**小馬**

import java.awt.BorderLayout;

import java.awt.Color;

import java.awt.Container;

import java.awt.Graphics;

import java.awt.Insets;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.BorderFactory;

import javax.swing.ImageIcon;

import javax.swing.\*;

import javax.swing.Timer;

import javax.swing.border.Border;

public class gg extends JFrame implements ActionListener {

ImageIcon p1 = new ImageIcon("hours1.png");//設定 圖檔來源

int o1; //Button 移動位置

//宣告 Timer

private Timer timer;

//設定 終點

int width = 735;

Container c = getContentPane();

Border bLine;

JButton bt1;

JPanel jp = new JPanel();

public gg() {

//設定 Button 物件

bt1 = new JButton("粉紅小馬");

//設定 Timer

timer = new Timer(100,this);

timer.setInitialDelay(0); //setInitialDelay = 0

c.setLayout(null);

c.setBackground(Color.white);

bLine = BorderFactory.createLineBorder(Color.blue);

JPanel jpane1 = new JPanel();

jpane1.setBounds(800, 1, 1, 500);

jpane1.setBorder(bLine);

c.add(jpane1, BorderLayout.EAST);

timer.start();

}

public void actionPerformed(ActionEvent arg0) {

//如果某匹馬 >= 終點 (1.結束timer 2.顯示 比賽結果)

if ((o1) >= 735) {

timer.stop(); // 結束timer

JOptionPane.showMessageDialog(jp, "粉紅小馬贏了","比賽結果",

JOptionPane.WARNING\_MESSAGE);

repaint();//重繪

}

public void paint(Graphics g) { // 執行賽馬

Insets ins = getInsets(); // 取得邊線尺寸

int width = getWidth() - (ins.left + ins.right);

int height = getHeight() - (ins.top + ins.bottom);

g.setColor(Color.WHITE);

g.fillRect(ins.left, ins.top, width, height); //畫出正方形

int rnd1 = (int) (Math.random() \* 30 + 1); //隨機 1 ~ n , ex. 1-100:Math.random() \* 100 +1

o1 = o1 + rnd1; //累加上次移動距離

bt1.setBounds(o1, 30, 65, 70); // 設定 X: 水平距離, Y:垂直距離

bt1.setIcon(p1); //設定 button 圖片

c.add(bt1);

g.setColor(Color.blue);

g.drawLine(810, 0, 810, 700);//畫終點線

int number = (int) (Math.random() \* 4) + 1; // 觀眾席

if (number == 1) {

g.setColor(Color.green);

g.drawString("加油!! 粉紅小馬!!", o1, getHeight() - 20);

}

}

public static void main(String[] args) {

gg app = new gg();

app.setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);

app.setSize(1000, 500);

app.setLocation(100, 100);

app.setVisible(true);

app.setTitle("賽彩虹小馬");

}

}

**計時器**

package g;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.\*;

public class gg extends JFrame implements ActionListener {

private Timer timer; //宣告 Timer

public int s=0;

public int ms=0;

public int m=0;

public String s\_s;

public String s\_ms;

public String s\_m;

JPanel jp = new JPanel();

JLabel jl = new JLabel("Timer: ");

JButton jb =new JButton("開始");

JButton jb1 =new JButton("停止");

JButton jb2 =new JButton("重製");

public gg() {

Container c = getContentPane(); //顯示視窗

c.setBackground(Color.WHITE); //設定背景色

jl.setFont(new Font("標楷體", Font.BOLD, 15));

//設定 Timer

timer = new Timer(10,this);

timer.setInitialDelay(0); //setInitialDelay = 0

jb.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

timer.start();

}

});

jb1.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

timer.stop();

}

});

jb2.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

timer.stop();

s=0;

ms=0;

m=0;

s\_s="00";

s\_ms="00";

s\_m="00";

jl.setText("Timer:"+s\_m+":"+s\_s+":"+s\_ms);

}

});

jp.add(jl);

jp.add(jb);

jp.add(jb1);

jp.add(jb2);

c.add(jp);

}

public void actionPerformed(ActionEvent arg0) {

ms++;

if(ms<10){s\_ms="0"+Integer.toString(ms);}

else if(ms>=10 && ms<=99){s\_ms=Integer.toString(ms);}

else if(ms>99){ms=0;s++;}

if(s<10){s\_s="0"+Integer.toString(s);}

else if(s>=10 && s<=59){s\_s=Integer.toString(s);}

else if(s>59){s=0;m++;}

if(m<10){s\_m="0"+Integer.toString(m);}

else if(m>=10 && m<=59){s\_m=Integer.toString(m);}

else if(m>59){m=0;}

jl.setText("Timer:"+s\_m+":"+s\_s+":"+s\_ms);

}

public static void main(String[] args) {

// TODO Auto-generated method stub

gg app = new gg();

app.setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);

app.setSize(300, 300);

app.setLocation(500, 200);

app.setVisible(true);

app.setTitle("計時器");

}

}

**執行緒**

class UserThread extends Thread { // 執行緒類別, 繼承Thread類別

private int length;

// 建構子

public UserThread(int length, String name) {

super(name); //呼叫Thread裡的function

this.length = length;

}

// 執行執行緒

public void run() {

int temp = 0;

for ( int i = 1; i <= length; i++ ) temp += i;

System.out.println("執行緒:"+Thread.currentThread() + " 總和 = " + temp+" 數量:"+Thread.activeCount() );

//Thread[main,5,main] 第一個main是執行緒名，5是指優先權，第二個main是指來自於哪個父類別，此處是來自main thread

}

}

public class ex1 { // 主類別

public static void main(String[] args) { // 主程式

System.out.println("執行緒: " + Thread.currentThread());

UserThread ut1 = new UserThread(5, "執行緒A"); // 建立執行緒物件

UserThread ut2 = new UserThread(10, "執行緒B");

ut1.start(); ut2.start(); // 啟動執行緒

}

}

**JTextField**

JTextField text = new JTextField(12);

JPasswordField password = new JPasswordField(12);

**JComboBox**

String[] items = { "程式語言", "計算機概論","資料庫系統", "網頁設計"};

JComboBox list = new JComboBox(items);

**showMessageDialog()**方法可顯示【確定】按鈕的訊息視窗

JOptionPane.showMessageDialog(jpane, "測試訊息視窗!");

**showConfirmDialog()** 方法可顯示詢問問題的對話方塊

int n = JOptionPane.showConfirmDialog(jpane," 測試訊息視窗! ","標題“,JOptionPane.YES\_NO\_OPTION);

**showOptionDialog()** 方法可顯示指定標題文字、圖示、訊息

Object[] options = {"showMessageDialog按鈕","showConfirmDialog按鈕"};

int m = JOptionPane.showOptionDialog(jpane, "哪一個按鈕顯示警告訊息?",

"操作問題", JOptionPane.YES\_NO\_OPTION, JOptionPane.QUESTION\_MESSAGE, null,

options, options[0]);