HILOS JAVA THREADS

* Un proceso es aquel que se ejecuta en el sistema operativo y consume los siguientes recursos:

Memoria

MicroProcesador

Heap(mem. monton)

SegmentoDeCodigo

Stack(Pila)

Proceso

Proceso pesado

* Un hilo es un proceso ligero porque solo consume los siguientes recursos:

Heap

Hilo

Recurso compartido

Red, archivos, base de datos

* La representación gráfica de un hilo

Hilo 3

Hilo 2

Hilo 1

* Programación concurrente con hilos. p = procesador

Hilo 1

Hilo 2

p

Hilo 3

* Programación paralela con hilos. p1, p2, p3 = procesadores

Hilo 1

P1

Hilo 2

P3

P2

Hilo 3

* Jerarquía de hilos

UnaClase

Proceso

MiHilo

MiHilo

Thread

Object

Extends extends implements

* La clase Thread

getName() isAlive()

setName() setPriority()-----de 1 a 10

currentThread() getPriority()

static

sleep()----tiempo ms MIN\_PRIORITY-----1

join() NORM\_PRIORITY-----5

resume() MAX\_PRIORITY------10

suspend() setDaemon()------true/false, sirve cuando se usan

start() aplicaciones web para poder matar los hilos

run() cuando el contenedor web se apague.

stop() despreciados

destroy() deprecate

* La interfaz Runnable

run()

* Ciclo de vida del hilo
* Creando hilos

public class MiHilo extends Thread{

private int contador;

private MiHilo(){

this(10,”Mi hilo”);

}

public MiHilo(int contador,String nombre){

super(nombre);

this.contador = contador;

}

public void run(){

try{

for(int n = contador;n >= 0;n--){

System.out.println(n + “:” + getName());

Thread.sleep(3000);

}

}catch(InterruptedException ie){

ie.printStackTrace();

}

}

}

La clase anterior se compila con la clase que viene comentada, después de compilarla y ejecurla descomentamos y volvemos a compilar, a ver como se compoeta

/\*

public class MiHilo extends Object implements Runnable{

private Thread hilo;

private int contador;

public MiHilo(){

this(10,”Mi hilo”);

}

public MiHilo(int contador, String nombre){

hilo = new Thread(this,nombre);

this.contador = contador;

}

public void start(){

hilo.start();

}

public void setPriority(int p){

hilo.setPriority(p);

}

public int getPriority(){

return hilo.getPriority();

}

public void run(){

try{

for(int n = contador;n >= 0;n--){

System.out.println(n + “:” + hilo.getNombre);

hilo.sleep(3000);

}

}catch(InterruptedExceptio ie){

Ie.printStackTrace();

}

}

}

\*/

public class CreadorDeHilo{

public static void main(String [] args){

int c = 10;

Thread principal = Thread.currentThread();

System.out.println(principal.getName);

Principal.setName(“Principal”);

MiHilo hilo1 = new MiHilo(c,”Hilo 1”);

MiHilo hilo2 = new MiHilo(c,”Hilo 2”);

MiHilo hilo3 = new MiHilo(c,”Hilo 3”);

hilo1.start(); hilo1.setPriority(2);

hilo2.start(); hilo2.setPriority(4);

hilo3.start(); hilo3.setPriority(8);

try{

for(int n = c;n >= 0;n--){

System.out.println(n + “:” + principal.getName());

Principal.sleep(3000 \* c \* 3);

}

}catch(InterruptedException ie){

Ie.printStackTrace();

}

}

}