



Programación de servicios y procesos

Act2.1. Fundamentos de la
programación multihilo

Francisco José García Cutillas | 2FPGS_DAM



Índice

Ejercicio 1	3
-------------------	---

Ejercicio 1

Ejecución de run en lugar de start.

- Transcribir el ejemplo de los ratones basado en la implementación de la interface Runnable, ejecutar el programa y observar cómo se genera la salida.

```
1 package com.mycompany.psp_pruebasut02;
2
3
4
5 /**
6  *
7  * @author Fran
8  */
9 public class Raton implements Runnable {
10
11     private String nombre;
12     private int tiempoAlimentacion;
13
14     public Raton(String _nombre, int _tiempoAlimentacion){
15
16         super();
17         this.nombre = _nombre;
18         this.tiempoAlimentacion = _tiempoAlimentacion;
19     }
20
21     public void comer() {
22
23         System.out.println("El ratón " + this.nombre + " ha comenzado a comer");
24         try {
25             Thread.sleep(this.tiempoAlimentacion * 1000);
26             System.out.println("El ratón " + this.nombre + " ha terminado de comer");
27         } catch (InterruptedException ex) {
28             System.out.println("Error: " + ex.getMessage());
29         }
30     }
31
32     @Override
33     public void run() {
34
35         this.comer();
36     }
37
38 }
```

```

package com.mycompany.psp_pruebasut02;

/**
 *
 * @author Fran
 */
public class PSP_pruebasUT02 {

    public static void main(String[] args) {

        Raton fievel = new Raton(_nombre: "Fievel", _tiempoAlimentacion: 4);
        Raton jerry = new Raton(_nombre: "Jerry", _tiempoAlimentacion: 5);
        Raton pinky = new Raton(_nombre: "Pinky", _tiempoAlimentacion: 3);
        Raton mickey = new Raton(_nombre: "Mickey", _tiempoAlimentacion: 6);

        new Thread(task: fievel).start();
        new Thread(task: jerry).start();
        new Thread(task: pinky).start();
        new Thread(task: mickey).start();

        //    fievel.run();
        //    jerry.run();
        //    pinky.run();
        //    mickey.run();
    }
}

```

```

-----< com.mycompany:PSP_pruebasUT02 >-----
Building PSP_pruebasUT02 1.0-SNAPSHOT
-----[ jar ]-----

--- exec-maven-plugin:3.0.0:exec (default-cli) @ PSP_pruebasUT02 ---
El ratón Pinky ha comenzado a comer
El ratón Jerry ha comenzado a comer
El ratón Mickey ha comenzado a comer
El ratón Fievel ha comenzado a comer
El ratónPinky ha terminado de comer
El ratónFievel ha terminado de comer
El ratónJerry ha terminado de comer
El ratónMickey ha terminado de comer

BUILD SUCCESS

Total time:  6.377 s
Finished at: 2023-10-06T18:15:21+02:00

```

- Cambiar las llamadas a los métodos start de los threads por llamadas a los métodos run.

```

1  package com.mycompany.psp_pruebasut02;
2
3  /**
4   *
5   * @author Fran
6   */
7  public class PSP_pruebasUT02 {
8
9      public static void main(String[] args) {
10
11          Raton fievel = new Raton(_nombre: "Fievel", _tiempoAlimentacion: 4);
12          Raton jerry = new Raton(_nombre: "Jerry", _tiempoAlimentacion: 5);
13          Raton pinky = new Raton(_nombre: "Pinky", _tiempoAlimentacion: 3);
14          Raton mickey = new Raton(_nombre: "Mickey", _tiempoAlimentacion: 6);
15
16          //      new Thread(fievel).start();
17          //      new Thread(jerry).start();
18          //      new Thread(pinky).start();
19          //      new Thread(mickey).start();
20
21          fievel.run();
22          jerry.run();
23          pinky.run();
24          mickey.run();
25
26      }
27  }
28

```

- Ejecutar el programa y comparar la manera en la que se genera la salida con la ejecución anterior.

```

-----< com.mycompany:PSP_pruebasUT02 >-----
Building PSP_pruebasUT02 1.0-SNAPSHOT
-----[ jar ]-----

--- exec-maven-plugin:3.0.0:exec (default-cli) @ PSP_pruebasUT02 ---
El ratón Fievel ha comenzado a comer
El ratónFievel ha terminado de comer
El ratón Jerry ha comenzado a comer
El ratónJerry ha terminado de comer
El ratón Pinky ha comenzado a comer
El ratónPinky ha terminado de comer
El ratón Mickey ha comenzado a comer
El ratónMickey ha terminado de comer
-----

BUILD SUCCESS
-----

Total time: 18.401 s
Finished at: 2023-10-06T18:16:16+02:00
-----

```

- **Responder a las siguientes preguntas:**

- **¿Qué diferencias se aprecian en el orden y los tiempos de las salidas de ambas ejecuciones?**

Se puede observar que a la hora de realizar la llamada con la clase Thread y el método start(), estamos iniciando 5 hilos simultáneamente, por lo que los ratones comienzan a comer al mismo tiempo, y van terminando dependiendo del retardo de tiempo que le hayamos puesto en el atributo “tiempo de alimentación”.

Sin embargo, en el caso de la ejecución con el método run de la clase Ratón, los ratones comerían de uno en uno, y hasta que no acabase de comer no empezaría el siguiente. Es decir, en este caso estaríamos ejecutando un hilo solamente a diferencia de los 5 simultáneos del caso anterior.

- **¿A qué se deben dichas diferencias?**

Como se ha descrito en el apartado anterior, la diferencia está en la ejecución de 5 hilos simultáneos con la clase Thread y el método start(), en la que se ejecutan los 5 procesos simultáneamente. Mientras que en el caso de ejecutar el método run() de la clase Ratón, sólo ejecutaríamos un hilo al mismo tiempo, teniendo que esperar los demás a que éste termine su ejecución.