TextComponent

1. 請參考投影片內容，建立以下視窗應用程式

請將1.程式**執行結果**截圖置入作業中、2.程式**原始檔**置入作業中

1. 建立數個TextField,相關的屬性如下
   1. TextField1, 背景顏色為 BLUE , 前景顏色為 MAGENTA, 選取所有文字內容
   2. TextField2, 回應字元為 #
   3. TextField3, 設定文字欄位的顯示文字行數為5, 不允許編輯
   4. TextField4, 設定文字欄位的顯示文字行數為10, 選取位置6~8的文字



import java.awt.\*;

import java.awt.event.\*;

public class TextFieldDemo extends java.awt.Frame {

public static void main(String args[]){

new TextFieldDemo();

}

// 建構函式

public TextFieldDemo() {

super("TextField Demo");

// 定義 Layout Manager 為 FlowLayout

setLayout(new FlowLayout());

java.awt.TextField textfield;

// 建構函式 1

textfield = new TextField();

// 設定文字欄位的顯示文字

textfield.setText("TextField 1");

// 設定文字欄位的顯示文字字數

textfield.setColumns(3);

// 設定背景顏色

textfield.setBackground(Color.BLUE);

// 設定前景顏色

textfield.setForeground(Color.MAGENTA);

// 選取所有的文字內容

textfield.selectAll();

add(textfield);

// 建構函式 2

textfield = new TextField("TextField 2");

// 設定回應字元

textfield.setEchoChar('#');

add(textfield);

// 建構函式 3

textfield = new TextField(5);

// 設定文字欄位的顯示文字

textfield.setText("TextField 3");

// 設定是否允許編輯

textfield.setEditable(false);

add(textfield);

// 建構函式 4

textfield = new TextField("TextField 4", 10);

// 選取由selectionStart啟始位置

// 至selectionEnd結束位置的文字內容

textfield.select(6, 8);

add(textfield);

// 設定視窗的大小

this.setSize(200, 100);

// Center the frame

Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();

Dimension frameSize = this.getSize();

if (frameSize.height > screenSize.height)

frameSize.height = screenSize.height;

if (frameSize.width > screenSize.width)

frameSize.width = screenSize.width;

this.setLocation((screenSize.width - frameSize.width) / 2, (screenSize.height - frameSize.height) / 2);

// 顯示視窗

this.setVisible(true);

this.addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent e) {

System.exit(0);

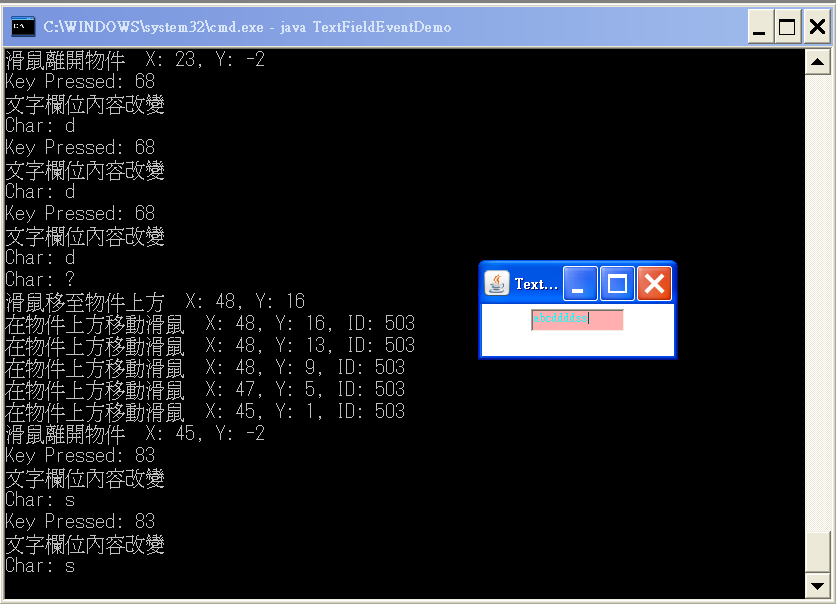
}

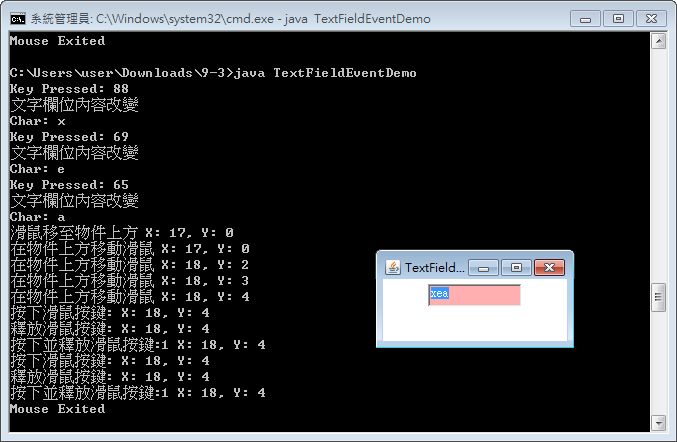
});

}

}

1. 建立具有TextListener的視窗應用程式,
   1. 建立1個TextField, 相關屬性設定請自行定義
   2. 將ActionListener, KeyListener, MouseListener, MouseMotionListener,TextListener實做與註冊
   3. 執行結果請參考moodle上所放的程式





import java.awt.\*;

import java.awt.event.\*;

public class TextFieldEventDemo extends java.awt.Frame {

java.awt.TextField textfield;

public static void main(String args[]){

new TextFieldEventDemo();

}

// 建構函式

public TextFieldEventDemo() {

super("TextField Event Demo");

// 定義 Layout Manager 為 FlowLayout

setLayout(new FlowLayout());

// 建構函式 1

textfield = new TextField();

// 設定文字欄位的顯示文字

textfield.setText("TextField 1");

// 設定文字欄位的顯示文字字數

textfield.setColumns(10);

// 設定背景顏色

textfield.setBackground(Color.PINK);

// 設定前景顏色

textfield.setForeground(Color.CYAN);

// 選取所有的文字內容

textfield.selectAll();

// 以Inner Class的方式使用TextListener介面

textfield.addTextListener(new TextListener() {

public void textValueChanged(TextEvent e) {

if (e.getID() == TextEvent.TEXT\_VALUE\_CHANGED)

System.out.println("文字欄位內容改變");

}

});

textfield.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if (e.getSource().equals(textfield))

System.out.println("於文字欄位上按下Enter鍵");

}});

textfield.addKeyListener(new KeyListener() {

public void keyPressed(KeyEvent e) {

System.out.println("Key Pressed: " + e.getKeyCode());

}

public void keyReleased(KeyEvent e) {

System.out.println("Char: " + e.getKeyChar());

}

public void keyTyped(KeyEvent e) {}

});

textfield.addMouseListener(new MouseListener(){

public void mouseClicked(MouseEvent e) {

System.out.println("按下並釋放滑鼠按鍵:" + e.getButton() + " X: " + e.getX() + ", Y: " + e.getY());

}

public void mouseEntered(MouseEvent e) {

System.out.println("滑鼠移至物件上方" + " X: " + e.getX() + ", Y: " + e.getY());

}

public void mouseExited(MouseEvent e) {

System.out.println("Mouse Exited");

}

public void mousePressed(MouseEvent e) {

System.out.println("按下滑鼠按鍵:" + " X: " + e.getX() + ", Y: " + e.getY());

}

public void mouseReleased(MouseEvent e) {

System.out.println("釋放滑鼠按鍵:" + " X: " + e.getX() + ", Y: " + e.getY());

}});

textfield.addMouseMotionListener(new MouseMotionListener(){

public void mouseDragged(MouseEvent e) {

System.out.println("按下滑鼠按鍵並拖曳滑鼠" + " X: " + e.getX() + ", Y: " + e.getY());

}

public void mouseMoved(MouseEvent e) {

System.out.println("在物件上方移動滑鼠" + " X: " + e.getX() + ", Y: " + e.getY());

}});

add(textfield);

// 設定視窗的大小

this.setSize(200, 100);

// Center the frame

Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();

Dimension frameSize = this.getSize();

if (frameSize.height > screenSize.height)

frameSize.height = screenSize.height;

if (frameSize.width > screenSize.width)

frameSize.width = screenSize.width;

this.setLocation((screenSize.width - frameSize.width) / 2, (screenSize.height - frameSize.height) / 2);

// 顯示視窗

this.setVisible(true);

this.addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent e) {

System.exit(0);

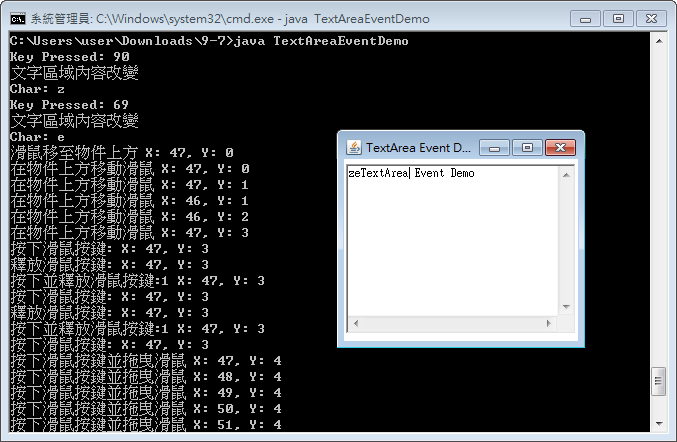
}

});

}

}

1. 建立有TextArea且具有事件監控的視窗應用程式,
   1. 建立TextArea, 相關屬性設定請自行定義
   2. 將KeyListener, MouseListener, MouseMotionListener, TextListener實做與註冊
   3. 執行結果請參考moodle上所放的程式



import java.awt.\*;

import java.awt.event.\*;

public class TextAreaEventDemo extends java.awt.Frame {

java.awt.TextArea textarea;

public static void main(String args[]){

new TextAreaEventDemo();

}

// 建構函式

public TextAreaEventDemo() {

super("TextArea Event Demo");

textarea = new TextArea();

// 設定顯示文字

textarea.setText("TextArea Event Demo");

// 設定TextArea的顯示列數

textarea.setRows(10);

// 設定TextArea的顯示行數

textarea.setColumns(30);

// 以Inner Class的方式使用TextListener介面

textarea.addTextListener(new TextListener() {

public void textValueChanged(TextEvent e) {

if (e.getID() == TextEvent.TEXT\_VALUE\_CHANGED)

System.out.println("文字區域內容改變");

}

});

textarea.addKeyListener(new KeyListener() {

public void keyPressed(KeyEvent e) {

System.out.println("Key Pressed: " + e.getKeyCode());

}

public void keyReleased(KeyEvent e) {

System.out.println("Char: " + e.getKeyChar());

}

public void keyTyped(KeyEvent e) {}

});

textarea.addMouseListener(new MouseListener(){

public void mouseClicked(MouseEvent e) {

System.out.println("按下並釋放滑鼠按鍵:" + e.getButton() + " X: " + e.getX() + ", Y: " + e.getY());

}

public void mouseEntered(MouseEvent e) {

System.out.println("滑鼠移至物件上方" + " X: " + e.getX() + ", Y: " + e.getY());

}

public void mouseExited(MouseEvent e) {

System.out.println("Mouse Exited");

}

public void mousePressed(MouseEvent e) {

System.out.println("按下滑鼠按鍵:" + " X: " + e.getX() + ", Y: " + e.getY());

}

public void mouseReleased(MouseEvent e) {

System.out.println("釋放滑鼠按鍵:" + " X: " + e.getX() + ", Y: " + e.getY());

}});

textarea.addMouseMotionListener(new MouseMotionListener(){

public void mouseDragged(MouseEvent e) {

System.out.println("按下滑鼠按鍵並拖曳滑鼠" + " X: " + e.getX() + ", Y: " + e.getY());

}

public void mouseMoved(MouseEvent e) {

System.out.println("在物件上方移動滑鼠" + " X: " + e.getX() + ", Y: " + e.getY());

}});

java.awt.Panel panel1 = new Panel();

panel1.add(textarea);

add(panel1);

// 設定視窗的大小

this.setSize(250, 220);

// Center the frame

Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();

Dimension frameSize = this.getSize();

if (frameSize.height > screenSize.height)

frameSize.height = screenSize.height;

if (frameSize.width > screenSize.width)

frameSize.width = screenSize.width;

this.setLocation((screenSize.width - frameSize.width) / 2, (screenSize.height - frameSize.height) / 2);

// 顯示視窗

this.setVisible(true);

this.addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent e) {

System.exit(0);

}

});

}

}