



Centro de Enseñanza Técnica Industrial

Desarrollo de Software

Programación de hilos por interface

Jesús Alberto Aréchiga Carrillo

22310439

5N

Profesor

José Francisco Pérez Reyes

Octubre 2024

Guadalajara, Jalisco

Introducción

En esta práctica se analiza el comportamiento del código de hilos por interface, ejecutando el código múltiples veces para responder las siguientes preguntas: ¿Se pueden controlar los tiempos de ejecución de cada hilo?, ¿Se puede garantizar el inicio de un hilo en específico?, ¿Puedes garantizar que un hilo en específico termine primero? ¿Existe alguna diferencia de hacerlo con herencia o interface?

Desarrollo

Código para implementar:

```
public class ThreadExample implements Runnable{
    public void run(){
        for(int i = 0; i < 10 ; i++){
            System.out.println(i + " " +
                Thread.currentThread().getName());
        }

        System.out.println("Termina thread " +
            Thread.currentThread().getName());
    }

    public static void main(String [] args){
        new Thread(new ThreadExample(), "Pepe").start();
        new Thread(new ThreadExample(), "Juan").start();
        System.out.println("Termina thread main");
    }
}
```

Resultados:

Termina thread main	Termina thread main	Termina thread main	Termina thread main
0 Pepe	0 Juan	0 Pepe	0 Juan
0 Juan	0 Pepe	0 Juan	0 Pepe
1 Pepe	1 Juan	1 Pepe	1 Juan
1 Juan	1 Pepe	1 Juan	1 Pepe
2 Pepe	2 Juan	2 Pepe	2 Juan
2 Juan	2 Pepe	2 Juan	2 Pepe
3 Pepe	3 Juan	3 Pepe	3 Juan
3 Juan	3 Pepe	3 Juan	3 Pepe
4 Pepe	4 Juan	4 Pepe	4 Juan
4 Juan	4 Pepe	4 Juan	4 Pepe
5 Pepe	5 Juan	5 Pepe	5 Juan
5 Juan	5 Pepe	5 Juan	5 Pepe
6 Pepe	6 Juan	6 Pepe	6 Juan
6 Juan	7 Pepe	6 Juan	6 Pepe
7 Pepe	6 Juan	7 Pepe	7 Juan
7 Juan	8 Pepe	8 Pepe	7 Pepe
8 Pepe	9 Pepe	7 Juan	8 Juan
8 Juan	7 Juan	9 Pepe	8 Pepe
9 Pepe	8 Juan	8 Juan	9 Juan
9 Juan	9 Juan	9 Juan	9 Pepe
Termina thread Pepe	Termina thread Juan	Termina thread Pepe	Termina thread Pepe
Termina thread Juan	Termina thread Pepe	Termina thread Juan	Termina thread Juan

Esta vez se utilizó el código para crear hilos mediante la interfaz Runnable, después de ejecutarlo varias veces no se encontraron diferencias notables con el uso de la clase Thread, como se pudo ver en clase el método Thread internamente utiliza un Runnable por lo que no debería existir diferencia en cuanto a procesamiento o el orden en que se ejecutan los hilos. Como se vió con la clase Thread al ejecutar el mismo código se obtienen resultados distintos dado que no se está controlando el tiempo de procesamiento de cada hilo. La única diferencia es que al implementar la interfaz runnable se puede heredar de otra clase mientras que al utilizar la clase Thread no es posible heredar de otra clase ya que java no soporta herencia múltiple.

¿Se pueden controlar los tiempos de ejecución de cada hilo?

No, siempre existen diferencias en los tiempos de ejecución, se podría implementar un mecanismo de planificación que podría controlar cuales son los hilos o instrucciones que se ejecutan primero, pero no es posible controlar cuanto tiempo tardan en ejecutarse.

¿Se puede garantizar el inicio de un hilo en específico?

Sí, el método start() sigue estando disponible en el objeto que implementa la interfaz Runnable y tiene el mismo funcionamiento que cuando hereda de la clase Thread (Nótese cómo al iniciar el hilo, la forma de hacerlo es primero llamando a la clase Thread que está dentro de la interfaz Runnable).

¿Se puede garantizar que un hilo en específico termine primero?

No, ya que no se puede controlar el tiempo de ejecución de un hilo, no es posible saber cuál hilo terminará primero. Sólo se podría si se implementa un mecanismo de planificación que pueda poner en espera todos los hilos hasta que se termine el hilo deseado, pero no es el caso de este código.

¿Existe alguna diferencia de hacerlo con herencia o interface?

En cuanto a funcionalidad, no existen diferencias, ya que se sigue utilizando la clase de Thread en ambos casos. En cuanto a la implementación, si existen diferencias en el código, se implementa la interfaz para poder heredar de otra clase en caso de ser necesario, pero de igual manera se tiene que crear una instancia de Thread.