Clase de PSP final de tema 1

Contenido

1 Introducción

En este problema trabajaremos sobre lo aprendido en el Tema 1 de programación de servicios y procesos. Se trata de una **tarea de resolución de problemas de alto nivel de pensamiento**. Se complementa con la **tarea guiada** que encontrareis en el fichero word **"Tema1. TareaGuiadaEjemploDeHilosYProcesosFinalizando.docx." y la tarea individual Tarea 1. PSP usando hilos y procesos.docx.** Si los alumnos son incapaces de realizarla por si mismos, **realizaran esta tarea guiada**, para la perfecta compresión de los resultados de aprendizaje, u objetivos relacionados con el tema.

2 Duración

2 horas.

3 Objetivos

a. Se han reconocido las características de la programación concurrente y sus ámbitos de aplicación.
b. Se han identificado las diferencias entre programación paralela y programación distribuida, sus ventajas e inconvenientes.
c. Se han analizado las características de los procesos y de su ejecución por el sistema operativo.
d. Se han caracterizado los hilos de ejecución y descrito su relación con los procesos.
e. Se han utilizado clases para programar aplicaciones que crean subprocesos.
f. Se han utilizado mecanismos para sincronizar y obtener el valor devuelto por los subprocesos iniciados.
g. Se han desarrollado aplicaciones que gestionen y utilicen procesos para la ejecución de varias tareas en paralelo.
h. Se han depurado y documentado las aplicaciones desarrolladas.

4 Clase

4.1 Introducción a la clase

La tarea es demostrar con tres clases java y lo aprendido **en el tema de las clases ProcessBuilder, Process y Thread**, demostrar que cuando un proceso no acaba, hasta que todos los hilos que han lanzado han acabado.

4.2 Materiales de base y conocimientos previos

La clase **UsaHilo3 y HiloSimple3**, que son muy similares a **UsoHilo2** y **HiloSimple2** que han trabajado en clase en el día anterior.

```
/**
 *
 * @author carlo
 */
public class UsaHilo3 {

    public static void main (String[] args) {

        for (int i=0; i<5 ; i++) {

            new HiloSimple3("Hilo"+String.valueOf(i)).start();
        }

        Thread hilo1= new HiloSimple3("Hilo1");
        hilo1.start();
         Thread hilo2= new HiloSimple3("Hilo2");
        hilo2.start();
         Thread hilo3= new HiloSimple3("Hilo3");
        hilo3.start();
         Thread hilo4= new HiloSimple3("Hilo4");
        hilo4.start();
         Thread hilo5= new HiloSimple3("Hilo5");
        hilo5.start();
        System.out.println("Proceso Principal terminado");


    }

}
```

```
/**
 *
 * @author carlo
 */
public class HiloSimple3 extends Thread{

    public HiloSimple3(String name) {
        super(name);
    }

    public void run() {

        String nombreHilo=this.getName();

        try {
            sleep(2000);
        }catch(Exception e) {
        }
```

```
            System.out.println("HILO " + nombreHilo + " TERMINADO");
        }

}
```

4.3 Secuenciación de la lección.

1. El **profesor enseña el programa** con el que **partimos de base**, UsaHilo3 y HiloSimple3.

2. Durante **los diez primeros minutos no se da pista a ningún alumno**. Los alumnos se enfrentarán al problema ellos solos. En este punto **alguno de los alumnos habrá empezado a encontrar la solución**.

3. **Pasados los diez minutos el profesor empezará** a ofrecer **pistas**. Dos opciones:
    a. Que **alumnos que han encontrado la solución** ofrezcan una primera **explicación**.
    b. Que **el profesor les ofrezca una pista**. Podemos empezar en este punto comentándoles que **deberían añadir una clase más** que use ProcessBuilder
4. Pasados otros diez minutos más alumnos habrán dado con la solución. Para los más rezagados como en el caso anterior.
    a. Que **alumnos que han encontrado la solución** ofrezcan una primera **explicación**.

    b. Que **el profesor les ofrezca una pista**. Podemos empezar en este punto comentándoles que **la nueva clase debe esperar a que UsaHilo3 acabe.**

5. A la media hora se le proporcionará la tarea guiada **TareaGuiadaEjemploDeHilosYProcesosFinalizando.docx** a los alumnos que no hayan conseguido completar los objetivos
6. Los **alumnos** que **complementen la tarea anterior** realizaran la Tarea **Tarea 1. PSP usando hilos y procesos.docx.** Los alumnos que no completaron **objetivos realizarán la tarea tras acabar la tarea guiada.**

7. **Al final de la clase** el alumno realizará **los diagramas de clases usando UML, con la herramienta UML Designer de NetBeans** que ya tenéis instalada, y ya hemos realizado anteriormente**.** El alumno explicará en un par de párrafos el diagrama de clases y el funcionamiento de los programas.

Actividad	**Duración**
Actividad inicial. Para enganchar a los alumnos	5 minutos
Problema del día	30 minutos

Explicación del profesor	5 minutos
Actividad Guiada **TareaGuiadaEjemploDeHilosYProcesosFinalizando.docx**	15 minutos
Explicación del profesor	10 minutos
Tarea 1. PSP usando hilos y procesos.docx	40 minutos
Actividad de salida: Diagrama de clases con explicación	10 minutos

4.4 Actividad Inicial

Como podéis comprobar si ejecutáis **la clase principal UsaHilo3,** parece que **los hilos se terminan después del proceso**. Este **resultado no nos está contando la verdad acerca** del funcionamiento de **los hilos y los procesos**.

```
Proceso Principal terminado
HILO Hilo0 TERMINADO
HILO Hilo1 TERMINADO
HILO Hilo5 TERMINADO
HILO Hilo4 TERMINADO
HILO Hilo1 TERMINADO
HILO Hilo2 TERMINADO
HILO Hilo4 TERMINADO
HILO Hilo3 TERMINADO
HILO Hilo2 TERMINADO
HILO Hilo3 TERMINADO
BUILD SUCCESSFUL (total time: 2 seconds)
```

4.5 Desarrollo del problema del día

Vamos a demostrar que este resultado es engañoso. En realidad, **sabemos por la teoría que un proceso no acaba hasta que todos sus hilos han acabado**. La labor del estudiante en este punto es crear un programa que demuestre nuestra afirmación. Un **programa que genere una salida como la siguiente**:

```
run:
HILO Hilo1 TERMINADO
HILO Hilo0 TERMINADO
HILO Hilo2 TERMINADO
HILO Hilo4 TERMINADO
HILO Hilo3 TERMINADO
HILO Hilo6 TERMINADO
HILO Hilo9 TERMINADO
HILO Hilo7 TERMINADO
HILO Hilo5 TERMINADO
HILO Hilo8 TERMINADO
PROCESO USAHILO3 TERMINADO CORRECTAMENTE. UN PROCESO TERMINA CUANDO TERMINAN TODOS SUS HILOS0
BUILD SUCCESSFUL (total time: 2 seconds)
```

Demostramos que un proceso acaba siempre después del hilo que lanza. Para ello:

a. El programa **debe usar UsaHilo3 y HiloSimple3**. Se **permiten sólo modificaciones del código** ofrecido en lo relativo **a la salida estándar.**
b. Es **obligatorio** usar **ProcessBuilder igualmente.**

5 Tarea guiada

La tarea guiada es la solución al problema anterior la encontrareis en **TareaGuiadaEjemploDeHilosYProcesosFinalizando.docx.** La trabajaran los alumnos que no hayan encontrado la solución al problema.

6 Tarea individual

La tarea individual la podéis encontrar en **Tarea 1. PSP usando hilos y procesos.docx.** Todos los alumnos deben realizarla.

7 Instrucción diferenciada

Los alumnos **intentarán buscar una solución mejorada al programa** que ofrece **del profesor**.

Opcionalmente **crearan el Javadoc**.

Opcionalmente, p**roporcionaran soporte a los alumnos** que no han acabado la tarea **antes de los 20 últimos minutos de clase**.

8 Actividad de salida

A elección del profesor los alumnos realizaran una de estas dos actividades:

8.1 Actividad de salida 1

Al final de la clase el alumno realizará **los diagramas de clases usando UML, con la herramienta UML Designer de NetBeans** que ya tienen instalada, y ya hemos realizado anteriormente**.** El **alumno explicará en un par de párrafos el diagrama de clases** y el **funcionamiento de los programas,** como complemento al diagrama de clases.

8.2 Actividad de salida 2

El alumno **comentará sus programas usando Javadoc** y generará la **documentación Javadoc**.