

자바 기초

May. 21, 2007

그래픽 사용자 인터페이스

류 대 우

davidryu@newtc.co.kr

학습 목표

- 학습 목표

- 자바 GUI 프로그래밍이 무엇인지 이해한다.
- AWT의 컴포넌트를 이해하고 이벤트를 처리할 수 있다.

GUI와 CUI

- UI (User Interface)
 - 컴퓨터와 사용자 사이에 명령을 전달하는 인터페이스 역할을 하는 것
- CUI (Character User Interface)
 - 키보드로 명령을 입력해서 컴퓨터에 명령을 전달하고 대화함
- GUI (Graphical User Interface)
 - 단추나 아이콘 같은 그래픽 도구를 이용함

패키지의 정의와 종류

- 패키지란?

- 서로 관련된 클래스와 인터페이스 집합
- 하나의 그룹으로 모아 사용한다.
- 자세한 내용은 5강을 참조한다.

패키지의 정의와 종류

- `import java.awt.*;` 처럼 사용된다.(`java.lang`은 생략 가능)
- `java.awt` : 그래픽 사용자 인터페이스를 위한 클래스를 제공하는 패키지
- `java.applet` : 웹에서 수행되는 프로그램을 위한 클래스를 제공하는 패키지
- `java.io` : 파일, 키보드 등 입출력 기능을 위한 클래스를 제공하는 패키지
- `java.lang` : 기본적인 기능을 위한 클래스를 제공하는 패키지
- `java.util` : 유용한 유틸리티 기능을 위한 클래스를 제공하는 패키지
- `java.net` : 통신 기능 등을 위한 클래스를 제공하는 패키지
- `java.sql` : 데이터베이스 기능 등을 위한 클래스를 제공하는 패키지

java.awt 패키지

- 단순 컴포넌트 및 컨테이너 등을 지원한다.
- 단순 컴포넌트란?
 - 버튼, 레이블, 텍스트필드, 텍스트영역, 체크박스, 라디오버튼, 리스트메뉴, 초이스 메뉴, 스크롤 바와 같은 객체임

이벤트 발생과 이벤트 처리

- 객체 이벤트

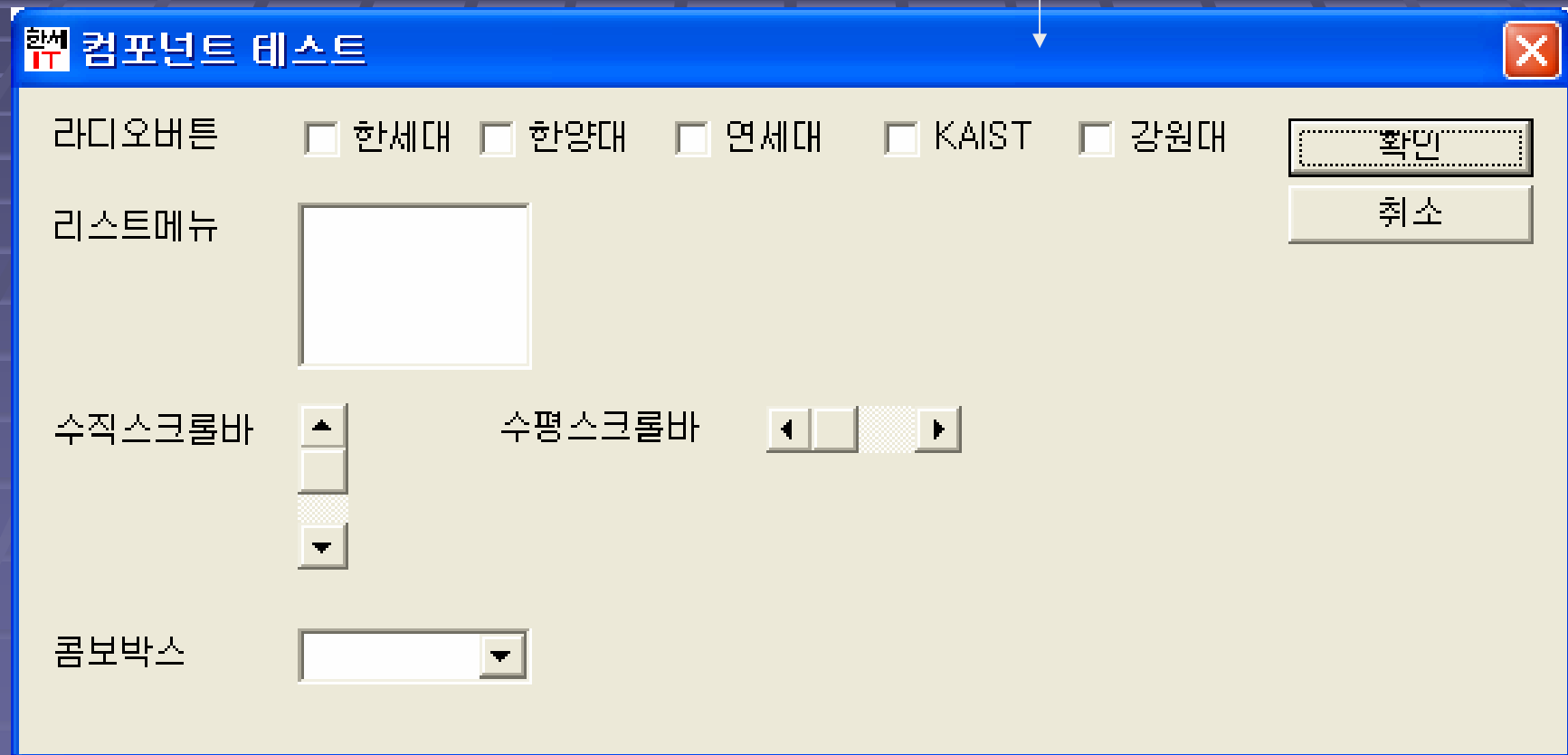
- 사용자가 해당 객체에 특정 조작을 하면 컴퓨터로부터 발생하는 일종의 신호
- 예를 들어,
 - 버튼을 마우스 왼쪽 버튼으로 누르는 행위
 - 텍스트 필드 등에 키보드를 이용해 입력하는 행위
 - 마우스 커서를 옮기는 행위
 - 윈도우의 크기를 조절하는 행위 등

- 이벤트 처리기(Event Handler)

- 이런 행위에 의해 발생한 이벤트를 처리하기 위해 필요한 자바의 메소드나 함수

이벤트 발생과 이벤트 처리

프레임 또는 윈도우라는 컨테이너



버튼 클래스

생성자명	기능
Button()	문자열이 없는 버튼 생성
Button(String label)	문자열이 있는 버튼 생성

Method 명	기능
String getLabel()	버튼의 레이블을 반환한다.
void setLabel(String label)	버튼의 레이블을 지정한다.

버튼 클래스를 생성한 후에 add() 메소드를 사용해 프레임 등 컨테이너에 추가한다.

java.awt 패키지

- java.awt
 - GUI 를 생성하고 그래픽스 및 이미지를 그리는 데 필요한 모든 인터페이스 및 클래스들을 모아둔 패키지입니다.
- java.awt.event
 - GUI 요소들을 조작해 그것을 작동하게 하는 이벤트에 관한 인터페이스 및 클래스들을 모아둔 패키지입니다
- java.awt.image
 - 이미지를 조작하는 각종 인터페이스 및 클래스들을 모아둔 패키지입니다

java.awt 패키지

- java.awt.font
 - 폰트와 관련한 인터페이스 및 클래스들을 제공합니다
- java.awt.geom
 - 2차원 기하와 관련한 기하 개체를 정의하고 조작하는 Java 2D 클래스들을 제공합니다
- java.awt.color
 - International Color Consortium (ICC) Profile Format Specification 에 기초한 색공간 및 색 프로파일 이행에 필요한 클래스들을 제공합니다.

java.awt 패키지

- java.awt.datatransfer
 - 데이터 자르기, 복사하기, 이동하기 등 애플리케이션 내부나 상호간에 데이터를 전달하는 데 필요한 인터페이스 및 클래스들을 제공합니다.
- java.awt.dnd
 - 자바에서 드래그앤드롭 실행에 필요한 인터페이스 및 클래스들을 제공합니다.
- java.awt.im
 - 이 패키지명 im 은 input method 의 약자로, 이 패키지는 입력 method 에 필요한 인터페이스 및 클래스들을 제공합니다. 이는 입력 method 를 통해 모든 컴포넌트가 우리 한국어를 비롯해 중국어나 일본어 텍스트 입력을 수용할 수 있도록 해줍니다
- java.awt.image.renderable
 - 렌더링-독립한 이미지를 만드는 데 필요한 인터페이스 및 클래스들을 제공합니다.
- java.awt.print 프린트에 필요한 인터페이스 및 클래스들을 제공합니다.

java.awt 패키지

- Component (implements MenuContainer)

- +-- Container
 - | +-- Window
 - | | +-- Dialog
 - | | | +-- FileDialog
 - | | +-- Frame (implements MenuContainer)
 - | +-- Panel
 - | +-- ScrollPane
- +-- Canvas
- +-- Button
- +-- CheckBox (implements ItemSelectable)
- +-- List (implements ItemSelectable)
- +-- Choice (implements ItemSelectable)
- +-- Label
- +-- ScrollBar (implements Adjustable)
- +-- TextComponent
 - +-- TextField
 - +-- TextArea

java.awt 패키지

- MenuComponent
 - +-- MenuItem
 - | +-- Menu (implements MenuContainer)
 - | | +-- PopupMenu
 - | +-- CheckboxMenuItem (implements ItemSelectable)
 - +-- MenuBar (implements MenuContainer)

java.awt 패키지

- 이벤트 관련 클래스 및 인터페이스
 - 아래는 상호 종속 관계가 없습니다.
 - Event
 - MenuShortcut
 - AWTEvent
 - AWTEventMulticaster
 - EventQueue

java.awt 패키지

- 그래픽스 관련 클래스 및 인터페이스

- Graphics

- +-- Graphics2D

- Font

- FontMetrics

- Color (implements java.awt.Paint)

- +-- SystemColor

- Image

- Point

- Rectangle (implements Shape)

- Polygon (implements Shape) Dimension

- RenderingHints

- RenderingHints.Key

- TexturePaint (implements Paint)

- GradientPaint (implements Paint)

- BasicStroke (implements Stroke)

- AlphaComposite (implements Composite)

java.awt 패키지

- 기타 지원 클래스 및 인터페이스

- 클래스

- CheckboxGroup
 - Cursor
 - MenuShortcut
 - +-- AWTPermission
 - GraphicsConfigTemplate
 - GraphicsConfiguration
 - GraphicsDevice
 - GraphicsEnvironment
 - ComponentOrientation
 - PrintJob
 - MediaTracker
 - Toolkit

- 인터페이스

- ActiveEvent
 - CompositeContext
 - PaintContext
 - PrintGraphics
 - Transparency

각 클래스에 대한 설명

- Canvas

- 캔버스Canvas 그 자체로 어떤 기능을 갖는 것이 아니라, 그림을 그리거나 텍스트를 편집하는 등 사용자가 자신의 작업을 수행하도록 하는 공간을 말합니다.

- Button

- 버튼Button은 보통 명령 전달에 사용됩니다. 따라서 이 버튼을 누르면 그 용도에 맞는 이벤트가 벌어집니다.

각 클래스에 대한 설명

- CheckBox

- 체크박스CheckBox 는 항목표시용으로 사용된다. CheckBox 중에도 여러 개를 한꺼번에 표시할 수 있는 것과 한 번에 하나씩만 표시할 수 있는 것이 있는데, 특히 한 번에 하나씩만 표시할 수 있는 체크 박스는 라디오버튼이라 부르고 CheckBoxGroup 클래스를 이용하여 생성한다.

- List

- 리스트List 는 선택가능한 텍스트 항목을 포함하는 영역으로 항목이 많을 때는 스크롤바가 부착된다.

각 클래스에 대한 설명

- Choice

- 선택창Choice 도 List 처럼 여러 개의 항목 중에 하나를 선택하는 기능을 갖지만, 스크롤바가 부착되어 있는 List 와는 달리 메뉴버튼을 누르면 항목이 나타나는 형식으로 일종의 메뉴에 해당한다. 흔히 Choice 를 팝업 리스트 pop-up list라 부르기도 한다.

- Label

- 라벨Label 은 컴포넌트 제목달기에 사용되는 것으로 특별한 기능을 수행하지는 않는다.

각 클래스에 대한 설명

- ScrollBar
 - 스크롤바ScrollBar 는 그 내용이 하나의 창에 다 나타나지 않을 때 스크롤하면서 내용을 확인할 수 있게 해주는 컴포넌트로, 주로 TextComponent 나 List 등에 부착된다.
- TextField
 - 텍스트필드TextField 는 한 줄로된 텍스트 디스플레이 창으로 보통 텍스트 입력용으로 사용된다.
- TextArea
 - 텍스트영역TextArea 는 여러 줄로 구성된 텍스트 디스플레이 창으로 보통 텍스트 출력용으로 사용된다

버튼 예제

- `import java.awt.*;`
- `public class ButtonEx extends Frame{`
- `private Button bt;`
- `public ButtonEx(){`
- `bt = new Button("버튼예제");`
- `add(bt);`
- `}`
- `public static void main(String args[]){`
- `ButtonEx ex = new ButtonEx();`
- `//→ 프레임의 크기를 컴포넌트 크기에 맞게 JVM이 정함`
- `ex.pack();`
- `ex.setVisible(true);`
- `}`
- `}`



전체 예제 1

- TextField에서 키가 눌리면 눌림을 인식해서 TextArea에 표시하도록 하시오.

해답

```
import java.awt.*;
import java.awt.event.*;

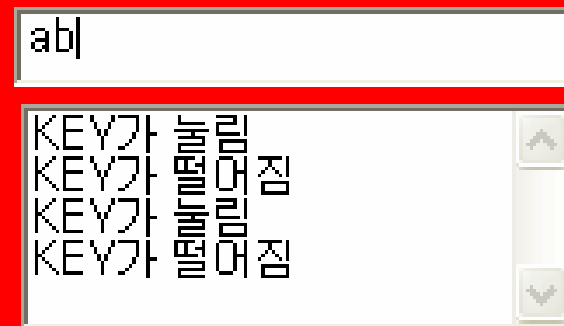
public class AwmEx2 extends Frame implements KeyListener{
    private Window w;
    private TextArea ta;
    private TextField tf;
    private static int pressed;

    public AwmEx2(){
        w = new Window(this);
        w.setLayout(new FlowLayout());
        w.add(new Label("키눌림 감지기"));
    }

    public void keyPressed(KeyEvent e){
        if(pressed == 0){
            ta.append("KEY가 눌림\n");
            pressed = 1;
        }
    }

    public void keyReleased(KeyEvent e){
        if(pressed == 1){
            ta.append("KEY가 떨어짐\n");
            pressed = 0;
        }
    }
}
```

키눌림 감지기



해답

```
import java.awt.*;
public void keyTyped(KeyEvent e){
}

public void startFrame(){
    tf = new TextField(20);
    tf.addKeyListener(this);
    w.add(tf);

    ta = new TextArea(3,20);
    ta.setEditable(true);
    w.add(ta);
    //w.pack();
    w.setSize(200,150);
    w.setBackground(Color.red);
    w.setVisible(true);
}

public static void main(String args[]){
    pressed=0;
    AwmEx2 ex = new AwmEx2();
    ex.startFrame();
    ex.pack();
    ex.setVisible(true);
}
}
```

전체 예제 2

- TextField에서 키가 눌리면 눌림을 인식해서 어떤 키가 눌렸는 지 TextArea에 표시하도록 하시오.

해답

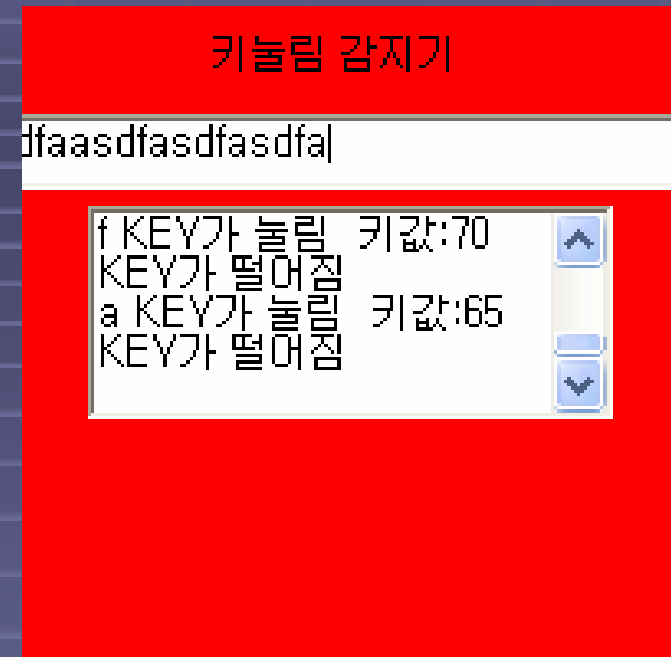
```
import java.awt.*;
import java.awt.event.*;

public class AwmEx3 extends Frame implements KeyListener{
    private Window w;
    private TextArea ta;
    private TextField tf;
    private static int pressed;

    public AwmEx3(){
        w = new Window(this);
        w.setLayout(new FlowLayout());
        w.add(new Label("키눌림 감지기"));
    }

    public void keyPressed(KeyEvent e){
        char c = e.getKeyChar();
        int kcode = e.getKeyCode();
        if(pressed == 0){
            ta.append(c+" KEY가 눌림 키값:"+kcode+"\n");
            pressed = 1;
        }
    }

    public void keyReleased(KeyEvent e){
        if(pressed == 1){
            ta.append("KEY가 떨어짐\n");
            pressed = 0;
        }
    }
}
```



해답

```
public void keyTyped(KeyEvent e){
}

public void startFrame(){
    tf = new TextField(40);
    tf.addKeyListener(this);
    w.add(tf);

    ta = new TextArea(3,20);
    ta.setEditable(true);
    w.add(ta);
    //w.pack();
    w.setSize(200,200);
    w.setBackground(Color.red);
    w.setVisible(true);
}

public static void main(String args[]){
    pressed=0;
    AwmEx3 ex = new AwmEx3();
    ex.startFrame();
    ex.pack();
    ex.setVisible(true);
}
}
```