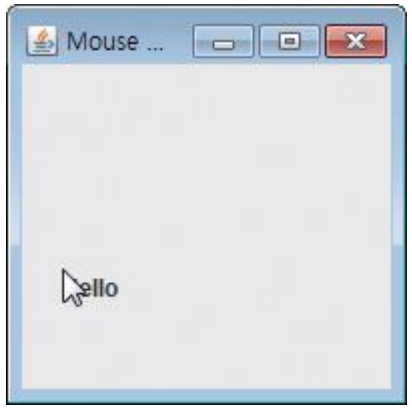
다양한 컴포넌트 활용

마우스 이벤트 리스너 작성 연습 - 마우스로 문자열 이동시키기

아래 실행 화면과 같이 프레임의 임의의 위치에 마우스 버튼을 누르면 마우스 포인터가 있는 위치에"Hello" 문자열을 출력하는 프로그램을 작성하라.

초기화면



마우스 다른 곳에 클릭한 경우

마우스 다른 곳에 클릭한 경우

- 마우스 버튼을 누르면 마우스가 있는 위치로 "Hello" 문자열을 이동시킨다.
- 이벤트와 리스너 : MouseEvent와 MouseListener
- 이벤트 소스 : 컨텐트팬
- 컨텐트팬의 배치관리자 : 배치관리자 삭제
- 구현할 리스너의 메소드 : mousePressed()
- "Hello" 문자열 : JLabel 컴포넌트 이용

정답

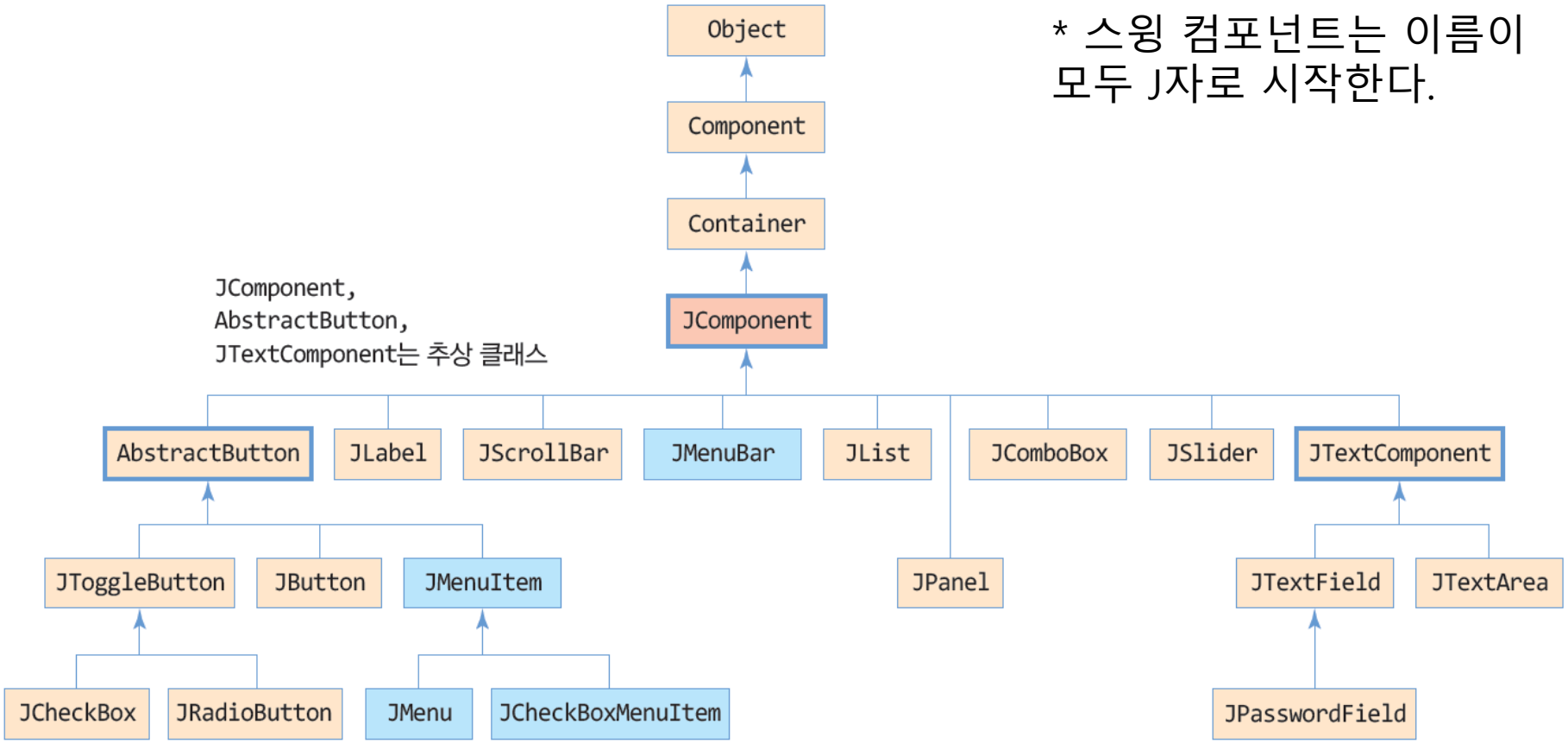
마우스 버튼이 눌러진 위치를 알아내어 "Hello" 를 옮긴다.

MouseListener의 5개 메소드를 모두 구현한다.

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

public class MouseListenerEx extends JFrame {
   JLabel la = new JLabel("Hello"); // "Hello" 출력용 레이블

   MouseListenerEx() {
      setTitle("Mouse 이벤트 예제");
      setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      Container c = getContentPane();
      c.addMouseListener(new MyMouseListener());

      c.setLayout(null);
      la.setSize(50, 20); // 레이블의 크기 50x20 설정
      la.setLocation(30, 30); // 레이블의 위치 (30,30)으로 설정
      c.add(la);

      setSize(200, 200);
      setVisible(true);
   }
```

```
   class MyMouseListener implements MouseListener {
      public void mousePressed(MouseEvent e) {
         int x = e.getX(); // 마우스의 클릭 좌표 x
         int y = e.getY(); // 마우스의 클릭 좌표 y
         la.setLocation(x, y); // (x,y) 위치로 레이블 이동
      }

      public void mouseReleased(MouseEvent e) {}
      public void mouseClicked(MouseEvent e) {}
      public void mouseEntered(MouseEvent e) {}
      public void mouseExited(MouseEvent e) {}
   }

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

컴포넌트 기반 GUI 프로그래밍에 사용되는 스윙 컴포넌트



* 스윙 컴포넌트는 이름이 모두 J자로 시작한다.

스윙 컴포넌트의 공통 메소드, JComponent의 메소드

- **JComponent**
  - 스윙 컴포넌트는 모두 상속받는 슈퍼 클래스, 추상 클래스
  - 스윙 컴포넌트들이 상속받는 공통 메소드와 상수 구현
  - JComponent의 주요 메소드 사례

컴포넌트의 모양과 관련된 메소드

```
void setForeground(Color)  전경색 설정
void setBackground(Color)  배경색 설정
void setOpaque(boolean)  불투명성 설정
void setFont(Font)  폰트 설정
Font getFont()  폰트 리턴
```

컴포넌트의 위치와 크기에 관련된 메소드

```
int getWidth()  폭 리턴
int getHeight()  높이 리턴
int getX()  x 좌표 리턴
int getY()  y 좌표 리턴
Point getLocationOnScreen()  스크린 좌표상에서의 컴포넌트 좌표
void setLocation(int, int)  위치 지정
void setSize(int, int)  크기 지정
```

컴포넌트의 상태와 관련된 메소드

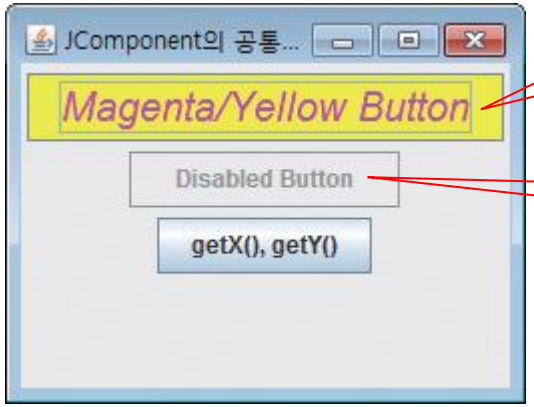
```
void setEnabled(boolean)  컴포넌트 활성화/비활성화
void setVisible(boolean)  컴포넌트 보이기/숨기기
boolean isVisible()  컴포넌트의 보이는 상태 리턴
```

컨테이너를 위한 메소드

```
Component add(Component)  자식 컴포넌트 추가
void remove(Component)  자식 컴포넌트 제거
void removeAll()  모든 자식 컴포넌트 제거
Component[] getComponents()  자식 컴포넌트 배열 리턴
Container getParent()  부모 컨테이너 리턴
Container getTopLevelAncestor()  최상위 부모 컨테이너 리턴
```

스윙 컴포넌트의 공통 기능, JComponent의 메소드

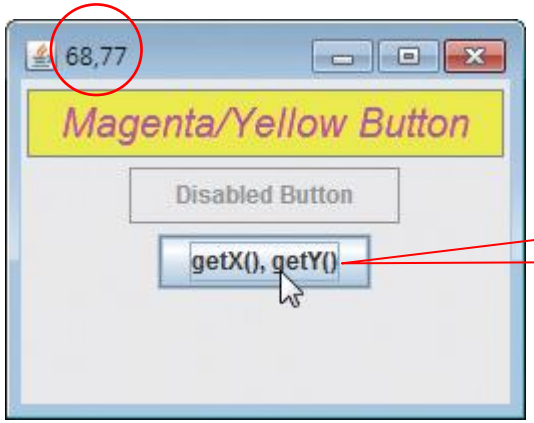
```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class JComponentEx extends JFrame {
    JComponentEx() {
        super("JComponent의 공통 메소드 예제");
        Container c = getContentPane();
        c.setLayout(new FlowLayout());
        JButton b1 = new JButton("Magenta/Yellow Button");
        JButton b2 = new JButton(" Disabled Button ");
        JButton b3 = new JButton("getX(), getY()");
        b1.setBackground(Color.YELLOW);
        b1.setForeground(Color.MAGENTA);
        b1.setFont(new Font("Arial", Font.ITALIC, 20));
        b2.setEnabled(false);
        b3.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                JButton b = (JButton)e.getSource();
                setTitle(b.getX() + "," + b.getY());
            }
        });
        c.add(b1); c.add(b2); c.add(b3);
        setSize(260,200); setVisible(true);
    }
    public static void main(String[] args) {
        new JComponentEx();
    }
}
```



"Arial" 로 20픽셀 크기의 폰트

비활성 버튼

초기 상태



클릭하면 타이틀바에 버튼의 좌표 출력

getX(), getY() 버튼이 클릭된 상태

JLabel로 문자열과 이미지 출력

- **JLabel의 용도**
  - 문자열이나 이미지를 화면에 출력하기 위한 목적

- **레이블 생성**

JLabel() 빈 레이블
JLabel(Icon image) 이미지 레이블
JLabel(String text) 문자열 레이블
JLabel(String text, Icon image, int hAlign) 문자열과 이미지 모두 가진 레이블
• hAlign: 수평 정렬 값으로 SwingConstants.LEFT, SwingConstants.RIGHT, SwingConstants.CENTER 중 하나

레이블 생성 예

- **문자열 레이블 생성**

```
JLabel textLabel = new JLabel("사랑합니다");
```

- **이미지 레이블 생성**
  - 이미지 파일로부터 이미지를 읽기 위해 ImageIcon 클래스 사용
  - 다룰 수 있는 이미지 : png, gif, jpg
    - sunset.jpg의 경로명이 "images/sunset.jpg"인 경우

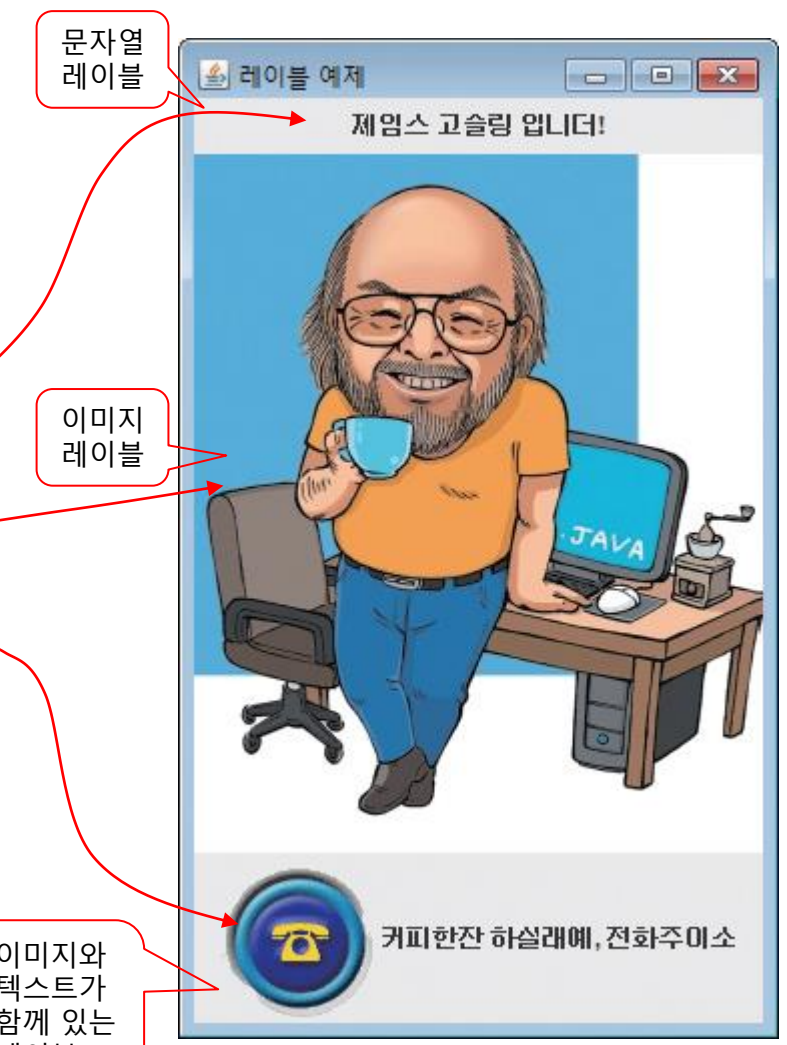
```
ImageIcon image = new ImageIcon("images/sunset.jpg");
JLabel imageLabel = new JLabel(image);
```

- **수평 정렬 값을 가진 레이블 컴포넌트 생성**
  - 수평 정렬로, 문자열과 이미지를 모두 가진 레이블

```
ImageIcon image = new ImageIcon("images/sunset.jpg");
JLabel label = new JLabel("사랑합니다", image, SwingConstants.CENTER);
```

JLabel을 이용한 레이블 만들기

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

public class LabelEx extends JFrame {
    LabelEx() {
        setTitle("레이블 예제");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container c = getContentPane();
        c.setLayout(new FlowLayout());

        JLabel textLabel = new JLabel("제임스 고슬링 입니더!");

        ImageIcon img = new ImageIcon("images/gosling.jpg");
        JLabel imageLabel = new JLabel(img);

        ImageIcon icon = new ImageIcon("images/icon.gif");
        JLabel label = new JLabel("커피한잔 하실래예, 전화주이소",
                                  icon, SwingConstants.CENTER);

        c.add(textLabel);       c.add(imageLabel);       c.add(label);
        setSize(300,500);
        setVisible(true);
    }
    public static void main(String [] args) {
        new LabelEx();
    }
}
```



JButton으로 버튼 만들기

- **JButton의 용도**
  - 버튼 모양의 컴포넌트. 사용자로부터 명령을 입력 받기 위한 목적
  - 버튼은 클릭될 때 Action 이벤트 발생

- **버튼 생성**

```
JButton() 빈 버튼
JButton(Icon image) 이미지 버튼
JButton(String text) 문자열 버튼
JButton(String text, Icon image) 문자열과 이미지 모두 가진 버튼
```

  - "hello" 문자열을 가진 버튼 생성 예

```
JButton btn = new JButton("hello");
```

이미지 버튼 만들기

- **하나의 버튼에 3 개의 이미지 등록**
  - 마우스 조작에 따라 3 개의 이미지 중 적절한 이미지 자동 출력

- **3 개의 버튼 이미지**
  - normalIcon
    - 버튼의 보통 상태(디폴트) 때 출력되는 이미지
    - 생성자에 이미지 아이콘 전달 혹은 JButton의 setIcon(normalIcon);
  - rolloverIcon
    - 버튼에 마우스가 올라갈 때 출력되는 이미지
    - 이미지 설정 메소드 : JButton의 setRolloverIcon(rolloverIcon);
  - pressedIcon
    - 버튼을 누른 상태 때 출력되는 이미지
    - 이미지 설정 메소드 : JButton의 setPressedIcon(pressedIcon)

이미지 버튼에 이미지 설정

- **이미지 로딩**
  - 필요한 이미지 로딩 : new ImageIcon(이미지 경로명);
  - 사례)

```
ImageIcon normalIcon = new ImageIcon("images/normalIcon.gif");
ImageIcon rolloverIcon = new ImageIcon("images/rolloverIcon.gif");
ImageIcon pressedIcon = new ImageIcon("images/pressedIcon.gif");
```

- **버튼에 이미지 등록**
  - JButton의 메소드를 호출하여 이미지 등록
  - 사례)

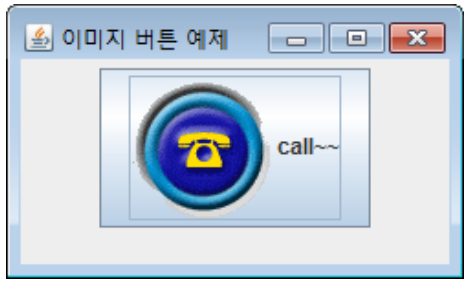
```
JButton button = new JButton("테스트버튼", normalIcon); // normalIcon 달기
button.setRolloverIcon(rolloverIcon);                   // rolloverIcon 달기
button.setPressedIcon(pressedIcon);                     // pressedIcon 달기
```

  - 실행 중에 normal 이미지(디폴트 이미지) 교체 사례

```
ImageIcon newIcon = new ImageIcon("images/newIcon.gif");
button.setIcon(newIcon);                                // 디폴트 이미지 변경
```

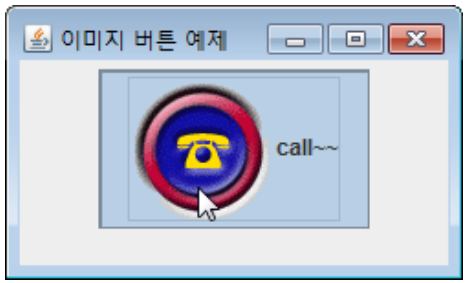
JButton을 이용한 이미지 버튼 만들기

그림과 같이 작동하는 이미지 버튼을 작성하라.



보통 상태에 있는 동안 (normalIcon.gif)

마우스가 버튼 위에 올라간 경우 (rolloverIcon.gif)

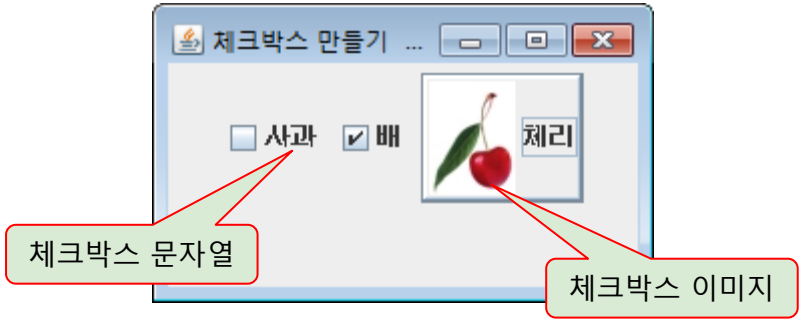


마우스가 눌러진 순간 (pressedIcon.gif)

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

public class ButtonImageEx extends JFrame {
    ButtonImageEx() {
        setTitle("이미지 버튼 예제");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container c = getContentPane();
        c.setLayout(new FlowLayout());

        ImageIcon normalIcon = new ImageIcon("images/normalIcon.gif");
        ImageIcon rolloverIcon = new ImageIcon("images/rolloverIcon.gif");
        ImageIcon pressedIcon = new ImageIcon("images/pressedIcon.gif");

        JButton btn = new JButton("call~~", normalIcon);
        btn.setPressedIcon(pressedIcon); // pressedIcon용 이미지 등록
        btn.setRolloverIcon(rolloverIcon); // rolloverIcon용 이미지 등록

        c.add(btn);
        setSize(250,150);
        setVisible(true);
    }
    public static void main(String [] args) {
        new ButtonImageEx();
    }
}
```

JCheckBox로 체크박스 만들기

- **JCheckBox의 용도**
  - 선택(selected)과 비선택(deselected) 두 상태만 가지는 버튼



- **체크박스 생성**

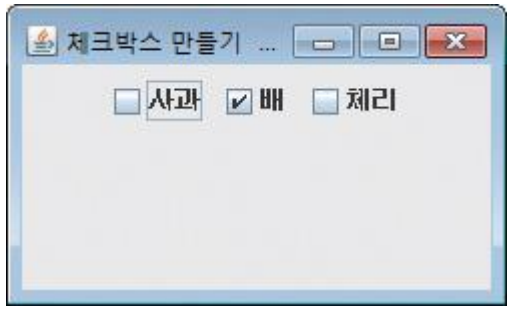
```
JCheckBox() 빈 체크박스
JCheckBox(Icon image) 이미지 체크박스
JCheckBox(Icon image, boolean selected) 이미지 체크박스
JCheckBox(String text, Icon image) 문자열과 이미지를 가진 체크박스
JCheckBox(String text, Icon image, boolean selected) 문자열과 이미지 체크박스
• selected: true면 선택 상태로 초기화
```

  - 문자열을 가진 체크박스 생성 예

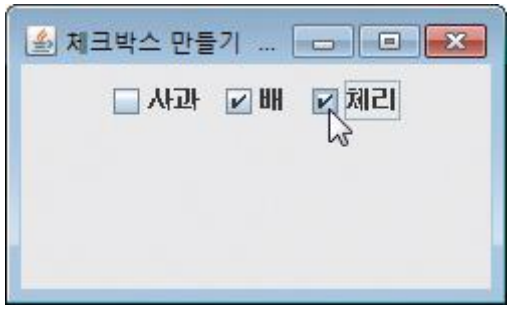
```
JCheckBox apple = new JCheckBox("사과"); // "사과" 체크박스 생성
JCheckBox pear = new JCheckBox("배", true); // 선택 상태의 "배" 체크박스 생성
```

JCheckBox로 체크박스 만들기

그림과 같은 3개의 문자열 체크박스를 가진 프로그램을 작성하라.



초기 상태



체리 체크박스를 선택한 상태

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

public class CheckBoxEx extends JFrame {
    CheckBoxEx() {
        setTitle("체크박스 만들기 예제");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container c = getContentPane();
        c.setLayout(new FlowLayout());

        // 3개의 체크박스를 생성한다.
        JCheckBox apple = new JCheckBox("사과");
        JCheckBox pear = new JCheckBox("배", true);
        JCheckBox cherry = new JCheckBox("체리");

        c.add(apple);
        c.add(pear);
        c.add(cherry);

        setSize(250,150);
        setVisible(true);
    }
    public static void main(String [] args) {
        new CheckBoxEx();
    }
}
```

선택 상태의 체크박스 생성

체크박스에 Item 이벤트 처리

- **Item 이벤트**
  - 체크 박스의 선택 상태에 변화가 생길 때 발생하는 이벤트
    - 사용자가 마우스나 키보드로 체크박스를 선택/해제할 때
    - 프로그램에서 체크박스를 선택/해제하여 체크 상태에 변화가 생길 때

```
JCheckBox c = new JCheckBox("사과");
c.setSelected(true); // 선택 상태로 변경
```

  - 이벤트가 발생하면 ItemEvent 객체 생성
  - ItemListener 리스너를 이용하여 이벤트 처리
- **ItemListener 리스너의 추상 메소드**

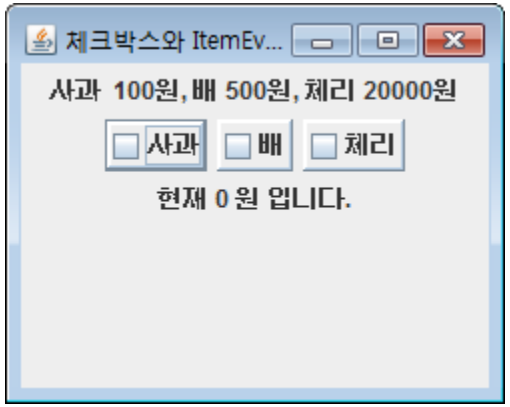
void itemStateChanged(ItemEvent e) 체크박스의 선택 상태가 변하는 경우 호출

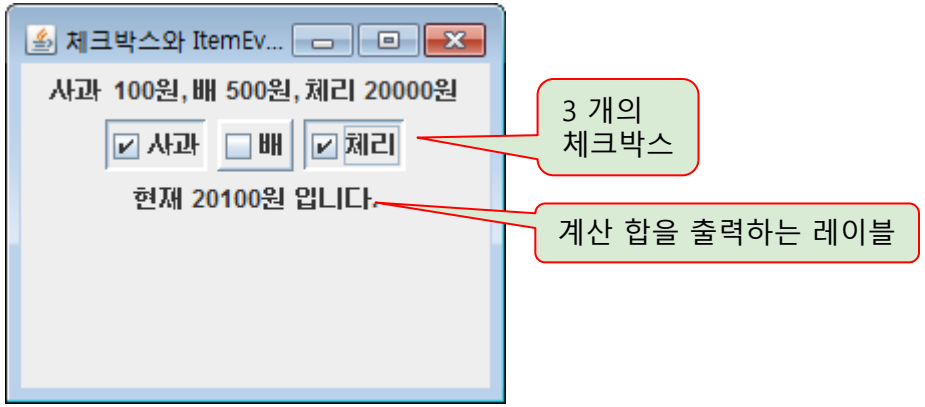
- **ItemEvent의 주요 메소드**

int getStateChange() 체크박스가 선택된 경우 ItemEvent.SELECTED를, 해제된 경우 ItemEvent.DESELECTED를 리턴한다.
Object getItem() 이벤트를 발생시킨 아이템 객체를 리턴한다. 체크박스의 경우 JCheckBox 컴포넌트의 레퍼런스를 리턴한다.

ItemEvent를 활용하여 체크박스로 가격 합산 응용

그림과 같이 사과, 배, 체리 체크박스를 만들고, 사용자가 과일을 선택하면 선택된 과일의 가격을 합산하여 출력하는 프로그램을 작성하라.





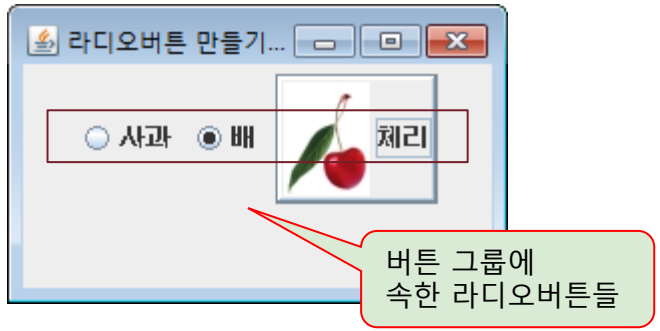
정답

```
import javax.swing.*;
import java.awt.event.*;
import java.awt.*;
public class CheckBoxItemEventEx extends JFrame {
    JCheckBox [] fruits = new JCheckBox [3];
    String [] names = {"사과", "배", "체리"};
    JLabel sumLabel;

    CheckBoxItemEventEx() {
        setTitle("체크박스와 ItemEvent 예제");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container c = getContentPane();
        c.setLayout(new FlowLayout());
        c.add(new JLabel("사과 100원, 배 500원, 체리 20000원"));
        MyItemListener listener = new MyItemListener();
        for(int i=0; i<fruits.length; i++) {
            fruits[i] = new JCheckBox(names[i]);
            fruits[i].setBorderPainted(true);
            c.add(fruits[i]);
            fruits[i].addItemListener(listener);
        }
        sumLabel = new JLabel("현재 0 원 입니다.");
        c.add(sumLabel);
        setSize(250,200);
        setVisible(true);
    }
```

```
    // Item 리스너 구현
    class MyItemListener implements ItemListener {
        int sum = 0; // 가격의 합
        public void itemStateChanged(ItemEvent e) {
            if(e.getStateChange() == ItemEvent.SELECTED) {
                if(e.getItem() == fruits[0])
                    sum += 100;
                else if(e.getItem() == fruits[1])
                    sum += 500;
                else
                    sum += 20000;
            }
            else {
                if(e.getItem() == fruits[0])
                    sum -= 100;
                else if(e.getItem() == fruits[1])
                    sum -= 500;
                else
                    sum -= 20000;
            }
            sumLabel.setText("현재 " + sum + "원 입니다.");
        }
    }
    public static void main(String [] args) {
        new CheckBoxItemEventEx();
    }
}
```

JRadioButton으로 라디오버튼 만들기

- **JRadioButton의 용도**
  - 버튼 그룹을 형성하고, 그룹에 속한 버튼 중 하나만 선택되는 라디오버튼
  - 체크박스와의 차이점
    - 체크 박스는 각각 선택/해제가 가능하지만,
      라디오버튼은 그룹에 속한 버튼 중 하나만 선택



- **라디오버튼 생성**

```
JRadioButton() 빈 라디오버튼
JRadioButton(Icon image) 이미지 라디오버튼
JRadioButton(Icon image, boolean selected) 이미지 라디오버튼
JRadioButton(String text) 문자열 라디오버튼
JRadioButton(String text, boolean selected) 문자열 라디오버튼
JRadioButton(String text, Icon image) 문자열과 이미지를 가진 라디오버튼
JRadioButton(String text, Icon image, boolean selected) 문자열과 이미지를 가진 라
    디오버튼
• selected: true면 선택 상태로 초기화
```

라디오버튼 생성 및 Item 이벤트 처리

- **버튼 그룹과 라디오버튼 생성 과정**

1. 버튼 그룹 객체 생성 → `ButtonGroup group = new ButtonGroup();`

2. 라디오버튼 생성 →
```
JRadioButton apple= new JRadioButton("사과");
JRadioButton pear= new JRadioButton("배");
JRadioButton cherry= new JRadioButton("체리");
```

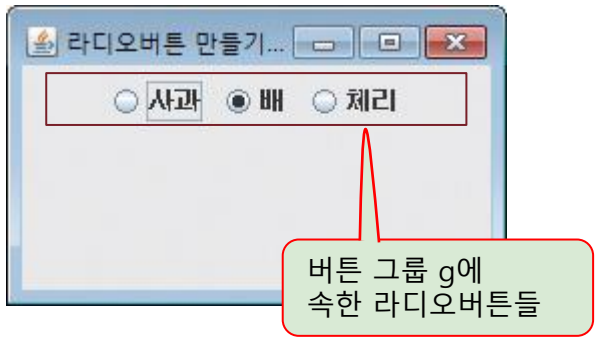
3. 라디오버튼을 버튼 그룹에 삽입 →
```
group.add(apple);
group.add(pear);
group.add(cherry);
```

4. 라디오버튼을 컨테이너에 삽입 →
```
container.add(apple);
container.add(pear);
container.add(cherry);
```

- **라디오버튼에 Item 이벤트 처리 : ItemListener 리스너 이용**
  - 라디오버튼이 선택/해제되어 상태가 달라지면, Item 이벤트 발생
    - 사용자가 마우스나 키보드로 선택 상태를 변경할 때
    - 프로그램에서 JRadioButton의 setSelected()를 호출하여 선택 상태를 변경할 때

JRadioButton으로 라디오버튼 만들기

그림과 같이 3개의 라디오버튼을 가진 프로그램을 작성하라.




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

public class RadioButtonEx extends JFrame {
    RadioButtonEx() {
        setTitle("라디오버튼 만들기 예제");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container c = getContentPane();
        c.setLayout(new FlowLayout());
        ButtonGroup g = new ButtonGroup(); // 버튼 그룹 객체 생성

        JRadioButton apple = new JRadioButton("사과");
        JRadioButton pear = new JRadioButton("배", true);
        JRadioButton cherry = new JRadioButton("체리");

        // 버튼 그룹에 3개의 라디오버튼 삽입
        g.add(apple);
        g.add(pear);
        g.add(cherry);

        // 컨텐트팬에 3개의 라디오버튼 삽입
        c.add(apple); c.add(pear); c.add(cherry);
        setSize(250,150);
        setVisible(true);
    }
    public static void main(String [] args) {
        new RadioButtonEx();
    }
}
```

JTextField로 한 줄 입력 창 만들기

- **JTextField**
  - 한 줄의 문자열을 입력 받는 창(텍스트필드)
    - 텍스트 입력 도중 <Enter>키가 입력되면 Action 이벤트 발생
    - 입력 가능한 문자 개수와 입력 창의 크기는 서로 다름

- **텍스트필드 생성**

```
JTextField() 빈 텍스트필드
JTextField(int cols) 입력 창의 열의 개수가 cols개인 텍스트필드
JTextField(String text) text 문자열로 초기화된 텍스트필드
JTextField(String text, int cols) 입력 창의 열의 개수는 cols개이고 text 문자열로 초기
    화된 텍스트필드
```

  - "소프트웨어학과"로 초깃값을 가지는 텍스트필드 생성 예

```
JTextField tf2 = new JTextField("소프트웨어학과");
```

JTextField로 텍스트필드 만들기

JTextField를 이용하여 그림과 같이 이름, 학과, 주소를 입력받는 폼을 만들어라.
입력 창의 열의 개수는 모두 20으로 한다.

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

public class TextFieldEx extends JFrame {
    TextFieldEx() {
        setTitle("텍스트필드 만들기 예제");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container c = getContentPane();
        c.setLayout(new FlowLayout());

        c.add(new JLabel("이름 "));
        c.add(new JTextField(20));
        c.add(new JLabel("학과 "));
        c.add(new JTextField( " 소프트웨어학과", 20));
        c.add(new JLabel("주소 "));
        c.add(new JTextField("서울시 ...", 20));

        setSize(300,150);
        setVisible(true);
    }
    public static void main(String [] args) {
        new TextFieldEx();
    }
}
```

TextArea로 여러 줄의 입력 창 만들기

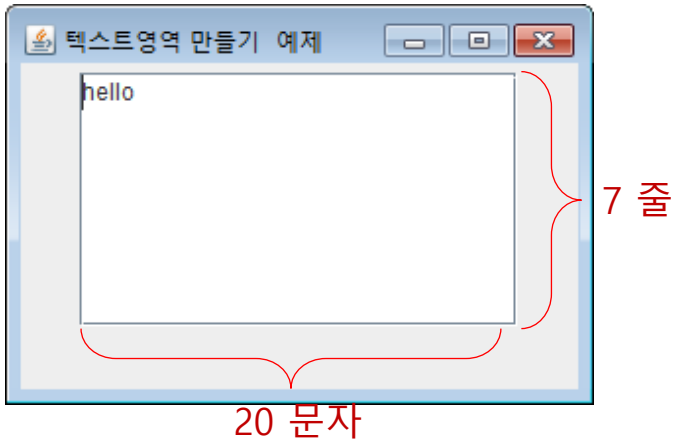
- **JTextArea**
  - 여러 줄의 문자열을 입력받을 수 있는 창(텍스트영역)
    - 스크롤바를 지원하지 않는다.
    - JScrollPane 객체에 삽입하여 스크롤바 지원받음

- **생성자**

```
JTextArea() 빈 텍스트영역
JTextArea(int rows, int cols) 입력 창이 rows × cols개의 문자 크기인 텍스트영역
JTextArea(String text) text 문자열로 초기화된 텍스트영역
JTextArea(String text, int rows, int cols) 입력 창이 rows × cols개의 문자 크기이며
    text 문자열로 초기화된 텍스트영역
```

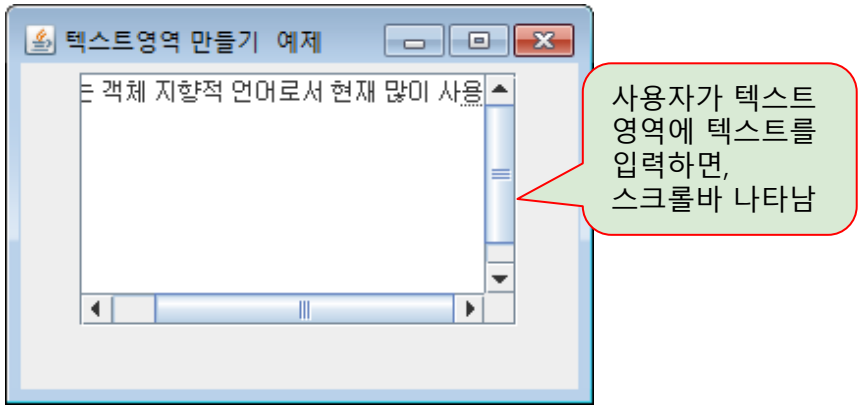
텍스트영역 생성 예

"hello" 문자열의 초깃값을 가지고
한 줄에 20개의 문자가 입력가능하며,
7줄로 구성된 텍스트 영역 만들기




```
JTextArea ta = new JTextArea("hello", 7, 20);
container.add(ta);
```

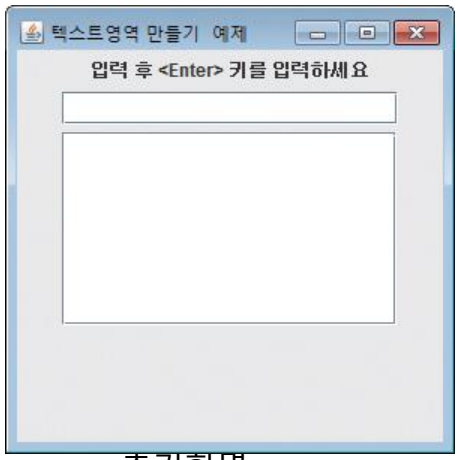
왼쪽에 만든 텍스트영역에
스크롤바 붙이기



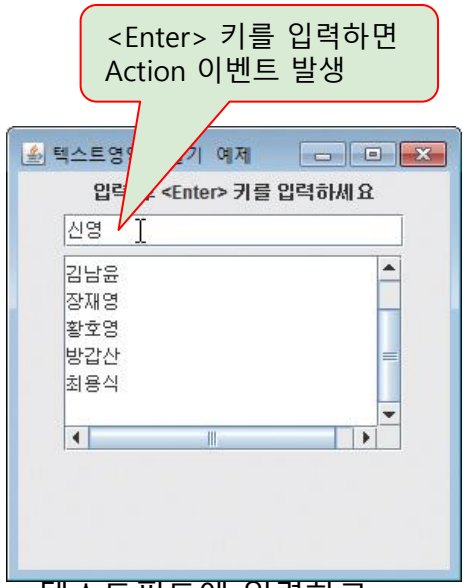

```
JTextArea ta = new JTextArea("hello", 7, 20);
container.add(new JScrollPane(ta));
```

JTextArea로 여러 줄이 입력되는 창 만들기

그림과 같이 텍스트필드에 문자열을 입력한 후 <Enter> 키를 입력하면 텍스트영역 창에 문자열을 추가하고 텍스트필드 입력 창은 지우는 프로그램을 작성하라.



초기화면



텍스트필드에 입력하고
<Enter> 키를 누른 경우

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

public class TextAreaEx extends JFrame {
    JTextField tf = new JTextField(20);
    JTextArea ta = new JTextArea(7, 20);

    TextAreaEx() {
        setTitle("텍스트영역 만들기 예제");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container c = getContentPane();
        c.setLayout(new FlowLayout());
        c.add(new JLabel("입력 후 <Enter> 키를 입력하세요"));
        c.add(tf);
        c.add(new JScrollPane(ta));

        tf.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                JTextField t = (JTextField)e.getSource();
                ta.append(t.getText() + "₩n");
                t.setText("");
            }
        });
        setSize(300,300);
        setVisible(true);
    }
    public static void main(String [] args) {
        new TextAreaEx();
    }
}
```

JList로 리스트 만들기

- **JList**
  - 하나 이상의 아이템을 보여주고 아이템을 선택하도록 하는 리스트
  - JScrollPane에 JList 컴포넌트를 삽입하여야 스크롤 가능

- **리스트 생성**

```
JList() 빈 리스트
JList(Vector listData) 벡터로부터 아이템을 공급받는 리스트
JList(Object [] listData) 배열로부터 아이템을 공급받는 리스트
```

  - 예) 9개의 과일 이름 문자열이 든 리스트 만들기

```
String [] fruits= {"apple", "banana", "kiwi", "mango", "pear",
                   "peach", "berry", "strawberry", "blackberry"};
JList strList = new JList(fruits);
```

JList로 다양한 리스트 만들기

그림과 같은 3개의 리스트를 가진 프로그램을 작성하라.


```
import javax.swing.*;
import java.awt.*;
public class ListEx extends JFrame {
   String [] fruits= {"apple", "banana", "kiwi", "mango", "pear", "peach",
                      "berry", "strawberry", "blackberry"};
   ImageIcon [] images = { new ImageIcon("images/icon1.png"),
                           new ImageIcon("images/icon2.png"),
                           new ImageIcon("images/icon3.png"),
                           new ImageIcon("images/icon4.png") };
   ListEx() {
      setTitle("리스트 만들기 예제");
      setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      Container c = getContentPane();
      c.setLayout(new FlowLayout());

      JList strList = new JList(fruits);
      c.add(strList);

      JList imageList = new JList();
      imageList.setListData(images);
      c.add(imageList);

      JList scrollList = new JList(fruits);
      c.add(new JScrollPane(scrollList));

      setSize(300,300); setVisible(true);
   }
   public static void main(String [] args) {
      new ListEx();
   }
}
```

JComboBox로 콤보박스 만들기

- **JComboBox**
  - 텍스트필드와 버튼, 그리고 드롭다운 리스트로 구성되는 콤보박스
  - 드롭다운 리스트에서 선택한 것이 텍스트필드에 나타남

- **콤보박스 생성**

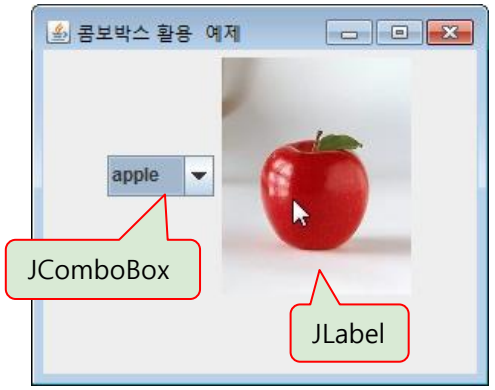
```
JComboBox() 빈 콤보박스
JComboBox(Vector items) 벡터로부터 아이템을 공급받는 콤보박스
JComboBox(Object [] items) 배열로부터 아이템을 공급받는 콤보박스
```

  - 예) 텍스트를 아이템으로 가진 콤보박스 생성

```
String [] fruits = {"apple", "banana", "kiwi",
                    "mango", "pear", "peach",
                    "berry", "strawberry", "blackberry" };
JComboBox combo = new JComboBox(fruits);
```

JComboBox로 콤보박스 만들고 활용하기

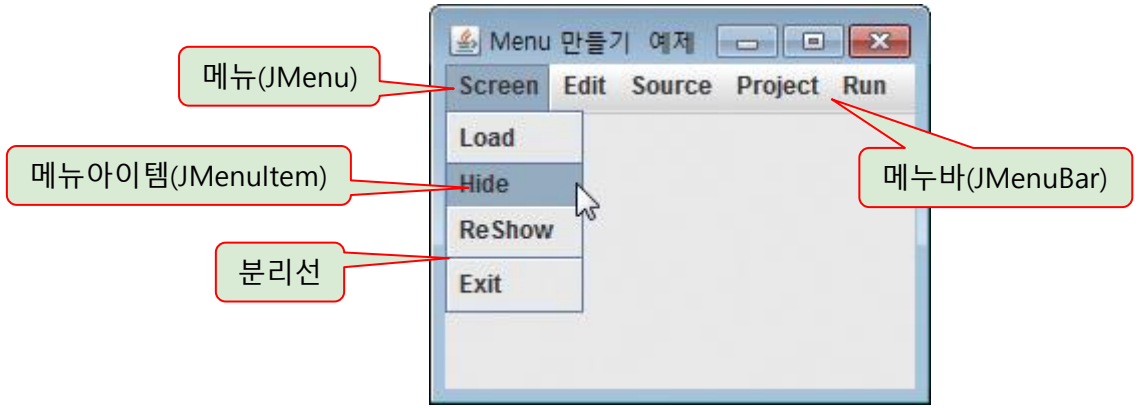
그림과 같이 "apple", "babana", "mango"의 과일 이름을 가진 콤보박스를 만들고 사용자가 선택한 과일의 이미지를 콤보박스 옆에 출력하는 프로그램을 작성하라.



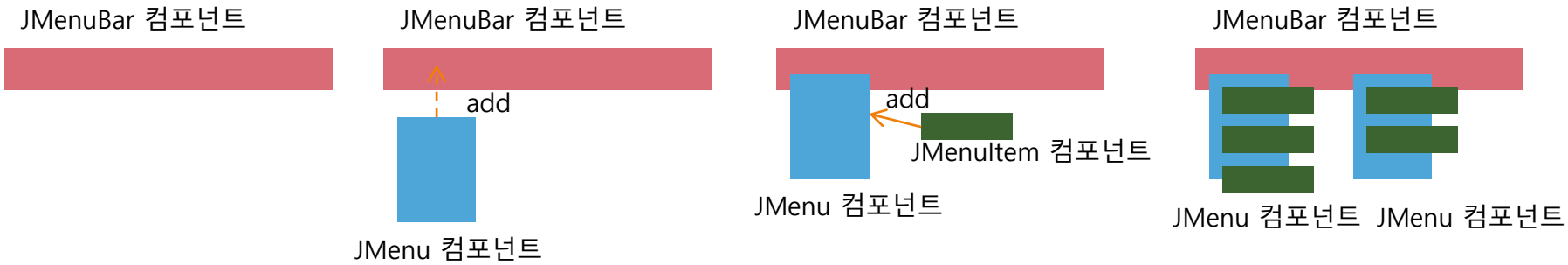

```
import javax.swing.*;
import java.awt.event.*;
import java.awt.*;

public class ComboActionEx extends JFrame {
    String [] fruits = {"apple", "banana", "mango"};
    ImageIcon [] images = { new ImageIcon("images/apple.jpg"),
                            new ImageIcon("images/banana.jpg"),
                            new ImageIcon("images/mango.jpg") };
    JLabel imgLabel = new JLabel(images[0]);

    ComboActionEx() {
        setTitle("콤보박스 활용 예제");
        Container c = getContentPane();
        c.setLayout(new FlowLayout());
        JComboBox combo = new JComboBox(fruits);
        c.add(combo); c.add(imgLabel);

        // 콤보박스에 Action 리스너 등록. 선택된 아이템의 이미지 출력
        combo.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                JComboBox cb = (JComboBox)e.getSource();
                int index = cb.getSelectedIndex();
                imgLabel.setIcon(images[index]);
            }
        });
        setSize(300,250);
        setVisible(true);
    }
    public static void main(String [] args) {
        new ComboActionEx();
    }
}
```

메뉴 구성

- **메뉴 만들기에 필요한 스윙 컴포넌트**
  - 메뉴아이템 – JMenuItem
  - 메뉴 – JMenu
    - 여러 개의 메뉴 아이템을 가짐
  - 메뉴바 – JMenuBar
    - 여러 개의 메뉴를 붙이는 바이며, 프레임에 부착됨
  - 분리선
    - 메뉴아이템 사이의 분리선으로 separator라고 부름
    - JMenu의 addSeparator()를 호출하여 삽입함



메뉴 만드는 과정

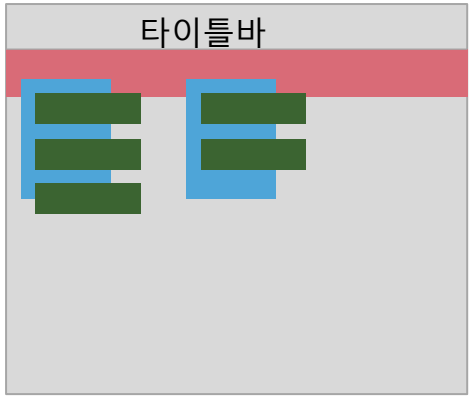


(1) JMenuBar 컴포넌트 생성

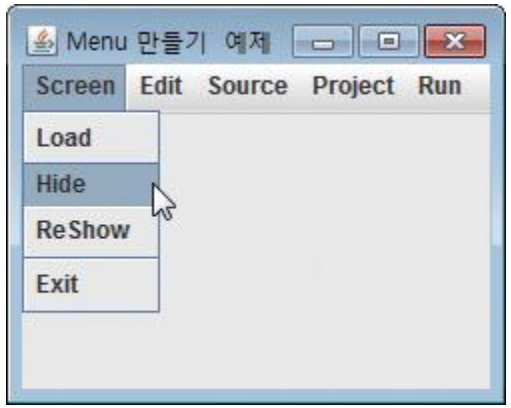
(2) JMenu 컴포넌트를 생성하여 JMenuBar에 붙인다.

(3) JMenuItem 컴포넌트를 생성하여 JMenu에 붙인다.

(3′) 여러 개의 메뉴와 메뉴 아이템을 생성한다.



(4) JMenuBar 컴포넌트를 JFrame에 붙인다.



(1) →
```
JMenuBar mb = new JMenuBar();
```

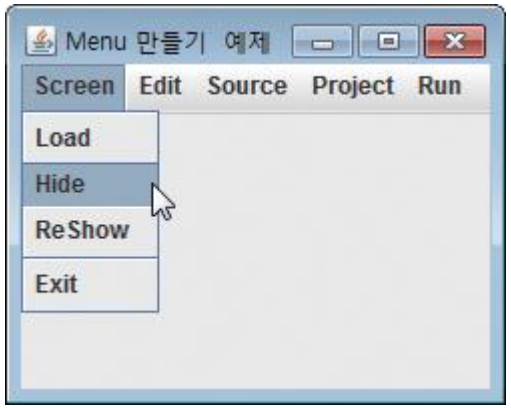
(2) →
```
JMenu screenMenu = new JMenu("Screen");
mb.add(screenMenu);
```

(3) →
```
screenMenu.add(new JMenuItem("Load"));
screenMenu.add(new JMenuItem("Hide"));
screenMenu.add(new JMenuItem("ReShow"));
screenMenu.addSeparator();
screenMenu.add(new JMenuItem("Exit"));
```

(4) →
```
frame.setJMenuBar(mb);
```

메뉴 만들기

그림과 같이 Screen, Edit, Source, Project, Run 의
5개 메뉴를 가지며, Screen 메뉴에만 4개의 메뉴아이템과 분리선(separator)을 가지도록 프로그램을 작성하라.




```
import javax.swing.*;

public class MenuEx extends JFrame {
    MenuEx() {
        setTitle("Menu 만들기 예제");
        createMenu(); // 메뉴 생성, 프레임에 삽입
        setSize(250,200);
        setVisible(true);
    }

    void createMenu() {
        JMenuBar mb = new JMenuBar();
        JMenu screenMenu = new JMenu("Screen");

        screenMenu.add(new JMenuItem("Load"));
        screenMenu.add(new JMenuItem("Hide"));
        screenMenu.add(new JMenuItem("ReShow"));
        screenMenu.addSeparator();
        screenMenu.add(new JMenuItem("Exit"));

        mb.add(screenMenu);
        mb.add(new JMenu("Edit"));
        mb.add(new JMenu("Source"));
        mb.add(new JMenu("Project"));
        mb.add(new JMenu("Run"));
        setJMenuBar(mb);
    }
    public static void main(String [] args) {
        new MenuEx();
    }
}
```

메뉴바를 프레임에 붙임.
비로소 메뉴가 보인다.

메뉴아이템에 Action 이벤트 달기

- **메뉴아이템을 클릭하면 Action 발생**
  - 메뉴아이템은 사용자로부터의 지시나 명령을 받는데 사용
  - ActionListener 인터페이스로 리스너 작성
  - 각 메뉴아이템마다 이벤트 리스너 설정

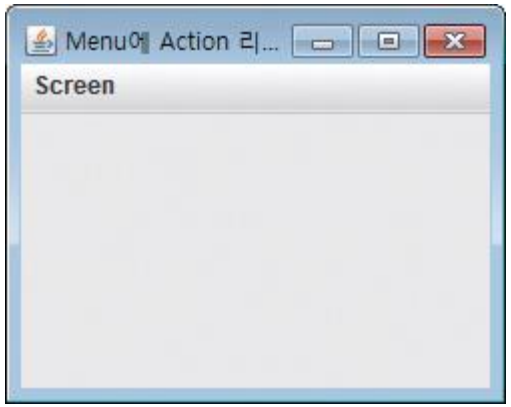
  - 예) Load 메뉴아이템에 Action 리스너를 작성하는 경우

```
JMenuItem item = new JMenuItem("Load");
item.addActionListener(new MenuActionListener()); // 메뉴아이템에 Action 리스너 설정
screenMenu.add(item);

class MenuActionListener implements ActionListener {
   public void actionPerformed(ActionEvent e) {
      // 사용자가 Load 메뉴아이템을 선택하는 경우 처리할 작업 구현
      ...
   }
}
```

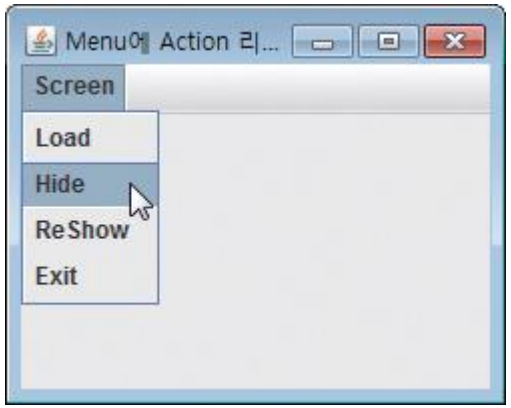
메뉴에 Action 리스너 활용

그림과 같이 Screen 메뉴에 4개의 메뉴아이템을 만들고, Load 메뉴아이템을 선택하면 이미지를 하나 로딩하여 출력하고, Hide 메뉴아이템을 선택하면 이미지를 보이지 않게 하며, ReShow 메뉴아이템을 선택하면 숨겨진 이미지를 다시 보이게 하고, Exit 메뉴아이템을 선택하면 프로그램을 종료하도록 Action 리스너를 작성하라.

초기 상태



Load 메뉴아이템 선택하면 레이블에 이미지 출력



Hide 메뉴아이템 선택으로 이미지를 보이지 않게 함

ReShow 메뉴아이템 선택으로 숨겨진 이미지가 다시 보이게 함

Exit 메뉴아이템 선택하면 프로그램 종료

정답

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

public class MenuActionEventEx extends JFrame {
    JLabel imgLabel = new JLabel(); // 빈 레이블

    MenuActionEventEx() {
        setTitle("Menu에 Action 리스너 만들기 예제");
        createMenu();
        getContentPane().add(imgLabel, BorderLayout.CENTER);
        setSize(250,200); setVisible(true);
    }

    void createMenu() {
        JMenuBar mb = new JMenuBar(); // 메뉴바 생성
        JMenuItem [] menuItem = new JMenuItem [4];
        String[] itemTitle = {"Load", "Hide", "ReShow", "Exit"};
        JMenu screenMenu = new JMenu("Screen");

        MenuActionListener listener = new MenuActionListener();
        for(int i=0; i<menuItem.length; i++) {
            menuItem[i] = new JMenuItem(itemTitle[i]);
            menuItem[i].addActionListener(listener);
            screenMenu.add(menuItem[i]);
        }
        mb.add(screenMenu);
        setJMenuBar(mb); // 메뉴바를 프레임에 부착
    }
```

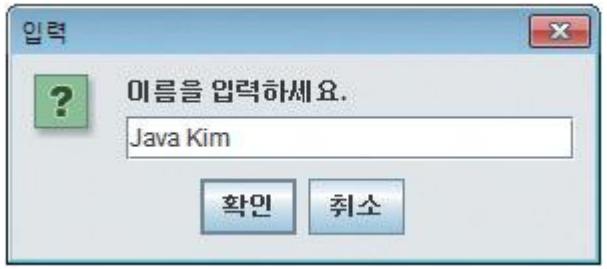
```
    class MenuActionListener implements ActionListener {
        public void actionPerformed(ActionEvent e) {
            String cmd = e.getActionCommand();
            switch(cmd) { // 메뉴 아이템의 종류 구분
                case "Load" :
                    if(imgLabel.getIcon() != null) return;
                                        // 이미 로딩되었으면 리턴
                    imgLabel.setIcon(new
                                ImageIcon("images/img.jpg"));
                    break;
                case "Hide" :
                    imgLabel.setVisible(false); break;
                case "ReShow" :
                    imgLabel.setVisible(true); break;
                case "Exit" :
                    System.exit(0); break;
            }
        }
    }

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

팝업 다이얼로그, JOptionPane

- **팝업 다이얼로그**
  - 사용자에게 메시지를 전달하거나 문자열을 간단히 입력받는 용도
  - JOptionPane 클래스를 이용하여 생성
    - static 타입의 간단한 메소드 이용

- **입력 다이얼로그 - JOptionPane.showInputDialog()**
  - 한 줄을 입력 받는 다이얼로그

```
static String JOptionPane.showInputDialog(String msg)
```
- *msg* : 다이얼로그 메시지
- 리턴 값 : 사용자가 입력한 문자열. 취소 버튼이 선택되거나 창이 닫히면 null 리턴



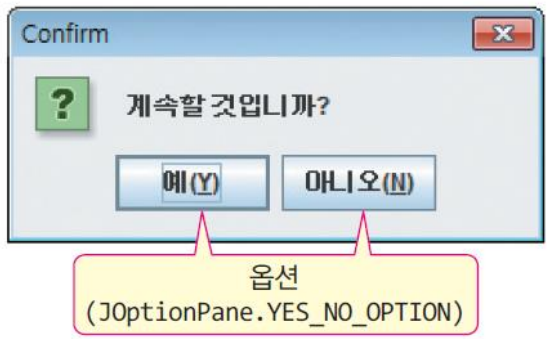

```
String name = JOptionPane.showInputDialog("이름을 입력하세요.");
// name에 "Java Kim"이 리턴
// 취소 버튼이나, 입력 없이 다이얼로그가 닫히면 null 리턴
```

확인 다이얼로그

- **확인 다이얼로그 - JOptionPane.showConfirmDialog()**
  - 사용자로부터 Yes/No 응답을 입력 받는 다이얼로그

```
static int JOptionPane.showConfirmDialog(Component parentComponent, Object msg,
                                         String title, int optionType)
```

- parentComponent : 다이얼로그의 부모 컴포넌트로서 다이얼로그가 출력되는 영역의 범위 지정을 위해 사용(예: 프레임). null이면 전체 화면 중앙에 출력
- msg : 다이얼로그 메시지
- title : 다이얼로그 타이틀
- optionType : 다이얼로그 옵션 종류 지정
  YES_NO_OPTION, YES_NO_CANCEL_OPTION, OK_CANCEL_OPTION
- 리턴 값 : 사용자가 선택한 옵션 종류
  YES_OPTION, NO_OPTION, CANCEL_OPTION, OK_OPTION, CLOSED_OPTION



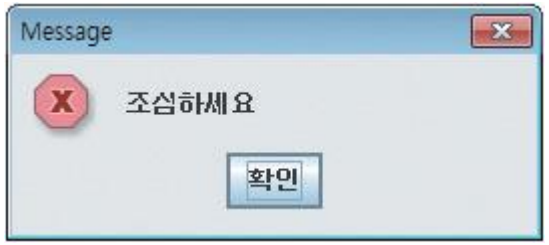

```
int result = JOptionPane.showConfirmDialog(null, "계속할 것입니까?",
                                           "Confirm", JOptionPane.YES_NO_OPTION);
if(result == JOptionPane.CLOSED_OPTION) {
    // 사용자가 "예", "아니오"의 선택 없이 다이얼로그 창을 닫은 경우
}
else if(result == JOptionPane.YES_OPTION) {
    // 사용자가 "예"를 선택한 경우
}
else {
    // 사용자가 "아니오"를 선택한 경우
}
```

메시지 다이얼로그

- **메시지 다이얼로그 – showMessageDialog**
  - 단순 메시지를 출력하는 다이얼로그

```
static void JOptionPane.showMessageDialog(Component parentComponent,
                                          Object msg, String title, int messageType)
```

- parentComponent : 다이얼로그의 부모 컴포넌트로서 다이얼로그가 출력되는 영역의 범위 지정을 위해 사용(예: 프레임). null이면 전체 화면 중앙에 출력
- msg : 다이얼로그 메시지
- title : 다이얼로그 타이틀
- messageType : 다이얼로그의 종류로서 다음 중 하나
  ERROR_MESSAGE, INFORMATION_MESSAGE, WARNING_MESSAGE, QUESTION_MESSAGE, PLAIN_MESSAGE

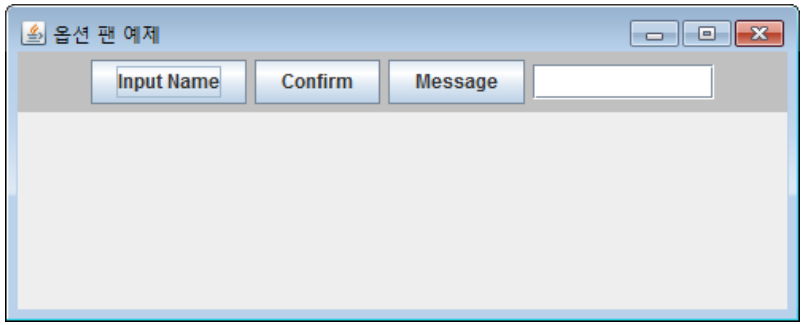



```
JOptionPane.showMessageDialog(null,
                "조심하세요", "Message",
                JOptionPane.ERROR_MESSAGE);
```

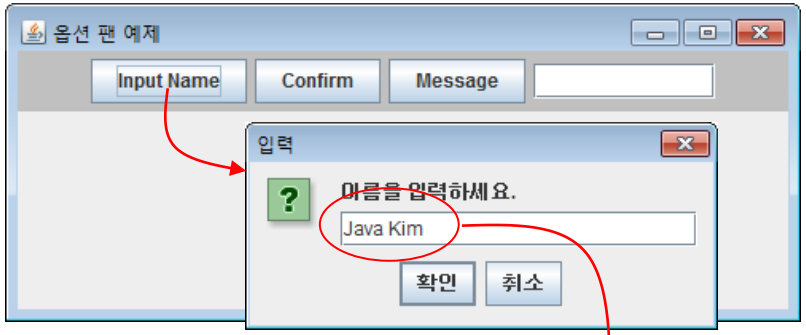
JOptionPane으로 3가지 팝업 다이얼로그 만들기

다음 그림과 같이
3개의 팝업 다이얼로그를
출력하는 응용프로그램을
작성해보라

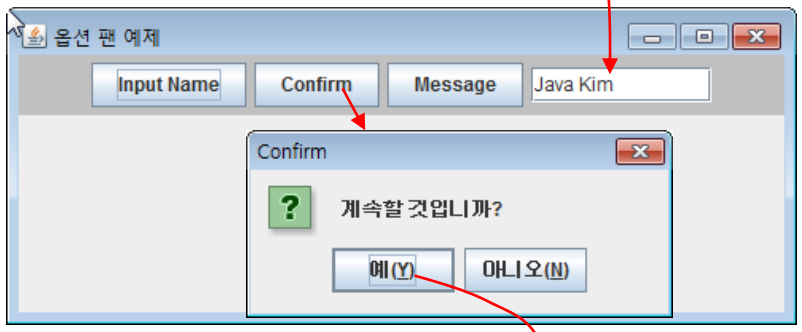
초기 화면



Input Name 버튼을 누르면
입력 다이얼로그 생성.
"Java Kim"을 입력하고
확인 버튼을 누르면
텍스트필드 창에 출력



Confirm 버튼을 누르면
확인 다이얼로그 생성
"예" 버튼을 누르면
텍스트필드 창에 "Yes" 출력



Message 버튼을 누르면
메시지 다이얼로그 생성
"확인" 버튼을 누르면
다이얼로그 종료

정답

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

public class OptionPaneEx extends JFrame {
    Container contentPane;

    OptionPaneEx() {
        setTitle("옵션 팬 예제");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        contentPane = getContentPane();
        setSize(500,200);
        contentPane.add(new MyPanel(),
                        BorderLayout.NORTH);
        setVisible(true);
    }

    class MyPanel extends Panel {
        JButton inputBtn = new JButton("Input Name");
        JTextField tf = new JTextField(10);
        JButton confirmBtn = new JButton("Confirm");
        JButton messageBtn = new JButton("Message");

        MyPanel() {
            setBackground(Color.LIGHT_GRAY);
            add(inputBtn);
            add(confirmBtn);
            add(messageBtn);
            add(tf);
            inputBtn.addActionListener(new ActionListener() {
                public void actionPerformed(ActionEvent e) {
                    String name =
                        JOptionPane.showInputDialog("이름을 입력하세요.");
                    if(name != null)
                        tf.setText(name);
                }
            });

            confirmBtn.addActionListener(new ActionListener() {
                public void actionPerformed(ActionEvent e) {
                    int result = JOptionPane.showConfirmDialog(null,
                                "계속할 것입니까?", "Confirm",
                                JOptionPane.YES_NO_OPTION);
                    if(result == JOptionPane.CLOSED_OPTION)
                        tf.setText("Just Closed without Selection");
                    else if(result == JOptionPane.YES_OPTION)
                        tf.setText("Yes");
                    else
                        tf.setText("No");
                }
            });

            messageBtn.addActionListener(new ActionListener() {
                public void actionPerformed(ActionEvent e) {
                    JOptionPane.showMessageDialog(null,
                                "조심하세요", "Message",
                                JOptionPane.ERROR_MESSAGE);
                }
            });
        }
    }

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