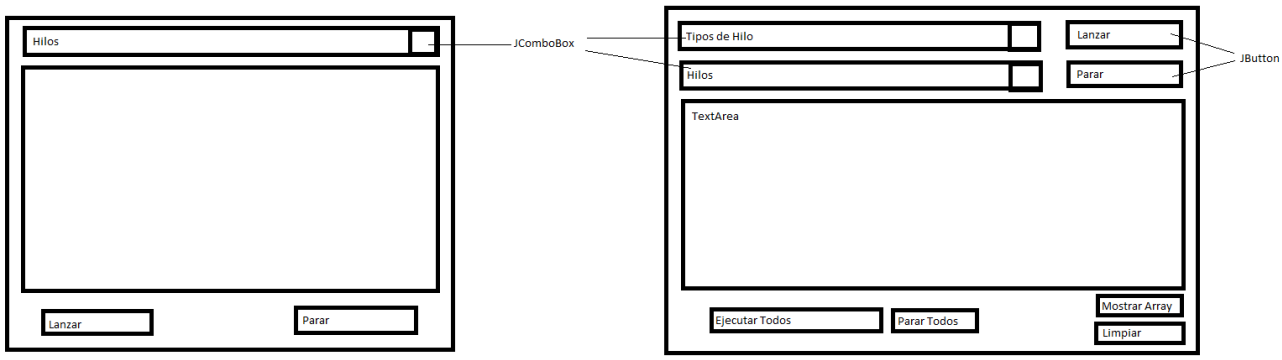
DAD-1: Hilos Parte 2

Cambiamos el primer ejercicio de hilos, a la izquierda para que se vea como el de la derecha



Especificación GUI

El programa que recibimos como base (Similar al de la izquierda) tiene un array de Hilos.

Botones

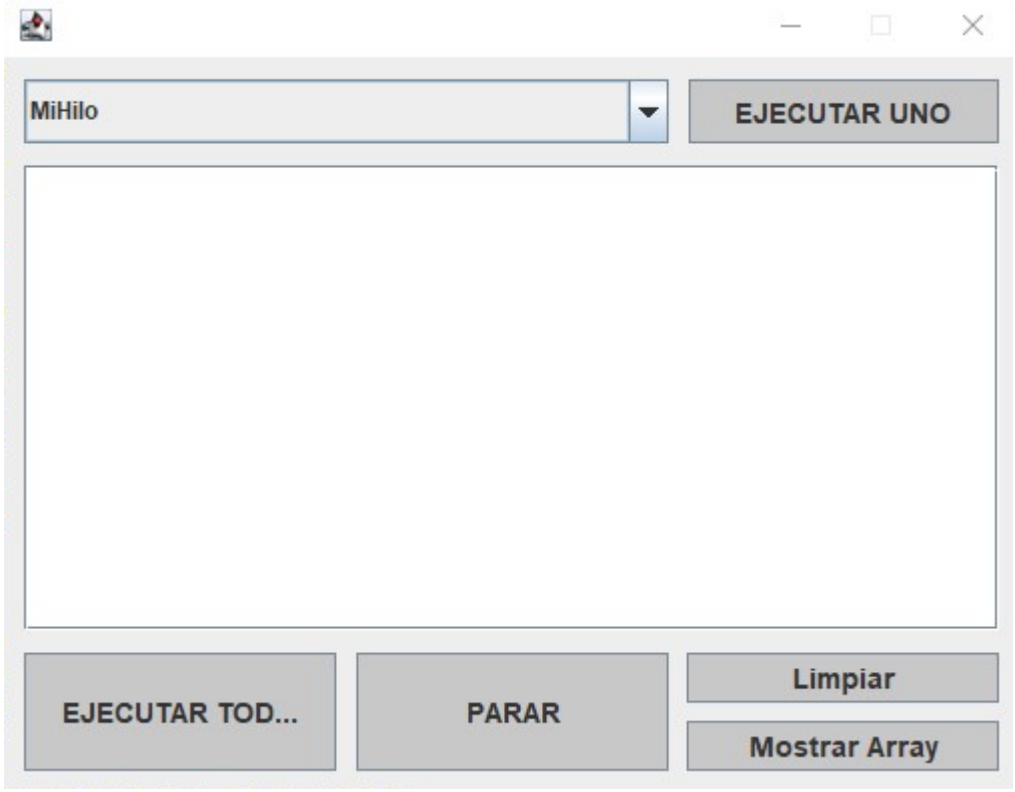
- Lanzar: Lanza 1 Hilo (Seleccionado en Tipos de Hilo)
- Parar: Para el hilo anteriormente lanzado
- Ejecutar Todos: Ejecuta todos los Hilos
- Parar Todos: Para todos los Hilos
- Mostrar Array: Muestra el array de hilos que se han arrancado

JComboBox

- Tipos de Hilo: Para seleccionar entre
- Hilos: Para seleccionar Hilos

Interfaz del Programa Recibido

Botones



- Ejecutar Uno: Ejecuta el hilo seleccionado en el ComboBox
- Ejecutar Todos: Ejecuta todos los hilos del Array de Hilos
- Parar: Para todos los hilos
- Limpiar: Borra el texto mostrado en el log
- Mostrar Array: Muestra todos los Hilos activos (Estan contenidos en un array.)

JComboBox

- MiHilo: Seleccionamos el Hilo que queremos ejecutar

Clases del Código recibido

Clase MiHilo

```java
public class MiHilo extends Thread {

    public MiHilo() {}

    @Override
    public void run() {
        try {
            for (int i = 1; i <= 5; i++) {

                if (Thread.interrupted()) {
                    return;
                }

                System.out.println("Hilo1: " + i);
                Thread.sleep(1000);
            }
        } catch (InterruptedException e) {
            System.out.println("Hilo1 interrumpido.");
        }
    }
}
```

Clase MiHilo2

```java
public class MiHilo2 extends Thread {
    private String threadWord;

    public MiHilo2(String threadWord) {
        this.threadWord = threadWord;
    }

    @Override
    public void run() {
        try {
            for (int i = 1; i <= 5; i++) {

                if (Thread.interrupted()) {
                    return;
                }

                System.out.println("Hilo2: " + threadWord);
                Thread.sleep(1000);
            }
        } catch (InterruptedException e) {
            System.out.println("Hilo2 interrumpido.");
        }
    }
}
```

Clase NombreHilos

```java
public enum NombreHilos {
    HILO_1("MiHilo"), HILO_2("MiHilo2");

    private final String nombreHilo;

    private NombreHilos(String nombreHilo) {
        this.nombreHilo = nombreHilo;
    }

    public String getNombreHilo() {
        return nombreHilo;
    }

    public static String[] getNombreHilosArray() {
        NombreHilos[] values = NombreHilos.values();
        String[] nombresHilosArray = new String[values.length];

        for (int i = 0; i < values.length; i++) {
            nombresHilosArray[i] = values[i].getNombreHilo();
        }

        return nombresHilosArray;
    }
```

Clase ButtonEjecutar

```java
import java.awt.event.ActionEvent;

public class ButtonEjecutar implements ActionListener {
    private MiHilo hilo1;//Clase Mi Hilo en Paquete Back
    private MiHilo2 hilo2;//Clase Mi Hilo2 en Paquete Back
    private List<Thread> hilos;//Declaración de Array de Hilos

    public ButtonEjecutar(List<Thread> hilos) {
        this.hilos = hilos;//Hilos es el Array
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        hilo1 = new MiHilo();//Nuevo Hilo
        hilo2 = new MiHilo2("working...");

        hilos.add(hilo1);//Metemos el nuevo hilo en el array
        hilos.add(hilo2);//Lo mismo pero con Hilo2

        hilo1.start();//Iniciamos el Hilo
        hilo2.start();//Lo mismo pero con Hilo2
    }

    public List<Thread> getHilos() {
        return hilos;
    }

    public MiHilo getHilo1() {
        return hilo1;
    }

    public MiHilo2 getHilo2() {
        return hilo2;
    }
}
```

Clase ButtonEjecutarUno

```java
import java.awt.event.ActionEvent;

public class ButtonEjecutarUno implements ActionListener {
    private MiHilo hilo1;//Declaramos hilo tipo 1
    private MiHilo2 hilo2;//Declaramos hulo tipo 2
    private JComboBox<String> comboBox;//Declaramos el jcombobox
    private List<Thread> hilos;//Declaramos el array de hilos

    public ButtonEjecutarUno(JComboBox<String> comboBox, List<Thread> hilos) {
        this.comboBox = comboBox;
        this.hilos = hilos;
    }

    @Override
    public void actionPerformed(ActionEvent e) {//acción realizada por el botón ejecutar 1
        String selectedThread = (String) comboBox.getSelectedItem();

        if (selectedThread != null) {//Si el combobox no está en NULL
            if (selectedThread.equals(NombreHilos.HILO_1.getNombreHilo())) {//Si el hilo seleccionado es Hilo1
                hilo1 = new MiHilo();//Declaramos Hilo1
                hilo1.start();//Arrancamos Hilo1
                hilos.add(hilo1);//Añadimos el Hilo1 al Array
            } else if (selectedThread.equals(NombreHilos.HILO_2.getNombreHilo())) {//En caso contrario, si es le Hilo2
                hilo2 = new MiHilo2("Studying...");//Declaramos Hilo2 con mensaje de arranque
                hilo2.start();//Iniciamos el Hilo2
                hilos.add(hilo2);//Metemos el Hilo en el array de Hilo
            }
        }

    }

    public List<Thread> getHilos() {
        return hilos;
    }

    public MiHilo getHilo1() {
        return hilo1;
    }

    public MiHilo2 getHilo2() {
        return hilo2;
    }

}
```

From:
http://www.knoppia.net/ - **Knoppia**

Permanent link:
http://www.knoppia.net/doku.php?id=dad1:hilos2

Last update: **2023/09/21 18:40**