

P.O.O.

Practica 4

Hilos

Instrucciones: Ver al final la estrategia para realizar esta practica

0.-Ticker

Colocar un texto en una etiqueta y mover las letras de dicho texto de izquierda a derecha (o de derecha a izquierda) sin mover el texto de lugar. Se puede quitar la primer letra y pegarla (con el operador +) al final y repetir esto de forma periódica.

1.-Tamagochi (Java3D, carpeta estatiProgBarSinRed)

Crear una interfaz gráfica (usando Swing) con 3 etiquetas (hambre, sueño, aburrimiento), 3 barras de progreso y 3 botones (comer, dormir, jugar).

- Cada cierto tiempo debe ir llenando las 3 barras de progreso (usar un hilo).
- Si cualquiera de las tres barras de progreso se llena entonces cambiar la textura (a la que corresponde a un tamagochi muerto) e imprimir un mensaje con el texto el tamagochi ha muerto.
- Cada que se presiona uno de los botones disminuir la barra de progreso correspondiente.

2.-Pégale al topo (continuación)

Si hizo el tablero de pégale al topo en una practica anterior agregue un hilo para sustituir (después de una cierta cantidad de segundos) el tablero actual por un tablero con una nueva distribución aleatoria de topos.

3.-Marco Digital (carpeta MarcoDigital)

Si hizo el álbum de fotos digital en una practica anterior (no es indispensable que lo haya hecho en una practica previa) use un hilo para que el el álbum de fotos digital tenga un modo de diapositivas en dicho modo la imagen actual se sustituye por la siguiente (de forma circular usando el operador modulo) después de cierto tiempo. Poner una ComboBox que permita elegir el lapso de tiempo entre imágenes.

4.-Animación con Sprites (carpeta flappy bird)

Usar un hilo para mover el sprite actual del ave de izquierda a derecha (o de derecha a izquierda) de la pantalla y que si el ave sale por un extremo entre por el otro. Cambiar el sprite actual para que se vea aletear al ave.

***5.-Descenso del nivel de agua en un tanque (carpeta válvula)**

El programa Simula.java de esta carpeta simula (usando un hilo) un tanque de agua, una válvula y una tubería. Cuando la válvula esta abierta la tubería se llena (el color de la tubería es azul cuando esta llena). Lea el código en los archivos de la carpeta para determinar cual hay que modificar para que el nivel del tanque de agua vaya descendiendo cada cierto tiempo hasta quedar vacío. Además modifique el código para que una vez que se vacía el tanque de agua también quede vacía la tubería (el color de la tubería es blanco cuando esta vacía).

***6.-Programa para presentaciones electrónicas**

Similar a la opción tres pero cuando el programa esta en modo de diseño permita que el usuario escriba varias líneas de texto en cada diapositiva. Use un hilo para que cuando el programa este en modo de presentación de diapositivas la diapositiva actual se sustituye por la siguiente después de cierto tiempo.

***7.-Aplicador de exámenes (Continuacion)**

Similar a las opciones 3 y 6 se trata de aplicar un examen de opción múltiple al usuario dándole cierta cantidad de segundos para responder cada pregunta del examen . Usar una etiqueta para la pregunta y 4 RadioButtons para las opciones A, B, C, y D.

Al final divida el numero de respuestas correctas entre el numero total de preguntas para evaluar al usuario e imprima la calificación.

***8.-Agenda de actividades con alarmas**

Se tiene una lista de actividades donde algunas pueden tener asociadas alarmas. Cada minuto se checa la lista de actividades para que si hay una alarma programada para esa hora y minuto mostrar una notificación (la cual puede mostrarse en un JOptionPane). Permitir que el usuario ponga la hora y el minuto de las alarmas.

Estrategia para esta practica

La práctica 4 es facil de hacer si se entiende el sig. codigo visto en clase

//Printer_.java

```
public class Printer_ implements Runnable {
    Thread PrinterThread;
    String string;
    int count;
    int sleepTime;

    public Printer_(String s, int howMany, int sleep){
        count=howMany;
        string=s;
        sleepTime=sleep;
        PrinterThread=new Thread(this);
        PrinterThread.start();
    }
    public void run(){
        while(count-- >0){
            System.out.println(string);
            try{
                Thread.sleep(sleepTime);//pausa
            } catch (Exception e) {
                return;
            }
        }
    }
    public static void main(String args[]){
        new Printer_("Ping", 5, 300);
        new Printer_("Pong", 5, 500);
    }
}
```

Puede ser muy util el esqueleto de abajo (basado en el codigo de arriba) de una clase que usa hilos (dado que es un esqueleto hay que poner el codigo que falte)

```
//MiClase.java
public class MiClase implements Runnable {
    Thread hilo;
    //poner variables de instancia
    public MiClase(){

        hilo=new Thread(this);
        hilo.start();
    }
    public void run(){
        //poner cuerpo del hilo
    }
    public static void main(String args[]){
        new MiClase();
    }
}
```

Esqueleto si la clase necesita una IGU

```
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
import java.util.*;
public class AppInterGrafı extends JFrame implements Runnable {
    Thread hilo;
    Panel p;

    public AppInterGrafı() {
        super("MiApp");
        p=new Panel();

        p.add(...); //agregar elementos de IGU al panel
        Container content = getContentPane();
        content.add(p, BorderLayout.CENTER);
        setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        setSize(400, 300); setVisible(true);
        hilo=new Thread(this); //debe ser de las ultimas lin. del constructor
        hilo.start(); //debe ser de las ultimas lineas del constructor
    }
    public void run(){
        //poner cuerpo del hilo
    }
    public static void main(String[] args) { new AppInterGrafı(); }
}
```

NOTAS:

- No basta con usar sleep se deben usar objetos de tipo Thread
- Repasar hilos en Java