

Programacion De servicios y procesos

Actividad 1 tema 2



carles Ricart ramon

# Ejercicio 1: Indica qué hace el programa siguiente usando los espacios reservados para los comentarios y cuál será su salida por pantalla.

**package** ejercicio1;

**import** java.lang.Math;

**public** **class** EjemploThread **extends** Thread{

**int** numero;

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

**try** {

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

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

*sleep*((**long**)(10000\*Math.*random*()));

}

} **catch** (InterruptedException ex) {

**return**;

}

}

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

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

EjemploThread hilos = **new** EjemploThread();

hilos.start();}

}

}

## Salida por pantalla ejer 1

Thread-0

Thread-2

Thread-4

Thread-1

Thread-3

Thread-5

Thread-6

Thread-8

Thread-7

Thread-9

Thread-5

Thread-9

Thread-7

Thread-2

Thread-2

Thread-3

Thread-1

Thread-4

Thread-0

Thread-7

Thread-7

Thread-0

Thread-5

Thread-8

Thread-9

Thread-6

# Ejercicios 2: Indica qué hace el programa y cuál será su salida por pantalla.

**package** ejercicio2;

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

String name; // nombre del hilo

Thread t;

NewThread(String threadname) {

name = threadname;

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

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

t.start(); // Comienza el hilo

}

// Este es el punto de entrada del hilo.

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

**try** {

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

System.*out*.println(name + ": " + i);

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) {

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

}

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

}

}

## Salida por pantalla ejer 2

Nuevo hilo: Thread[Uno,5,main]

Nuevo hilo: Thread[Dos,5,main]

Uno: 5

Nuevo hilo: Thread[Tres,5,main]

Tres: 5

Dos: 5

Dos: 4

Tres: 4

Uno: 4

Tres: 3

Uno: 3

Dos: 3

Tres: 2

Uno: 2

Dos: 2

Tres: 1

Uno: 1

Dos: 1

Sale del hilo hijoTres

Sale del hilo hijoDos

Sale del hilo hijoUno

Sale del hilo principal.

# Ejercicios 3: Indica qué hace el programa y cuál será su salida por pantalla.

**package** ejercicio3;

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

ThreadDemo() {

Thread ct = Thread.*currentThread*();

//Thread t = new Thread(this, "demo Thread");

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

t.setName("demo Thread");

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--) {

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 []) {

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

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

//h.start();

//new ThreadDemo();

}

}

## Salida por pantalla ejer 3

hilo actual: Thread[main,5,main]

Hilo creado: Thread[demo Thread,5,main]

demo Thread 5

demo Thread 4

demo Thread 3

saliendo del hilo main

demo Thread 2

demo Thread 1

saliendo del hilo hijo

# Ejercicio 5. Crea utilizando la clase Thread primero y extendiendo de la interfaz Runnable después , un programa que conste de dos hilos, los cuales se ejecuten concurrentemente e intercaladamente, de tal manera que el hilo 1 muestre por pantalla la palabra “hola” y el hilo 2 muestre por pantalla la palabra “adiós” durante 5 ciclos.

public class ThreadEjemplo extends Thread {

public ThreadEjemplo(String str) {

super(str);

}

public void run() {

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

System.out.println(i + " " + getName());

System.out.println("Termina thread " + getName());

}

public static void main (String [] args) {

new ThreadEjemplo("Pepe").start();

new ThreadEjemplo("Juan").start();

System.out.println("Termina thread main");

}

public class ThreadEjemplo implements Runnable {

public void run() {

for (int i = 0; i < 5 ; i++)

System.out.println(i + " " +

Thread.currentThread().getName());

System.out.println("Termina thread " +

Thread.currentThread().getName());

}

public static void main (String [] args) {

new Thread ( new ThreadEjemplo(), Pepe").start();

new Thread ( new ThreadEjemplo() , "Juan").start();

System.out.println("Termina thread main");

}

}

# Ejercicio 6. Crea utilizando primero la clase Thread y después extendiendo de la interfaz runnable , dos hilos, los cuales se ejecuten concurrentemente e intercaladamente, de tal manera que el hilo 1 muestre por pantalla la palabra “bienvenido, soy el hilo1 y estoy en la iteración %” y el hilo 2 muestre pon pantalla “soy el hilo 2 y es mi iteración número %”, durante 10 ciclos, y que una vez finalizados muestre un mensaje que diga ”Finaliza el hilo\*\*” y además una vez finalizados muestre el mensaje “main terminado”.

**public** **class** ThreadSaludo **extends** Thread

{

String nombre;

**public** ThreadSaludo(String nombre)

{

**this**.nombre = nombre;

//return null;

}

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

{

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

**try**

{

Thread.*sleep*(10);

System.*out*.println(nombre+ i);

} **catch** (InterruptedException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();//se muestra por la salida estándar la traza de errores que se han producido

}

}

}

**public** **static** **void** main (String[] args) **throws** InterruptedException

{

ThreadSaludo t1 = **new** ThreadSaludo("Bienvenido a mi iteración número");

ThreadSaludo t2 = **new** ThreadSaludo("soy el hilo 2 y es mi ciclo número");

t1.start();

t2.start();

**try**

{

t1.join();

t2.join();

System.*out*.println("Finaliza el hilo "+ t1.getName());

System.*out*.println("Finaliza el hilo "+ t2.getName());

} **catch** (InterruptedException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

System.*out*.println("Main Terminado");

}

}



