

050242596probadorsubprocesos.doc abenr Galindo

Iniciando subprocesos

Subprocesos iniciados, termina main

subproceso1 va a estar inactivo durante 3979

subproceso2 va a estar inactivo durante 210

subproceso3 va a estar inactivo durante 3558

subproceso2 termino su inactividad

subproceso3 termino su inactividad

subproceso1 termino su inactividad

GENERACIÓN CORRECTA (tiempo total: 10 segundos)

// Fig. 16.3: ProbadorSubprocesos.java

// Varios subprocesos imprimiendo en distintos intervalos.

```
class ProbadorSubprocesos {
    public static void main( String [] args )
    {
        // crear y nombrar a cada subproceso
        ImprimirSubproceso subproceso1 = new ImprimirSubproceso( "subproceso1" );
        ImprimirSubproceso subproceso2 = new ImprimirSubproceso( "subproceso2" );
        ImprimirSubproceso subproceso3 = new ImprimirSubproceso( "subproceso3" );
        System.err.println( "Iniciando subprocesos" );
        subproceso1.start(); // iniciar subproceso1 y colocarlo en estado listo
        subproceso2.start(); // iniciar subproceso2 y colocarlo en estado listo
        subproceso3.start(); // iniciar subproceso3 y colocarlo en estado listo
        System.err.println( "Subprocesos iniciados, termina main\n" );
    } // fin de main
} // fin de la clase ProbadorSubprocesos
// la clase ImprimirSubproceso controla la ejecución de los subprocesos
class ImprimirSubproceso extends Thread {
    private int tiempoInactividad;
    // asignar nombre a subproceso, llamando al constructor de la superclase
    public ImprimirSubproceso( String nombre )
    {
        super( nombre );
        // elegir tiempo de inactividad aleatorio entre 0 y 5 segundos
        tiempoInactividad = ( int ) ( Math.random() * 5001 );
    }
    // el método run es el código a ejecutar por el nuevo subproceso
    public void run()
    {
        // colocar subproceso en inactividad durante el tiempo indicado por
        tiempoInactividad
        try {
            System.err.println(
                getName() + " va a estar inactivo durante " + tiempoInactividad );
            Thread.sleep( tiempoInactividad );
        }
        // si el subproceso se interrumpió durante su inactividad, imprimir rastreo de la pila
        catch ( InterruptedException excepcion ) {
            excepcion.printStackTrace();
        }
        // imprimir nombre del subproceso
        System.err.println( getName() + " termino su inactividad" );
    } // fin del método run
} // fin de la clase ImprimirSubproceso
```