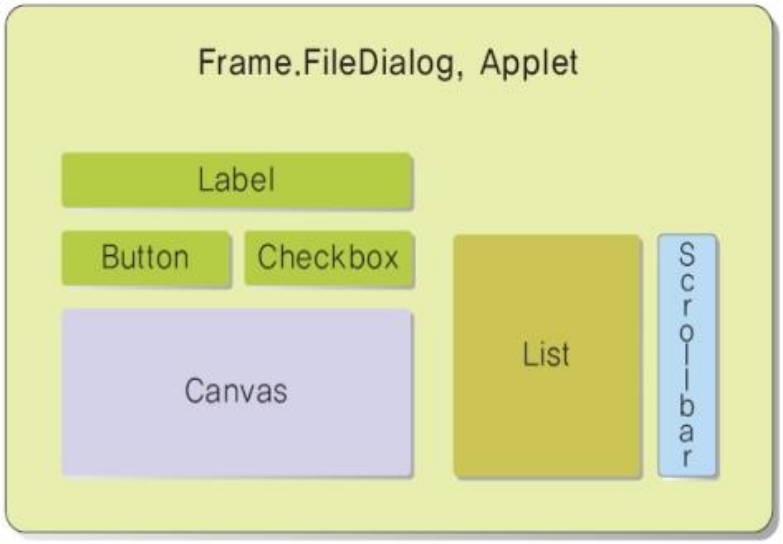
JAVA Programming

Practice 8: AWT

BeomSeok Kim
Department of Computer Engineering
KyungHee University
passion0822@khu.ac.kr

AWT의 구성

- AWT의 구성
  - ✓ 컴포넌트(Component)
    - ➢ 사용자 인터페이스 상에 위치하는 개체들
    - ➢ Button, Canvas, Checkbox, Choice,Label, List, Scrollbar, TextArea,TextField
  - ✓ 컨테이너(Container)
    - ➢ 다른 컴포넌트를 포함할 수 있는 컴포넌트
    - ➢ Panel, ScrollPane, Window, Dialog, Frame, FileDialog, Applet
  - ✓ 배치관리자(LayoutManager)
    - ➢ 컴포넌트들이 컨테이너 안에서 어떻게 배치되는 가를 정의하는 객체



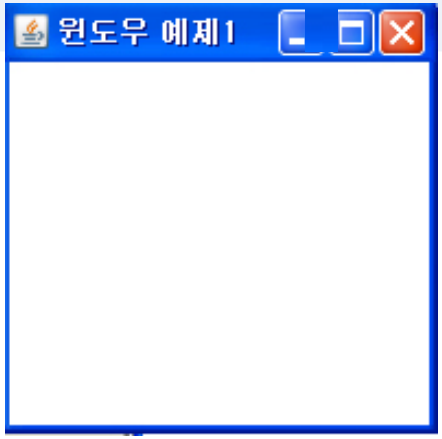
[그림 8-5] AWT 컴포넌트를 이용한 GUI 프로그램의 구조

- 프로그램 화면을 구성 시 레이아웃을 잡는 부분이 가장 중요
  - ✓ 각 컴포넌트 만을 전체화면에 배치할 경우 유지보수하기가 어려움
  - ✓ 컨테이너를 만들어 작업하는 것이 유지보수하기가 유리
    - ➢ 레이아웃 관리자를 이용해 화면을 관리하기 편리하므로 좀더 효율적으로 프로그래밍이 가능
    - ➢ 작은 컴포넌트 객체를 이용한 컨테이너의 객체화

[실습 8-1] 컨테이너 생성 예제

■ 프레임 생성

```
package javabook.ch08;
import java.awt.*;

public class FrameExam {

    public static void main(String[ ] args) {
        Frame f = new Frame("윈도우 예제");
        f.setSize(200, 200);
        f.setVisible(true);
    }

}
```



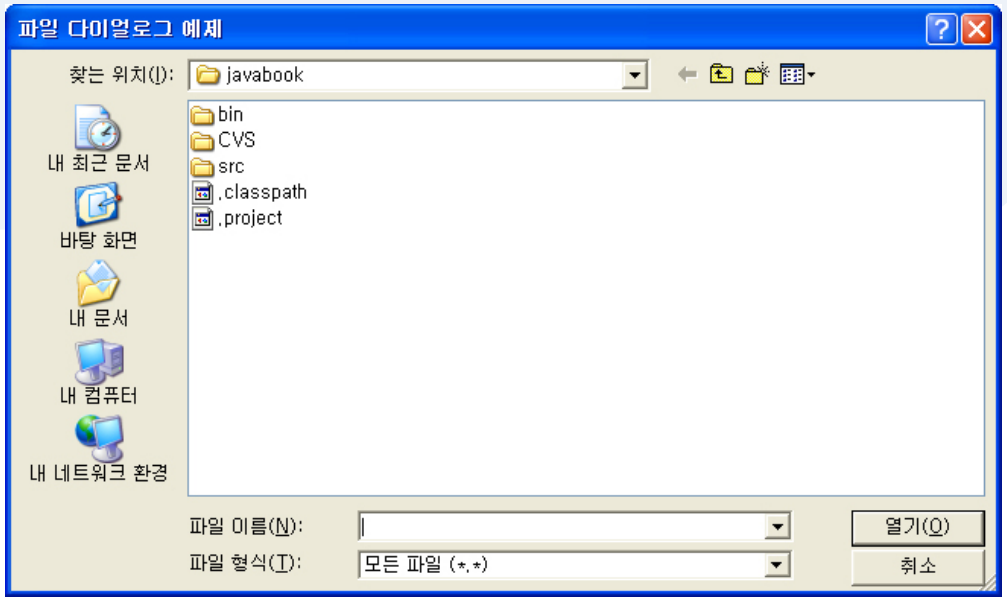
■ 다이얼로그 생성

```
package javabook.ch08;
import java.awt.*;

public class DialogExam {

    public static void main(String[ ] args) {
        Frame f = new Frame( );
        Dialog d = new Dialog(f, "다이얼로그 예제");
        d.setSize(200, 200);
        d.setVisible(true);
    }

}
```

[실습 8-1] 컨테이너 생성 예제

- 파일 다이얼로그 생성

```
package javabook.ch08;
import java.awt.*;

public class FileDialogExam {

    public static void main(String[ ] args) {
        Frame f = new Frame( );
        FileDialog d = new FileDialog(f, "파일 다이얼로그 예제");
        d.setSize(200, 200);
        d.setVisible(true);
    }

}
```



AWT 프로그래밍의 주요 클래스

컴포넌트	메소드
Label	특정한 문자열을 화면에 그대로 보여 주는 용도의 컴포넌트
Buttons	버튼에 적혀질 문자를 표시하고, 누르면 이벤트를 처리
Canvas	선, 사각형, 원과 같은 그래픽을 그릴 수 있는 컴포넌트
Checkbox CheckboxGroup	선택항목의 문자열을 보여 주고, 사용자의 선택에 따라 처리 선택항목이 여럿 인 경우 그룹에 포함하여 처리
Choice	선택목록을 풀다운 메뉴로 보여 주고 사용자의 선택에 따라 처리
TextField TextArea	사용자로부터 한 줄 내의 문자열 정보를 입력 받는 필드 여러 줄의 문자열 정보를 입력 받기 위한 컴포넌트
List	선택목록을 보여 주고 하나 이상을 선택할 수 있도록 하는 컴포넌트
Scrollbar	List나 Text와 같이 사용되며, 보이는 면적이 부족할 때 사용
MenuBar	복수의 메뉴가 들어있는 라인
Menu	메뉴 바 중의 특정한 하나의 메뉴
MenuItem	특정한 메뉴에 속한 하나의 메뉴 항목

[실습 8-2] 기초문제 1

- Button과 Label을 하나의 프레임에 구현
  - ✓ 프레임에 2개의 button 을 만들고, 각 button 의 왼쪽에 Label을 이용한 string 출력을 구현

Button 구현의 예

```
Button b = new Button("OK");        // "OK"라는 이름의 버튼 생성
String buttonName = b.getLabel(); // 버튼의 명칭 획득
```

Label 구현의 예

```
Label l = new Label("성명");     // "성명"이라는 이름의 레이블 생성
String labelName = l.getString();    // 레이블 명칭의 획득
```

[실습 8-3] 기초문제 2

- Label과 Choice를 하나의 프레임에 구현
  - ✓ 프레임에 1개의 Label로 string을 출력하고, choice는 4개의 선택항목을 추가하여 Label 오른쪽에 구현

Label 구현의 예

```
Label l = new Label("성명");     // "성명"이라는 이름의 레이블 생성
String labelName = l.getString();    // 레이블 명칭의 획득
```

Choice 클래스의 주요 메소드

메소드	설명
void add(String item)	문자열로 된 item을 Choice에 추가한다.
String getItem(int index)	Index 위치의 Choice 문자열 항목을 반환한다.
int getItemCount()	항목의 개수를 반환한다.
int getSelectedIndex()	현재 선택된 항목의 위치를 반환한다.
String getSelectedItem()	현재 선택된 항목의 문자열을 반환한다.
void insert(String item, int index)	index 항목에 item 항목을 삽입한다.
void remove(int position)	position 위치의 항목을 삭제한다.
void remove(String item)	item의 문자열과 동일한 항목을 삭제한다.
void removeAll()	모든 항목을 삭제한다.
void select(int pos)	pos 위치의 항목을 선택 상태로 설정한다.
void select(String str)	str 문자열과 동일한 항목을 선택 상태로 설정한다.

[실습 8-4] 기초문제 3

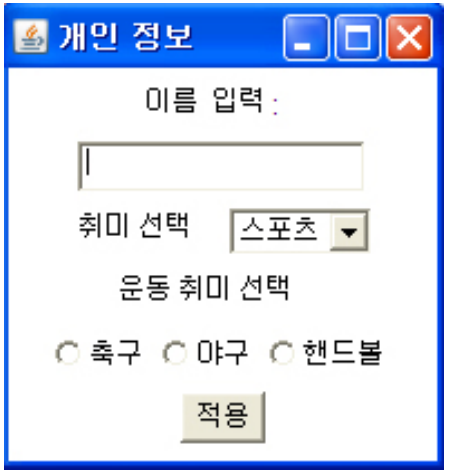
- Checkbox와 CheckboxGroup 을 하나의 프레임에 구현
  - ✓ 프레임에 6개의 checkbox를 만들고, CheckboxGroup을 이용하여 각 3개의 Checkbox를 그룹화

Checkbox와 CheckboxGroup 구현의 예

```
CheckboxGroup cbg = new CheckboxGroup(); // 체크그룹박스 생성
Checkbox cb1 = new Checkbox("여성", cbg, true);  // 체크박스1
Checkbox cb2 = new Checkbox("남성", cbg, false); // 체크박스2
If(cb1.getState() == true) System.out.println("성별 : 여성\n");
else if(cb2.getState() == true) System.out.println("성별 : 남성\n");
```

[실습 8-5] 응용

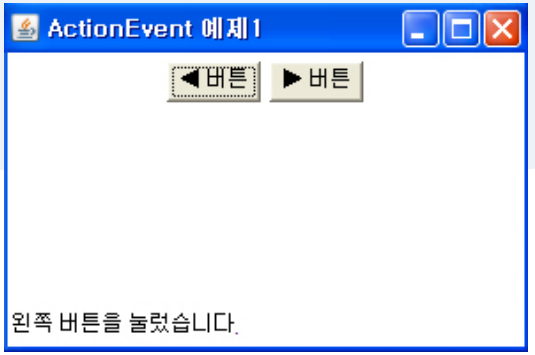
- AWT 컴포넌트를 하나의 프레임에 구현
  - ✓ Button, Checkbox, Choice, TextField 에 대하여 구현
    - ➢ 취미는 “스포츠”, “독서”, “여행” 으로 구성
    - ➢ Checkbox는 단일 선택이 되도록 구현
    - ➢ Layout은 FlowLayout 사용



[실습 8-6] Action Listener를 이용한 이벤트 처리

- 이벤트 처리가 필요한 클래스에 Interface인 ActionListener를 구현
  - ✓ 처리에 필요한 컴포넌트에 addActionListerner 메소드를 이용하여 액션 처리
  - ✓ 이벤트 처리는 public void actionPerformed(ActionEvent e) 에서 구현

```
package javabook.ch08;
import java.awt.*;
import java.awt.event.*;

    public class ActionEventExam2 extends Frame implements ActionListener {

    Label lbl_info;
    Button btn1, btn2;

    public ActionEventExam2(String str) {
        super(str);
        lbl_info = new Label("버튼을 눌러 주세요.");
        btn1 = new Button("◀ 버튼");
        btn2 = new Button("▶ 버튼");
        btn1.addActionListener(this);
        btn2.addActionListener(this);
        Panel panel = new Panel( );
        panel.add(btn1);
        panel.add(btn2);
        add("Center", panel);
        add("South", lbl_info);
        setSize(300, 200);
        setVisible(true);
    }

    public void actionPerformed(ActionEvent e) {
        Object obj = e.getSource( );
        if((Button)obj == btn1) {
            lbl_info.setText("왼쪽 버튼을 눌렀습니다.");
        } else {
            lbl_info.setText("오른쪽 버튼을 눌렀습니다.");
        }
    }

    public static void main(String[ ] args) {
        new ActionEventExam2("ActionEvent 예제");
    }

}
```



[과제 8-1]

- 프레임에 컴포넌트들을 한 개 이상씩 구현하고 그에 따른 액션 이벤트 처리를 추가
  - ✓ 4 페이지에 나열된 AWT 클래스에 명시된 11개의 컴포넌트 중 6개 이상 구현
  - ✓ 1개의 Button은 반드시 구현하고, 해당 Button을 누르면 프로그램 종료
  - ✓ 다른 컴포넌트는 본인의 의도에 맞게 자유 구현

Thank You!
Q&A