

Hilos

PROGRAMACIÓN DE SERVICIOS Y PROCESOS

Marc Archer Ciscar | Dam | 12/11/2018

# Act 1

**package** ejercicio1;

**import** java.lang.Math;

**public** **class** EjemploThread **extends** Thread{ // La clase deriva de Thread, para definer hilos de ejecución concurrentes.

**int** numero;

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

**try** {

**while** (**true**){

System.*out*.println(**this**.getName());

*sleep*((**long**)(10000\*Math.*random*()));// El programa se duerme un tiempo de entre 0 y 10000 milisegundos.

}

} **catch** (InterruptedException ex) {

**return**;

}

}

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

**for** (**int** i=0;i<10;i++) {

EjemploThread hilos = **new** EjemploThread();//Llama a la clase de arriba.

hilos.start();}//Empieza el programa.

}

}

# Act 2

**package** ejercicio5;

**class** NewThread **implements** Runnable {

String name; //Definimos una variable de tipo String con nombre “name”.

Thread t;

NewThread(String threadname) {

name = threadname;

t = **new** Thread(**this**, name);

System.*out*.println("Nuevo hilo: " + t);

t.start(); //Se ejecuta el codigo.

}

//

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

**try** {

**for**(**int** i = 5; i > 0; i--) {

System.*out*.println(name + ": " + i); //Se imprime el nombre del thread y su prioridad.

Thread.*sleep*(1000);

}

} **catch** (InterruptedException e) {

System.*out*.println(name + "Interrupción del hilo hijo" +name);

}

System.*out*.println("Sale del hilo hijo" + name);

}

}

**class** MultiThreadDemo {

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

**new** NewThread("Uno"); // comienzan los hilos.

**new** NewThread("Dos");

**new** NewThread("Tres");

**try** {

// espera un tiempo para que terminen los otros hilos.

Thread.*sleep*(10000);

} **catch** (InterruptedException e) { // Si hay un error se ejecutara esto.

System.*out*.println("InterrupciÃ³n del hilo principal");

}

System.*out*.println("Sale del hilo principal.");

}

}

# Act 3

**package** ejercicio3;

**class** ThreadDemo **implements** Runnable {

ThreadDemo() {

Thread ct = Thread.*currentThread*();//Obteniene referencia al main principal

Thread t = **new** Thread(**this**);

t.setName("demo Thread"); // cambiar o poner el nombre del hilo

System.*out*.println("hilo actual: " + ct);

System.*out*.println("Hilo creado: " + t);

t.start();

**try** {

Thread.*sleep*(3000);

} **catch** (InterruptedException e) {

System.*out*.println("Interrumpido");

}

System.*out*.println("saliendo del hilo main");

}

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

**try** {

**for** (**int** i = 5; i > 0; i--) { // Imprimira el nombre del hilo que se está ejecutando, hasta que el contador llegue a 0

System.*out*.println(Thread.*currentThread*().getName()+" " + i);

Thread.*sleep*(1000);

}

} **catch** (InterruptedException e) {

System.*out*.println("hijo interrumpido");

}

System.*out*.println("saliendo del hilo hijo");

}

**public** **static** **void** main (String args []) { //Esta es la clase main

ThreadDemo hilo = **new** ThreadDemo();

Thread h = **new** Thread(hilo);

}

}

# Act 4

Thread t1=new Thread((new ThreadAct4(),”hola”).start);

Thread t2=new Thread((new ThreadAct4(),”adios”).start);

Public void run(){

Try{

For(int i=0; i<5;i++){

System.out.println(Thread.currentThread().getName());

Thread.currentThread().sleep(600);

}

}catch(interruptedExcepcion e){

e.printStackTrace();

}

# Act 5

Public void run(){

For(int i=0;i<5;i++){

System.out.println(Thread.currentThread().getName()+1);

Try{

Thread.currentThread().sleep(600);

}catch (InterruptedExcepted e){

e.printStackTrace();

}

}

Public static void main(){

Thread t1=new Thread((new ThreadAct45(),”1”).start);

Thread t2=new Thread((new ThreadAct45(),”2”). start);

T1.setName(“bienvenido, soy el hilo1 y estoy en la iteración ””);

T2.setName(“soy el hilo 2 y es mi iteración número ”);

}

}