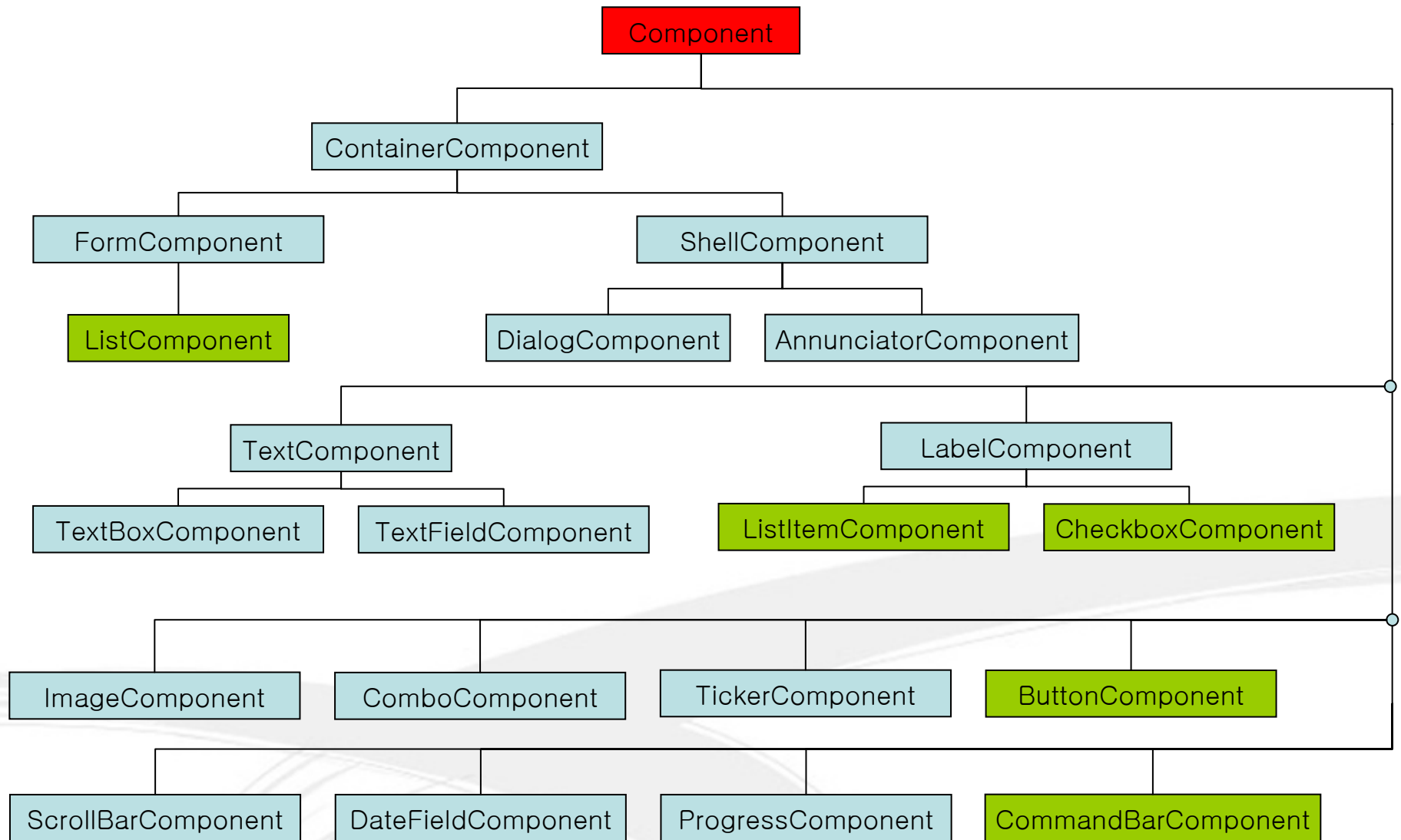


WIPI JAVA 강좌II - 4

org.kwis.msp.lwc package(II)
- Component

1. 컴포넌트의 Listener
2. ButtonComponent
3. CommandBarComponent
4. ListComponent



1. 컴포넌트의 Listener

🌀 Listener 종류

- ◆ ActionListener : Button, List에서 사용
 - `public void action(Component cmp, Object o)`
- ◆ ChangeListener : List, checkbox, progress 에서 사용
 - `public void changed(Component cmp, Object o)`
- ◆ CommandListener : CommandBar에서 사용
 - `public void commandAction(Command c, int type, Object o)`
 - TYPE : FOCUS_CHANGE, SELECT 2가지가 있다.
- ◆ EventListener : 모든 컴포넌트의 이벤트를 받음
 - `public void eventNotify(int type, int arg1, int arg2, int arg3, o)`

2. ButtonComponent 클래스(I)

@ 클래스 설명

- ◆ ActionListener를 등록하여 이벤트를 제어
- ◆ 컴포넌트가 포커스를 가지고 있을때 FIRE버튼을 누르면 이벤트가 발생

@ 주요메소드

- ◆ `setActionListener(listener, obj)` : 리스너 등록, obj는 같이 넘어가는 인자 값
- ◆ `setString(str)` : 버튼에 들어가는 글자를 수정한다.
- ◆ `keyNotify(int type, int key)` : 키 입력을 제어(오버라이드 용)
- ◆ `setImage(img)` : 이미지를 삽입한다.

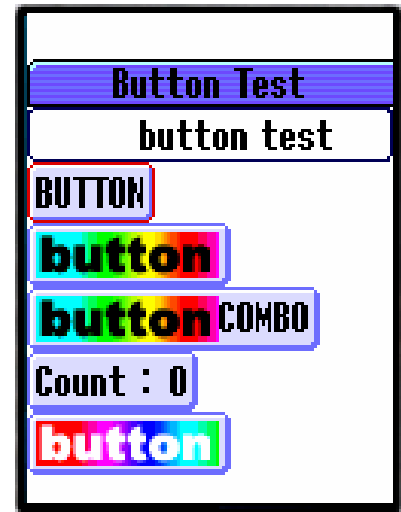
2. ButtonComponent 클래스(II)

@ 사용예

```
ButtonComponent button = new ButtonComponent();  
ActionListener listener = new ActionListenerImp();  
button1.setActionListener(listener, this);  
.....
```

```
public void ActionListenerImp implements ActionListener{  
    public void action(Component cmp, Object obj){  
        // 버튼을 눌렀을때 동작  
    }  
}
```

```
ButtonComponent button1 = new ButtonComponent();  
button1.setActionListener(new ActionListener(){  
    public action(Component cmp, Object obj){  
        // 버튼을 눌렀을때 동작  
    }  
}, this);
```



3. CommandBarComponent 클래스

클래스 설명

- ◆ Shell하단부에 삽입되어 명령 버튼들의 집합을 다룬다.
(`shell.setCommand(commandbar)`로 삽입)
- ◆ `CommandListener`를 이용하여 이벤트를 제어한다.
- ◆ `Command` 클래스를 생성하여 삽입해야 한다.
※ `Command`는 `Component`가 아니다.

주요 메소드

- ◆ `addCommand(command)` : command추가
- ◆ `removeCommand(command)` : command삭제
- ◆ `setActiveIndex(int index)` : 커맨드를 지정(주로 처음에 사용)
- ◆ `setCommandListener(listener, obj)` : 리스너를 등록

Command 클래스

- ◆ Active이미지, normal이미지와 String 1개가 사용

4.ListComponent 클래스

🌀 클래스 설명

- ◆ ListItemComponent만 삽입 가능
- ◆ 삽입순서대로 list에 표시
- ◆ 3가지 타입 지정 가능
 - SELECT_IMPLICIT : 현재 포커스를 이용하여 아이템 선택
 - SELECT_MULTIPLE : 체크박스과 유사
 - SELECT_EXCLUSIVE : 라디오 버튼과 유사
- ◆ 2가지의 이벤트 리스너를 등록
 - ActionListener : SELECT_IMPLICIT에서 주로 사용되며 FIRE로 동작
 - ChangeListListener : 나머지 타입에서 사용되며 방향키로 동작

🌀 주요메소드

- ◆ Append(str, img) : str을 이용하여 ListItem을 자동생성
- ◆ addComponent(listItemComponent) : 직접 ListItem 입력
- ◆ getSelectedIndex(), getSlectedIndexs() : 선택된 index반환
- ◆ setActionListener, setChangeListener : 리스너 등록
- ◆ controlNumber(boolean b) : 리스트번호 사용여부

5. CheckboxComponent 클래스

☞ 클래스 설명

- ◆ 체크 박스 기능을 수행
- ◆ CheckboxGroup을 이용하여 라디오 버튼으로 활용가능

☞ 주요 메소드

- ◆ CheckboxComponent(str, img, checkboxGroup)
문자열과 이미지를 이용하여 생성하며 checkboxGroup이 들어갈수도 있다.
- ◆ setState(boolean state) : 상태를 바꿀때 사용되며 라디오 버튼은 상태가 바뀌면 같은 그룹의 상태도 바뀐다.
- ◆ setChangeListener(listener, obj) : 상태변화를 위한 리스너를 등록하면 변화에 따라 리스너가 동작한다.

☞ CheckboxGroup

- ◆ 체크박스를 라디오 버튼으로 변화시켜줌
- ◆ select(checkbox) : 해당 checkbox를 선택시켜줌
- ◆ getSelectedCheckbox() : 현재의 check된 box를 반환