Ejemplos de hilos con Java

Ejemplo 1 - herencia

```
public class HelloThread extends Thread {

 public void run() {
  System.out.println("Hello from a thread!");
 }
 public static void main(String args[]) {
  (new HelloThread()).start();
 }
```

```
}

// HelloThread.java
```

Ejemplo 1 - interfaz

```
public class HelloRunnable implements Runnable {
public void run() {
   System.out.println("Hello from a thread!");
}    public static void main(String args[]) {
(new Thread(new HelloRunnable())).start();
 }
}

//HelloRunnable.java
```

Ejemplo 2 - sleep

- Imprime mensajes en intervalos de 4 segundos.

```
public class SleepMessages {
 public static void main(String args[]) throws
   InterruptedException {

   String importantInfo[] = { "Mares eat oats", "Does
   eat oats", "Little lambs eat ivy", "A kid will eat
  ivy too" };

   for (int i = 0; i < importantInfo.length; i++) {
   //Pause for 4 seconds
              Thread.sleep(4000); //Print a message
              System.out.println(importantInfo[i]);
       }
 }
}
```

Ejemplo 3

```
class RunnableDemo implements Runnable
{    private Thread t;    private
String threadName;    RunnableDemo(
String name){        threadName = name;
       System.out.println("Creating " +  threadName );
   }
   public void run() {
      System.out.println("Running " +  threadName
);       try {          for(int i = 4; i > 0; i--)
{
            System.out.println("Thread: " + threadName + ", " + i);
            // Let the thread sleep for a while.
            Thread.sleep(50);
         }
     } catch (InterruptedException e) {
         System.out.println("Thread " +  threadName + "
interrupted.");      }
     System.out.println("Thread " +  threadName + " exiting.");
   }
   public void start (){
      System.out.println("Starting " +  threadName );
if (t == null)
      {
         t = new Thread (this, threadName);
```

```
            t.start ();
        }
    }
}
```

Ejemplo 3, continuación.

```
public class TestThread {    public static
void main(String args[]) {

        RunnableDemo R1 = new RunnableDemo(
  "Thread-1");
        R1.start();

        RunnableDemo R2 = new RunnableDemo(
  "Thread-2");
        R2.start();
    }
}
//TestThread.java
```

Ejemplo 3, continuación.

Salida:

```
~/ProgramasJava$ java TestThread
Creating Thread-1
Starting Thread-1
Creating Thread-2
Starting Thread-2
Running Thread-1
Thread: Thread-1, 4
Running Thread-2
Thread: Thread-2, 4
Thread: Thread-1, 3
Thread: Thread-2, 3
Thread: Thread-1, 2
Thread: Thread-2, 2
Thread: Thread-1, 1
Thread: Thread-2, 1
```

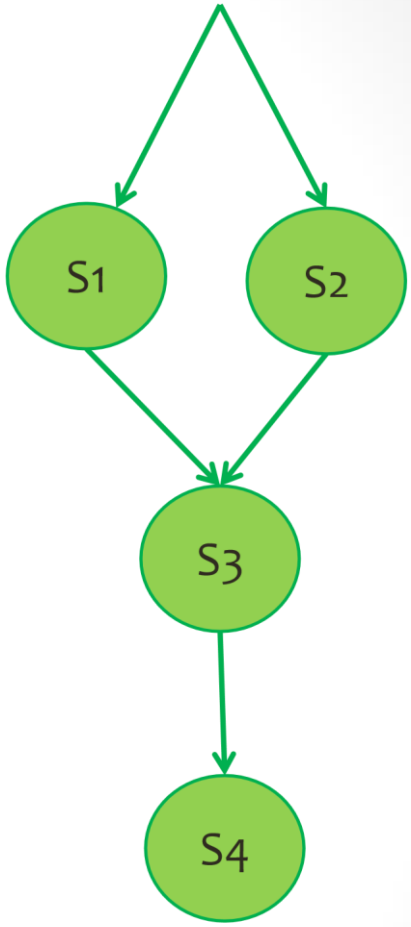
```
Thread Thread-1 exiting.
Thread Thread-2 exiting.
```

Prácticas de laboratorio

- Implemente un programa en java que use hilos independientes. Un hilo imprime números pares del 1 al 10, y otro hilo imprime números impares del 1 al 10. Cree dos instancias (hilos) de cada uno y muestre la salida. Realice el programa utilizando herencia y otro con la interfaz.
- Implemente un programa en java que utilice hilos. El tamaño del vector es desde el main como parámetro. El llenado del vector es aleatorio. El hilo muestra la suma de los elementos de un vector, la suma de los cuadrados de los elementos del vector y la media. Cree dos

instancias con diferente tamaño de vector y presente los resultados.

Ejemplo



S1→a:=x+y;

S2→b:=z-1;

S3→c:=a-b;

- Algoritmo:
- Construya 2 hilos (t1 y t2) que realicen las asignaciones de las instrucciones S1 y S2.
- En el main() se debe lanzar a ejecución t1 y t2 una vez que terminen se realizan las instrucciones S3 y S4.
- El main() debe esperar a que terminen t1 y t2, utilizando la instrucción: join.

```
 //let all threads finish execution before finishing main thread
try {              t1.join();              t2.join();
        } catch (InterruptedException e) {              // TODO
Auto-generated catch block
            e.printStackTrace();
        }
```

Referencias

- Revisar:

- https://docs.oracle.com/javase/tutorial/essential/concurrency/runthread.html
- http://www.tutorialspoint.com/java/java_multithreading.htm