

예제 10-6 : 다양한 KeyEvent와 KeyListener 활용

1

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

public class KeyListenerEx extends JFrame {
    private JLabel [] keyMessage;

    public KeyListenerEx() {
        setTitle("keyListener 예제");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container c = getContentPane();
        c.setLayout(new FlowLayout());

        c.addKeyListener(new MyKeyListener());

        keyMessage = new JLabel [3];
        keyMessage[0] = new JLabel(" getKeyCode() ");
        keyMessage[1] = new JLabel(" getKeyChar() ");
        keyMessage[2] = new JLabel(" getKeyText() ");

        for(int i=0; i<keyMessage.length; i++) {
            c.add(keyMessage[i]);
            keyMessage[i].setOpaque(true);
            keyMessage[i].setBackground(Color.YELLOW);
        }
    }
}
```

```
setSize(300,150);
setVisible(true);

c.setFocusable(true);
c.requestFocus();
}

class MyKeyListener extends KeyAdapter {
    public void keyPressed(KeyEvent e) {
        int keyCode = e.getKeyCode();
        char keyChar = e.getKeyChar();

        keyMessage[0].setText(Integer.toString(keyCode));
        keyMessage[1].setText(Character.toString(keyChar));
        keyMessage[2].setText(e.getKeyText(keyCode));
    }
}

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

➔ 바탕색을 본인이 원하는 색으로 변경하시오

예제 10-7 : <F1> 키를 입력 받으면 콘텐츠팬의 배경을 초록색으로, % 키를 입력 받으면 노란색으로 변경

2

- ➔ r을 입력하면 빨간색, g는 초록색, y는 노란색, b는 파란색, 그 외의 입력에 대해서는 배경을 흰색으로 변경하고 이에 맞게 타이틀 내용도 수정하시오.
- ➔ 또, 입력 된 키 정보를 "XXX(본인이름)가(이) XX(입력 키)를 입력했습니다" 형태로 출력하도록 하시오.

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

public class KeyCodeEx extends JFrame {
    private JLabel la = new JLabel();

    public KeyCodeEx() {
        setTitle("Key Code 예제 : F1키:초록색, % 키 노란색");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container c = getContentPane();

        c.addKeyListener(new MyKeyListener());
        c.add(la);

        setSize(300,150);
        setVisible(true);

        c.setFocusable(true);
        c.requestFocus();
    }
}
```

```
class MyKeyListener extends KeyAdapter {
    public void keyPressed(KeyEvent e) {
        Container contentPane = (Container)e.getSource();
        la.setText(e.getKeyText(e.getKeyCode())
                  + "키가 입력되었음");

        if(e.getKeyChar() == '%')
            contentPane.setBackground(Color.YELLOW);
        else if(e.getKeyCode() == KeyEvent.VK_F1)
            contentPane.setBackground(Color.GREEN);
    }
}

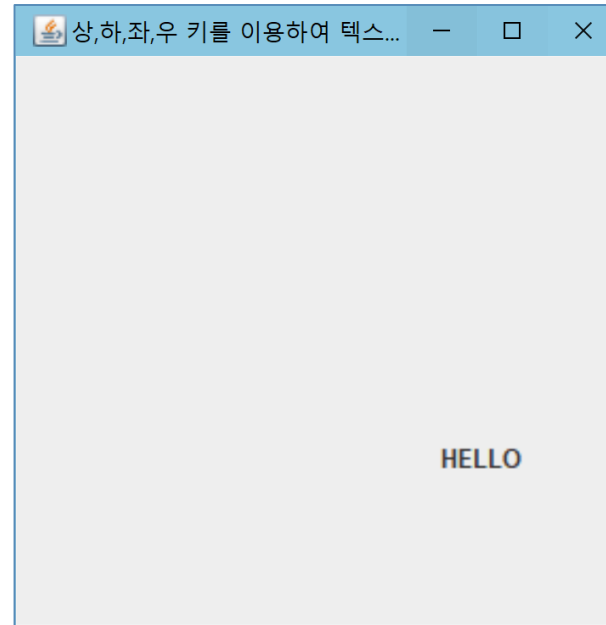
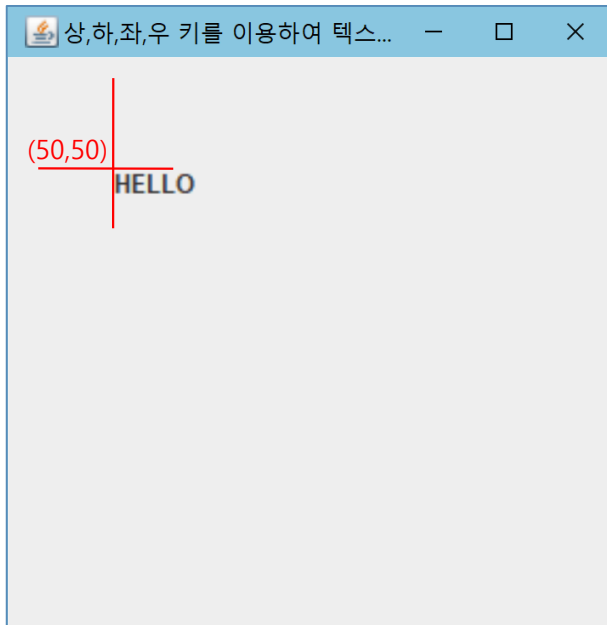
public static void main(String [] args) {
    new KeyCodeEx();
}
```

예제 10-8 : 상(UP), 하(DOWN), 좌(LEFT), 우(RIGHT) 키로 "HELLO" 문자열을 마음대로 움직이기

3

상, 하, 좌, 우 키를 이용하여 "HELLO" 문자열을 움직이는 응용프로그램을 작성하라.

➔ "HELLO" 문자열 대신 본인 이름을 JLabel 컴포넌트로 만들어 컨텐트팬에 부착하고 상, 하, 좌, 우 키를 움직이면 키 방향으로 한 번에 20픽셀씩 움직인다. 이를 위해 컨텐트팬의 배치관리자를 삭제하여야 한다. 문자열을 초기에 (50, 50) 위치에 출력하라.



상,하,좌,우 키를 움직이면 한 번에 20픽셀씩 텍스트는 상,하,좌,우로 이동한다.
이 텍스트는 프레임의 영역을 벗어나서 움직일 수 있다.

예제 소스: 상,하,좌,우 키로 텍스트 움직이기

4

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

public class FlyingTextEx extends JFrame {
    private final int FLYING_UNIT = 10;
    private JLabel la = new JLabel("HELLO");
    public FlyingTextEx() {
        setTitle("상,하,좌,우 키를 이용하여 텍스트 움직이기");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container c = getContentPane();
        c.setLayout(null);
        c.addKeyListener(new MyKeyListener());

        la.setLocation(50,50);
        la.setSize(100,20);
        c.add(la);
        setSize(300,300);
        setVisible(true);
        c.setFocusable(true);
        c.requestFocus();

        c.addMouseListener(new MouseAdapter() {
            public void mouseClicked(MouseEvent e) {
                Component com = (Component)e.getSource();
                com.setFocusable(true);
                com.requestFocus();
            }
        });
    }
}
```

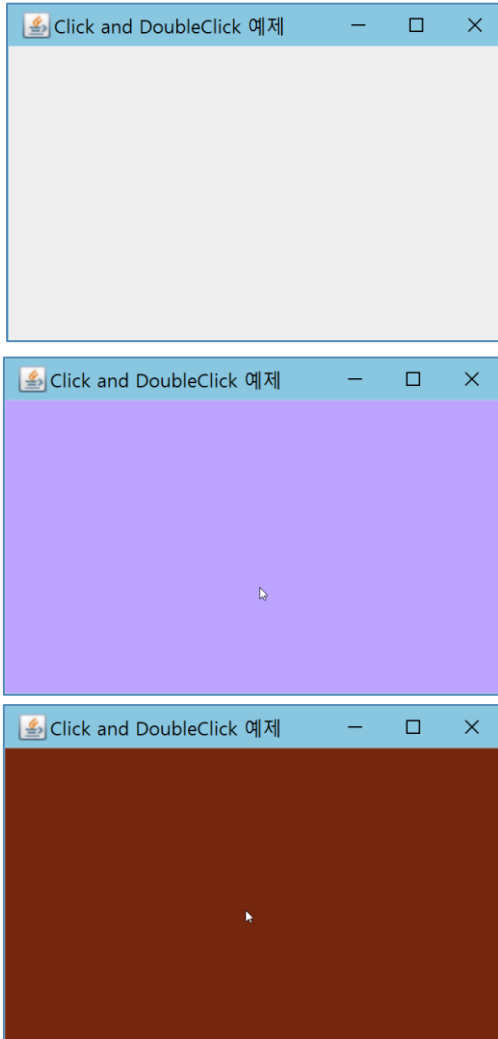
```
class MyKeyListener extends KeyAdapter {
    public void keyPressed(KeyEvent e) {
        int keyCode = e.getKeyCode();

        switch(keyCode) {
            case KeyEvent.VK_UP:
                la.setLocation(la.getX(), la.getY()-FLYING_UNIT);
                break;
            case KeyEvent.VK_DOWN:
                la.setLocation(la.getX(), la.getY()+FLYING_UNIT);
                break;
            case KeyEvent.VK_LEFT:
                la.setLocation(la.getX()-FLYING_UNIT, la.getY());
                break;
            case KeyEvent.VK_RIGHT:
                la.setLocation(la.getX()+FLYING_UNIT, la.getY());
                break;
        }
    }
}

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

예제 10-10 : 더블클릭 시 콘텐츠팬의 배경색 변경

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();
    }
}
```