

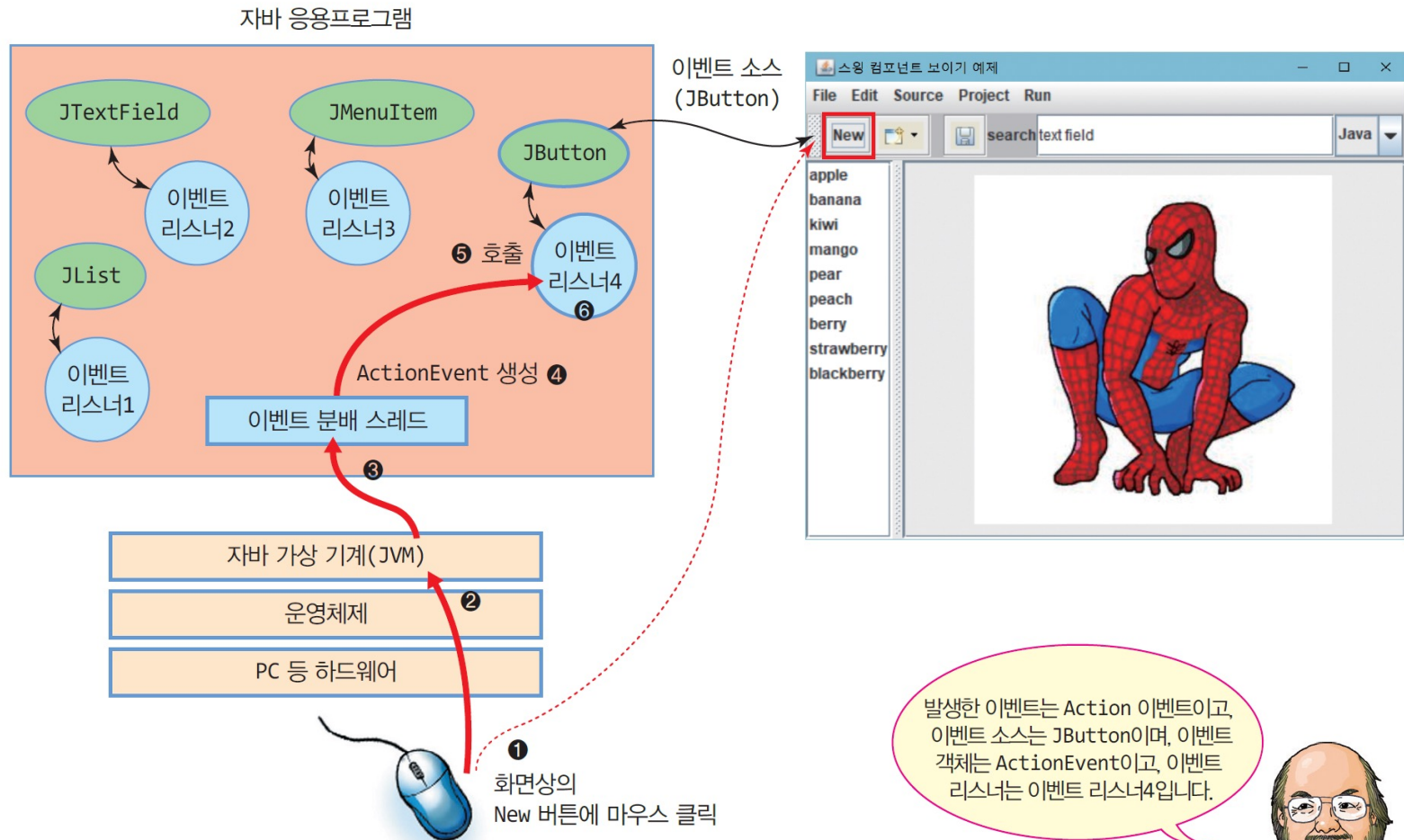
(INTERMEDIATE) JAVA PROGRAMMING

17. SWING – Mouse Event Handling
Chapter 10

JAVA:

REVIEW: EVENT HANDLING

Review: 자바의 이벤트 기반 GUI 구성



Review: 이벤트 관련 용어

- 이벤트 소스
 - 이벤트를 발생시킨 GUI 컴포넌트
- 이벤트 객체
 - 발생한 이벤트에 대한 정보
 - 예) 이벤트 종류, 이벤트 소스, 화면 좌표, 마우스 버튼 종류, 눌려진 키
- 이벤트 리스너
 - 이벤트를 처리하는 코드
 - 컴포넌트에 등록되어야 작동 가능
- 이벤트 분배 스레드
 - 동작
 - 자바 가상 기계로부터 이벤트의 발생을 통지 받음
 - 이벤트 소스와 이벤트 종류 결정
 - 적절한 이벤트 객체 생성, 이벤트 리스너를 찾아 호출
 - 무한 루프를 실행하는 스레드

Review: 이벤트 객체와 이벤트 소스

이벤트 객체	이벤트 소스	이벤트가 발생하는 경우
ActionEvent	JButton	마우스나 <Enter> 키로 버튼 선택
	JMenuItem	메뉴 아이템 선택
	TextField	텍스트 입력 중 <Enter> 키 입력
ItemEvent	JCheckBox	체크박스의 선택 혹은 해제
	JRadioButton	라디오버튼의 선택 상태가 변할 때
	JCheckBoxMenuItem	체크박스 메뉴 아이템의 선택 혹은 해제
ListSelectionEvent	JList	리스트에서 선택된 아이템이 변경될 때
KeyEvent	Component	키가 눌러지거나 눌러진 키가 떼어질 때
MouseEvent	Component	마우스 버튼이 눌러지거나 떼어질 때, 마우스 버튼이 클릭될 때, 컴포넌트 위에 마우스가 올라갈 때, 올라간 마우스가 내려올 때, 마우스가 드래그될 때, 마우스가 단순히 움직일 때
FocusEvent	Component	컴포넌트가 포커스를 받거나 잃을 때
WindowEvent	Window	Window를 상속받는 모든 컴포넌트에 대해 윈도우 활성화, 비활성화, 아이콘화, 아이콘에서 복구, 윈도우 열기, 윈도우 닫기, 윈도우 종료
AdjustmentEvent	JScrollBar	스크롤바를 움직일 때
ComponentEvent	Component	컴포넌트가 사라지거나, 나타나거나, 이동, 크기 변경 시
ContainerEvent	Container	Container에 컴포넌트의 추가 혹은 삭제

Review: 이벤트 리스너(Event Listener)

- 이벤트 리스너란?
 - 이벤트를 처리하는 코드, 클래스로 작성
 - JDK에서 이벤트 리스너 작성을 위한 인터페이스(interface) 제공
 - 개발자가 리스너 인터페이스의 추상 메소드 구현
 - 이벤트가 발생하면 자바 플랫폼은 리스너 인터페이스의 추상 메소드 호출
 - 예) ActionListener 인터페이스

```
interface ActionListener { // 아래 메소드를 개발자가 구현해야 함
    public void actionPerformed(ActionEvent e); // Action 이벤트 발생시 호출됨
}
```

- 예) MouseListener 인터페이스

```
interface MouseListener { // 아래의 5개 메소드를 개발자가 구현해야 함
    public void mousePressed(MouseEvent e); // 마우스 버튼이 눌러지는 순간 호출
    public void mouseReleased(MouseEvent e); // 눌러진 마우스 버튼이 떼어지는 순간 호출
    public void mouseClicked(MouseEvent e); // 마우스가 클릭되는 순간 호출
    public void mouseEntered(MouseEvent e); // 마우스가 컴포넌트 위에 올라가는 순간 호출
    public void mouseExited(MouseEvent e); // 마우스가 컴포넌트 위에서 내려오는 순간 호출
}
```

리스너 인터페이스와 메소드

이벤트 종류	리스너 인터페이스	리스너의 추상 메소드	메소드가 호출되는 경우
Action	ActionListener	void actionPerformed(ActionEvent)	Action 이벤트가 발생하는 경우
Item	ItemListener	void itemStateChanged(ItemEvent)	Item 이벤트가 발생하는 경우
Key	KeyListener	void keyPressed(KeyEvent)	모든 키에 대해 키가 눌릴 때
		void keyReleased(KeyEvent)	모든 키에 대해 눌려진 키가 떼어질 때
		void keyTyped(KeyEvent)	유니코드 키가 입력될 때
Mouse	MouseListener	void mousePressed(MouseEvent)	마우스 버튼이 눌릴 때
		void mouseReleased(MouseEvent)	눌려진 마우스 버튼이 떼어질 때
		void mouseClicked(MouseEvent)	마우스 버튼이 클릭될 때
		void mouseEntered(MouseEvent)	마우스가 컴포넌트 위에 올라올 때
		void mouseExited(MouseEvent)	컴포넌트 위에 올라온 마우스가 컴포넌트를 벗어날 때
Mouse	MouseMotionListener	void mouseDragged(MouseEvent)	마우스를 컴포넌트 위에서 드래그할 때
		void mouseMoved(MouseEvent)	마우스가 컴포넌트 위에서 움직일 때
Focus	FocusListener	void focusGained(FocusEvent)	컴포넌트가 포커스를 받을 때
		void focusLost(FocusEvent)	컴포넌트가 포커스를 잃을 때
ListSelection	ListSelectionListener	void valueChanged(ListSelectionEvent)	JList에 선택된 아이템이 변경될 때
Window	WindowListener	void windowOpened(WindowEvent)	윈도우가 생성되어 처음으로 보이게 될 때
		void windowClosing(WindowEvent)	윈도우의 시스템 메뉴에서 윈도우 닫기를 시도할 때
		void windowIconified(WindowEvent)	윈도우가 아이콘화 될 때
		void windowDeiconfied(WindowEvent)	아이콘 상태에서 원래 상태로 복귀할 때
		void windowClosed(WindowEvent)	윈도우가 닫혔을 때
		void windowActivated(WindowEvent)	윈도우가 활성화될 때
		void windowDeactivated(WindowEvent)	윈도우가 비활성화될 때
Adjustment	AdjustmentListener	void adjustmentValueChanged(AdjustmentEvent)	스크롤바를 움직일 때
Component	ComponentListener	void componentHidden(ComponentEvent)	컴포넌트가 보이지 않는 상태로 될 때
		void componentShown(ComponentEvent)	컴포넌트가 보이는 상태로 될 때
		void componentResized(ComponentEvent)	컴포넌트의 크기가 변경될 때
		void componentMoved(ComponentEvent)	컴포넌트의 위치가 변경될 때
Container	ContainerListener	void componentAdded(ContainerEvent)	컴포넌트가 컨테이너에 추가될 때
		void componentRemoved(ContainerEvent)	컴포넌트가 컨테이너에서 삭제될 때

Review: 이벤트 리스너 작성 방법

- 3 가지 방법

1. 독립 클래스로 작성

- 이벤트 리스너를 완전한 클래스로 작성
- 이벤트 리스너를 여러 곳에서 사용할 때 적합

2. 내부 클래스(inner class)로 작성

- 클래스 안에 멤버처럼 클래스 작성
- 이벤트 리스너를 특정 클래스에서만 사용할 때 적합

3. 익명 클래스(anonymous class)로 작성

- 클래스의 이름 없이 간단히 리스너 작성
- 클래스 조차 만들 필요 없이 리스너 코드가 간단한 경우에 적합

JAVA: MOUSE EVENT HANDLING

마우스 이벤트와 마우스 관련 리스너

• 마우스 이벤트

- 사용자의 마우스 조작에 따라 발생하는 이벤트, 8가지 경우

Mouse 이벤트가 발생하는 경우	리스너의 메소드	리스너
마우스가 컴포넌트 위에 올라갈 때	<code>void mouseEntered(MouseEvent e)</code>	<code>MouseListener</code>
마우스가 컴포넌트에서 내려올 때	<code>void mouseExited(MouseEvent e)</code>	<code>MouseListener</code>
마우스 버튼이 눌러졌을 때	<code>void mousePressed(MouseEvent e)</code>	<code>MouseListener</code>
눌러진 버튼이 떼어질 때	<code>void mouseReleased(MouseEvent e)</code>	<code>MouseListener</code>
마우스로 컴포넌트를 클릭하였을 때	<code>void mouseClicked(MouseEvent e)</code>	<code>MouseListener</code>
마우스가 드래그되는 동안	<code>void mouseDragged(MouseEvent e)</code>	<code>MouseMotionListener</code>
마우스가 움직이는 동안	<code>void mouseMoved(MouseEvent e)</code>	<code>MouseMotionListener</code>
마우스 휠이 구르는 동안	<code>void mouseWheelMoved (MouseWheelEvent e)</code>	<code>MouseWheelListener</code>

- 마우스가 눌러진 위치에서 떼어지는 경우 메소드 호출 순서

```
mousePressed(), mouseReleased(), mouseClicked()
```

- 마우스가 드래그될 때 호출되는 메소드 호출 순서

```
mousePressed(), mouseDragged(), mouseDragged(),..., mouseDragged(), mouseReleased()
```

마우스 리너스 달기

- `MouseListener`의 5 개의 이벤트를 처리하는 경우
 - `mouseEntered()`, `mouseExited()`, `mousePressed()`, `mouseReleased()`, `mouseClicked()`
 - 마우스 리스너 등록
 - `component.addMouseListener(myMouseListener);`
- `MouseMotionListener`의 이벤트도 함께 처리하고자 하는 경우
 - `mouseDragged()`, `mouseMoved()`
 - 마우스 모션 리스너 등록 필요
 - `component.addMouseMotionListener(myMouseMotionListener);`

MouseEvent 객체로부터 얻을 수 있는 정보

- 마우스 포인터의 위치
 - int getX(), int getY(),
 - Point getPoint()
- 입력된 마우스 버튼
 - int getButton()
- 마우스 클릭 횟수
 - int getClickCount()

```
public void mousePressed(MouseEvent e) {  
    int x = e.getX();  
    int y = e.getY();  
}
```

```
public void mousePressed(MouseEvent e) {  
    if(e.getButton() == MouseEvent.BUTTON1)  
        System.out.println("Left Button Pressed");  
}
```

```
public void mouseClicked(MouseEvent e) {  
    if(e.getClickCount() == 2) {  
        // 더블클릭을 처리하는 루틴  
    }  
}
```

예제 10-9 : 마우스와 마우스 모션 이벤트 활용

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

public class MouseListenerAllEx extends JFrame {
    private JLabel la = new JLabel("No Mouse Event");

    public MouseListenerAllEx() {
        setTitle("MouseListener와 MouseMotionListener 예제");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container c = getContentPane();
        c.setLayout(new FlowLayout());

        MyMouseListener listener =
            new MyMouseListener();
        c.addMouseListener(listener);
        c.addMouseMotionListener(listener);

        c.add(la);

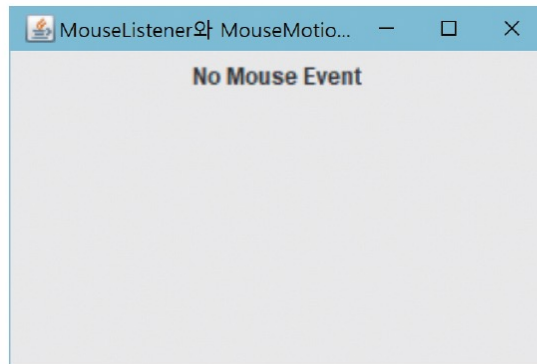
        setSize(300,200);
        setVisible(true);
    }
}
```

```
class MyMouseListener implements MouseListener,
    MouseMotionListener {
    public void mousePressed(MouseEvent e) {
        la.setText("mousePressed (" + e.getX() + ", " + e.getY() + ")");
    }
    public void mouseReleased(MouseEvent e) {
        la.setText("MouseReleased (" + e.getX() + ", " + e.getY() + ")");
    }
    public void mouseClicked(MouseEvent e) {
    public void mouseEntered(MouseEvent e) {
        Component c = (Component)e.getSource();
        c.setBackground(Color.CYAN);
    }
    public void mouseExited(MouseEvent e) {
        Component c = (Component)e.getSource();
        c.setBackground(Color.YELLOW);
    }

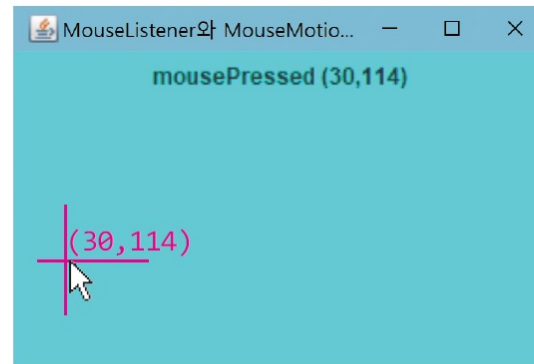
    public void mouseDragged(MouseEvent e) {
        la.setText("MouseDragged (" + e.getX() + ", " + e.getY() + ")");
    }
    public void mouseMoved(MouseEvent e) {
        la.setText("MouseMoved (" + e.getX() + ", " + e.getY() + ")");
    }
}

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

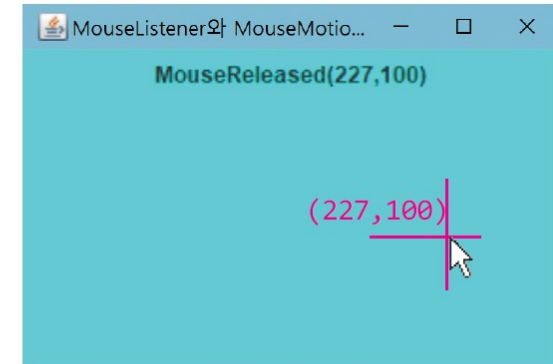
예제 10-9 실행



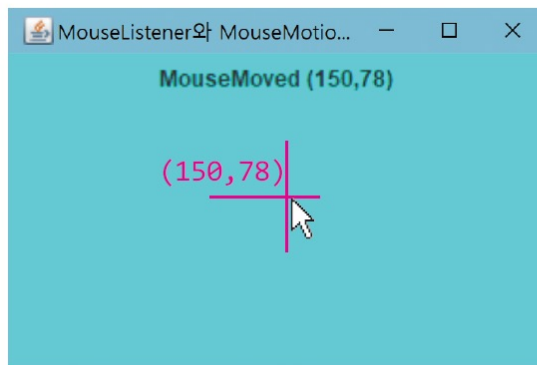
초기 화면



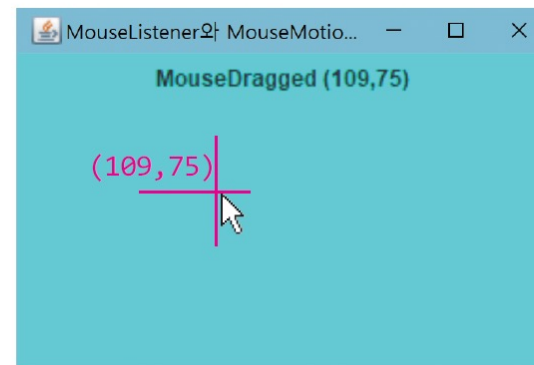
mouseEntered()에 의해 배경색 변경.
마우스 버튼이 눌러진 순간



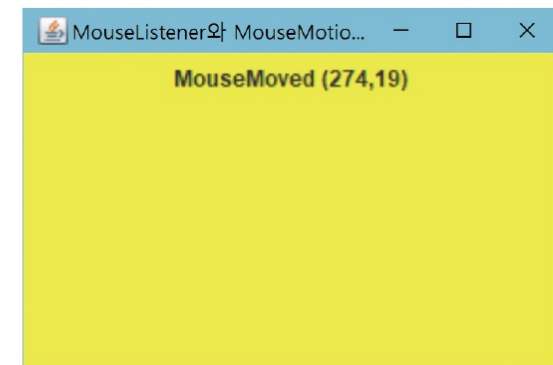
눌러진 마우스 버튼이 떼어진 순간



마우스가 패널 위에 이동하는 동안



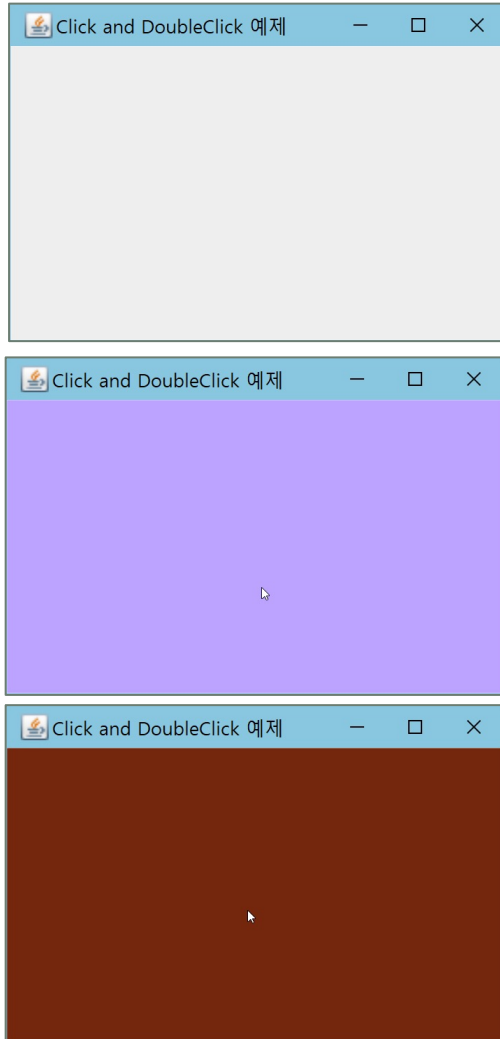
마우스가 패널 위에 드래그하는 동안



마우스가 패널 바깥으로 나가면
mouseExited()에 의해 배경색 변경

예제 10-10 : 더블클릭 시 콘텐츠팬의 배경색 변경 ¹⁵

더블클릭할 때마다 콘텐츠팬의 배경색을 랜덤하게 변경한다.



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

public class ClickAndDoubleClickEx extends JFrame {
    public ClickAndDoubleClickEx() {
        setTitle("Click and DoubleClick 예제");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        Container c = getContentPane();
        c.addMouseListener(new MyMouseListener());
        setSize(300,200);
        setVisible(true);
    }

    class MyMouseListener extends MouseAdapter {
        public void mouseClicked(MouseEvent e) {
            if(e.getClickCount() == 2) {
                int r = (int)(Math.random()*256);
                int g = (int)(Math.random()*256);
                int b = (int)(Math.random()*256);
                Component c = (Component)e.getSource();
                c.setBackground(new Color(r,b,g));
            }
        }
    }

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

MouseEvent와 MouseListener

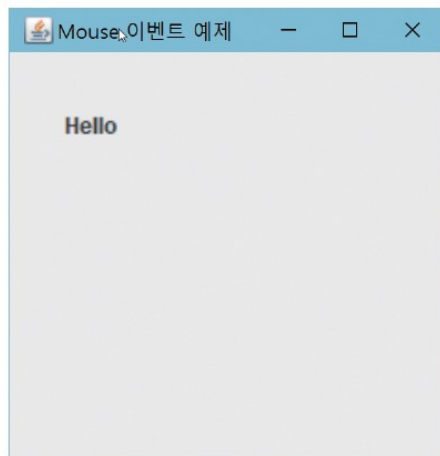
- 마우스 휠 이벤트
 - 마우스 휠이 구를 때마다 발생하는 이벤트
 - 이벤트 발생시마다 MouseEvent 객체 생성
- MouseListener
 - 마우스 휠 이벤트 리스너
 - 마우스 휠 이벤트가 발생할 때마다 호출되는 메소드
 - `Public void mouseWheelMoved(MouseEvent e)`
- 마우스 휠 리스너 구현

```
component.addMouseListener(new MouseListener() {  
    public void mouseWheelMoved(MouseEvent e) {  
        // 마우스 휠의 구르는 방향에 따라 이벤트를 처리한다.  
    }  
});
```

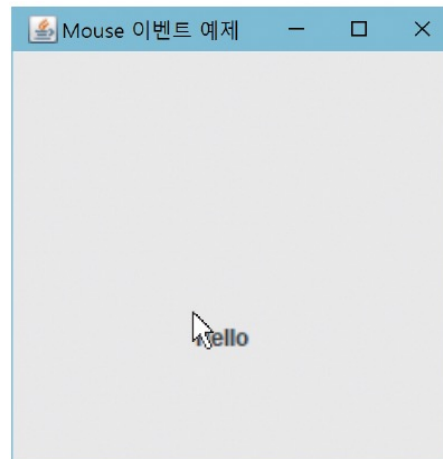
- 실습문제 10-7 참고

예제 10-4 : 마우스로 문자열 이동시키기 – 마우스 이벤트 연습

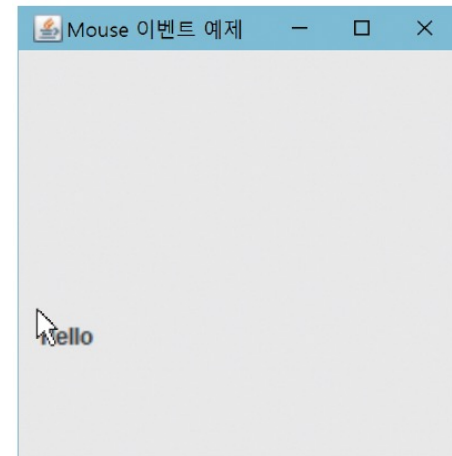
컨텐츠팬의 아무 위치에 마우스 버튼을 누르면 마우스 포인트가 있는 위치로 "hello" 문자열을 옮기는 스윙 응용프로그램을 작성하라.



초기화면



마우스 다른 곳에 클릭한 경우



마우스 다른 곳에 클릭한 경우

예제 10-4의 정답

마우스 버튼이 눌러진
위치를 알아내어
la("hello" 문자열)를 그
위치로 옮긴다.

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

```
public class MouseListenerEx extends JFrame {
    private JLabel la = new JLabel("Hello");

    public MouseListenerEx() {
        setTitle("Mouse 이벤트 예제");

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container c = getContentPane();
        c.addMouseListener(new MyMouseListener());

        c.setLayout(null);
        la.setSize(50, 20);
        la.setLocation(30, 30);
        c.add(la);

        setSize(250, 250);
        setVisible(true);
    }
}
```

```
class MyMouseListener implements MouseListener {
    public void mousePressed(MouseEvent e) {
        int x = e.getX();
        int y = e.getY();
        la.setLocation(x, y);
    }

    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();
}
```

어댑터(Adapter) 클래스

- 이벤트 리스너 구현에 따른 부담 해소를 위해
 - 리스너 작성시 추상 메소드들을 모두 구현해야 하는 부담
 - 마우스 리스너에서 마우스가 눌려지는 경우(mousePressed())만 처리하고자 하는 경우에도 나머지 4 개의 메소드를 모두 구현해야 하는 부담
 - 어댑터 클래스
 - 리스너의 모든 메소드가 단순 리턴하도록 구현해 놓은 클래스
 - `MouseAdapter` 예
- ```
class MouseAdapter implements MouseListener, MouseMotionListener,
 MouseWheelListener {
 public void mousePressed(MouseEvent e) {}
 public void mouseReleased(MouseEvent e) {}
 public void mouseClicked(MouseEvent e) {}
 public void mouseEntered(MouseEvent e) {}
 public void mouseExited(MouseEvent e) {}
 public void mouseDragged(MouseEvent e) {}
 public void mouseMoved(MouseEvent e) {}
 public void mouseWheelMoved(MouseWheelEvent e) {}
}
```
- 추상 메소드가 하나뿐인 리스너는 어댑터 클래스 없음
    - `ActionAdapter`, `ItemAdapter` 클래스는 존재하지 않음

# JDK에서 제공하는 어댑터 클래스

| 리스너 인터페이스           | 대응하는 어댑터 클래스                       |
|---------------------|------------------------------------|
| ActionListener      | 없음                                 |
| ItemListener        | 없음                                 |
| KeyListener         | KeyAdapter                         |
| MouseListener       | MouseAdapter                       |
| MouseMotionListener | MouseMotionAdapter 혹은 MouseAdapter |
| FocusListener       | FocusAdapter                       |
| WindowListener      | WindowAdapter                      |
| AdjustmentListener  | 없음                                 |
| ComponentListener   | ComponentAdapter                   |
| ContainerListener   | ContainerAdapter                   |

# 어댑터 사용 예(MouseAdapter)

```
JLabel la;
contentPane.addMouseListener(new MyMouseListener());
.....
```

```
class MyMouseListener implements MouseListener {
 public void mousePressed(MouseEvent e) {
 int x = e.getX();
 int y = e.getY();
 la.setLocation(x, y);
 }
 public void mouseReleased(MouseEvent e) {}
 public void mouseClicked(MouseEvent e) {}
 public void mouseEntered(MouseEvent e) {}
 public void mouseExited(MouseEvent e) {}
}
```

MouseListener를 이용한 경우

```
JLabel la;
contentPane.addMouseListener(new MyMouseAdapter());
.....
```

```
class MyMouseAdapter extends MouseAdapter {
 public void mousePressed(MouseEvent e) {
 int x = e.getX();
 int y = e.getY();
 la.setLocation(x, y);
 }
}
```

MouseAdapter를 이용한 경우



# 예제 10-5 : MouseAdapter 사용하기

MouseAdapter를 이용하여 예제 10-4를 수정하라.

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

public class MouseAdapterEx extends JFrame {
 private JLabel la = new JLabel("Hello");

 public MouseAdapterEx() {
 setTitle("Mouse 이벤트 예제");
 setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
 Container c = getContentPane();
 c.addMouseListener(new MyMouseAdapter());

 c.setLayout(null);
 la.setSize(50, 20);
 la.setLocation(30, 30);
 c.add(la);

 setSize(250, 250);
 setVisible(true);
 }
}
```

```
class MyMouseAdapter extends MouseAdapter {
 public void mousePressed(MouseEvent e) {
 int x = e.getX();
 int y = e.getY();
 la.setLocation(x, y);
 }
}

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