**UD2. Boletín 1. Programación multihilo**

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.

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

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

• Responder a las siguientes preguntas:

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

- ¿A qué se deben dichas diferencias?

**public** **class** Raton **implements** Runnable{

**private** String nombre;

**private** **int** tiempoAlimentacion;

**public** Raton(String nombre, **int** tiempoAlimentacion) {

**super**();

**this**.nombre = nombre;

**this**.tiempoAlimentacion= tiempoAlimentacion;

}

**public** **void** comer () {

**try** {

System. ***out*** .printf ("El ratón %s ha comenzado a alimentarse%n",nombre) ;

Thread.*sleep*(tiempoAlimentacion \* 1000) ;

System.***out***.printf("El ratón %s ha terminado de alimentarse%n",nombre) ;

} **catch** (InterruptedException e) {

e.printStackTrace () ;

}

}

@Override

**public** **void** run () {

**this**.comer () ;

}

**public** **static** **void** main(String[] args) {

Raton fievel = **new** Raton( "Fievel" , 4);

Raton jerry = **new** Raton( "Jerry" , 5);

Raton pinky = **new** Raton( "Pinky" , 3);

Raton mickey = **new** Raton ("Mickey" , 6);

**new** Thread (fievel).start();

**new** Thread (jerry).start();

**new** Thread (pinky).start();

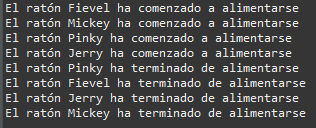
**new** Thread (mickey).start();

}

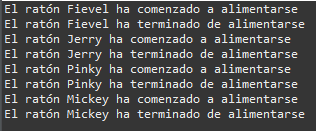
}

**DIFERENCIAS**

En la primera ejecucion, con el metodo start, podemos ver que todos los hilos se ejecutan a la vez y acaban en cuanto el tiempo de alimentacion, previamante definido, termine, teniendo un tiempo de ejecucion de 7 seg.

****

Sin embargo, utilizando el metodo run, podemos observar que los hilos se van ejecutando uno a uno y hasta que no acabe el proceso anterior no se ejecuta el siguiente. Esto nos da un tiempo de ejecucion de 18 seg.

****