



Benemérita Universidad Autónoma De Puebla



Facultad de Ciencias de la Computación

Materia:

Programación Concurrente y Paralela (25581)

Título:

Actividad 1, 2 y 3

Profesora:

Carmen Cerón Garnica

Alumno:

Jared Rai Serrano Navarro (202160998)

Periodo:

Otoño 2025

Actividad 1: Agregar dos hilos más

CODIGO: ACTIVIDAD 1 THREAD CON HERENCIA

```
class ThreadConHerencia extends Thread
{
    String palabra;
    public ThreadConHerencia (String _palabra) {
        palabra = _palabra;
    }
    public void run( ) {
        for (int i=0; i<10; i++)
            System.out.println (palabra);
    }

    public static void main(String[] args) {
        Thread a = new ThreadConHerencia ("hiloUno ");
        Thread b = new ThreadConHerencia ("hiloDos ");
        Thread c= new ThreadConHerencia ("hiloTres ");
        Thread d = new ThreadConHerencia ("hiloCuatro ");
        Thread e = new ThreadConHerencia ("hiloCinco ");

        a.start();
        b.start();
        c.start();
        d.start();
        e.start();

        System.out.println ("Hilo Uno"+ " "+a.getName());
        System.out.println ("Hilo Dos"+ " "+b.getName());
        System.out.println ("Hilo Tres"+ " "+c.getName());
        System.out.println ("Hilo Cuatro"+ " "+d.getName());
        System.out.println ("Hilo Cinco"+ " "+e.getName());
        System.out.println ("Fin del hilo principal");
    }
}
```

CODIGO: ACTIVIDAD 1 THREAD CON RUNNABLE

```
class ThreadConRunnable implements Runnable{

    String palabra;

    public ThreadConRunnable (String _palabra)
    {
        palabra = _palabra;
    }

    public String par(int num)
    {
        return (num %2==0) ? "par":"impar";
    }
    public void run()
```

```

{
    for (int i = 0;i<10;i++)
        System.out.println (palabra + " "+ i + " "+ par(i));
}

public static void main (String args []) {
    ThreadConRunnable a = new ThreadConRunnable ("hiloUno");
    ThreadConRunnable b = new ThreadConRunnable ("hiloDos");
    ThreadConRunnable c = new ThreadConRunnable ("hiloTres");
    ThreadConRunnable d = new ThreadConRunnable ("hiloCuatro");
    ThreadConRunnable e = new ThreadConRunnable ("hiloCinco");

    Thread t1 = new Thread (a);
    Thread t2 = new Thread (b);
    Thread t3 = new Thread (c);
    Thread t4 = new Thread (a);
    Thread t5 = new Thread (b);

    t1.start();
    t2.start();
    t3.start();
    t4.start();
    t5.start();

    t1.setPriority(10); // Prioridad máxima
    t2.setPriority(5);
    t3.setPriority(3);
    System.out.println("-----Hilos con Runnable-----");
    System.out.println ("prioridad hiloUno"+ " "+t1.getName()+" "+t1.getPriority());
    System.out.println ("prioridad hilodos"+t2.getName()+t2.getPriority());
    System.out.println ("prioridad hiloUno"+t3.getName()+" "+t3.getPriority());
    System.out.println("Prioridad hiloCuatro: "+t4.getName()+" "+t4.getPriority());

    System.out.println ("Fin del hilo principal");
}

```

Salida thread con herencia:

```
hiloCinco
hiloCinco
hiloCinco
hiloCinco
hiloCinco
hiloCinco
hiloTres
hiloCuatro
hiloUno
hiloUno
hiloUno
hiloUno
hiloUno
hiloUno
hiloUno
hiloUno
hiloUno
hiloTres
hiloTres
hiloDos
hiloDos
Hilo Uno Thread-0
Hilo Dos Thread-1
Hilo Tres Thread-2
Hilo Cuatro Thread-3
Hilo Cinco Thread-4
Fin del hilo principal
```

```
Process finished with exit code 0
```

Salida thread con runnable:

```
C:\Tools\Java\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2025.2\lib\idea_rt.jar=-Xmx1024m" -Dfile.encoding=UTF-8 -jar C:\Users\josep\OneDrive\Documentos\NetBeansProjects\Hilos\dist\bin\Hilos.jar
-----
-----Hilos con Runnable-----
prioridad hiloUno Thread-0 10
prioridad hilodosThread-15
prioridad hiloUnoThread-23
Prioridad hiloCuatro: Thread-3 5
Fin del hilo principal
hiloUno 0 par
hiloUno 0 par
hiloTres 0 par
hiloUno 1 impar
hiloUno 1 impar
hiloUno 2 par
hiloTres 1 impar
hiloDos 0 par
hiloUno 3 impar
hiloUno 2 par
hiloDos 0 par
hiloUno 4 par
hiloDos 1 impar
hiloTres 2 par
hiloDos 2 par
hiloUno 5 impar
hiloDos 1 impar
hiloUno 3 impar
hiloDos 2 par
hiloUno 6 par
hiloDos 3 impar
hiloTres 3 impar
hiloDos 4 par
hiloUno 7 impar
hiloDos 3 impar
hiloUno 4 par
hiloDos 4 par
hiloUno 8 par
hiloDos 5 impar
hiloTres 4 par
hiloDos 6 par
```

```
hiloUno 9 impar
hiloDos 5 impar
hiloUno 5 impar
hiloDos 6 par
hiloDos 7 impar
hiloTres 5 impar
hiloDos 8 par
hiloDos 7 impar
hiloUno 6 par
hiloDos 8 par
hiloDos 9 impar
hiloTres 6 par
hiloDos 9 impar
hiloUno 7 impar
hiloTres 7 impar
hiloUno 8 par
hiloTres 8 par
hiloUno 9 impar
hiloTres 9 impar

Process finished with exit code 0
```

Actividad 2: Agregar decrementar

CODIGO: HERENCIA

Hilo1.java

```
package Herencia;

public class Hilo1 extends Thread
{
    private RecursoCompartido n;
    private String nombre;

    public Hilo1(RecursoCompartido numero, String nombre1) {
        // TODO Auto-generated constructor stub
        n=numero;
        nombre=nombre1;
    }

    public void run(){
        for(int i=1; i<=10; i++){
            n.incrementar(nombre);
        }
    }
}
```

```

//decrementar
for(int i=1; i<=10; i++){
    n.decrementar(nombre);
}
}

Hilo2.java

package Herencia;

public class Hilo2 extends Thread {

    private RecursoCompartido n;
    private String nombre;

    public Hilo2(RecursoCompartido numero, String nombre1) {
        // TODO Auto-generated constructor stub
        n=numero;
        nombre=nombre1;
    }
    public void run(){
        for(int i=1; i<=10; i++){
            n.incrementar(nombre);
        }
        //decrementar
        for(int i=1; i<=10; i++) {
            n.decrementar(nombre);
        }
    }
}

```

PrincipalHerencia.java

```

package Herencia;

public class PrincipalHerencia {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        RecursoCompartido n=new RecursoCompartido();
        Hilo1 h1=new Hilo1(n,"Hilo A");
        Hilo2 h2=new Hilo2(n,"Hilo B");

        h1.start();
        h2.start();
    }
}

```

RecursoCompartido.java

```
package Herencia;

public class RecursoCompartido{
    private int N;
    public RecursoCompartido(){
        N=0;
    }
    public synchronized void incrementar(String Nombre){
        N=N+1;
        System.out.println ( "Nombre: "+ Thread.currentThread().getName() + " "+ Nombre + " numero
= " + N );
    }
    public synchronized void decrementar(String Nombre){
        N=N-1;
        System.out.println ( "Nombre: "+ Thread.currentThread().getName() + " "+ Nombre + " numero
= " + N );
    }
}
```

CODIGO: INTERFACER

Hilo1R.java

```
package InterfaceR;

public class Hilo1R implements Runnable {
    private RecursoCompartidoR numero;
    private String nombreh;

    public Hilo1R(RecursoCompartidoR numero1, String nombre1) {
        numero = numero1;
        nombreh = nombre1;
    }

    public void run() {
        for (int i = 1; i <= 10; i++) {
            numero.incrementar(nombreh);
        }

        //decrementar
        for (int i = 1; i <= 10; i++) {
            numero.decrementar(nombreh);
        }
    }
}
```

Hilo2R.java

```
package InterfaceR;

public class Hilo2R implements Runnable {
```

```

private RecursoCompartidoR numero;
private String nombreh;

public Hilo2R(RecursoCompartidoR numero1, String nombre1) {
    // TODO Auto-generated constructor stub
    numero=numero1;
    nombreh=nombre1;
}
public void run(){
    for(int i=1; i<=10; i++){
        numero.incrementar(nombreh);
    }

    //decrementar
    for(int i=1; i<=10; i++) {
        numero.decrementar(nombreh);
    }
}
}

```

PrincipalRunnable.java

```

package InterfaceR;

public class PrincipalRunnable {

    public static void main(String[] args)
    {

        RecursoCompartidoR recurso= new RecursoCompartidoR();
        Hilo1R objeto1 =new Hilo1R (recurso,"hiloA");
        Hilo2R objeto2 =new Hilo2R (recurso,"hiloB");

        Thread h1= new Thread(objeto1);
        Thread h2= new Thread(objeto2);
        h1.start();
        h2.start();

    }
}

```

RecursoCompartidoR.java

```

package InterfaceR;

public class RecursoCompartidoR{
    private int N;
    public RecursoCompartidoR ()
    {
        N=0;
    }
}

```

```

public synchronized void incrementar(String Nombre){
    N=N+1;
    System.out.println ( "Nombre: "+ Thread.currentThread().getName() + " "+ Nombre + " numero
= " + N );
}

public synchronized void decrementar(String Nombre){
    N=N-1;
    System.out.println ( "Nombre: "+ Thread.currentThread().getName() + " "+ Nombre + " numero
= " + N );
}

```

Salida Herencia:

```

C:\Tools\Java\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2025.2\lib\idea_rt.ja
Nombre: Thread-0 Hilo A numero = 1
Nombre: Thread-0 Hilo A numero = 2
Nombre: Thread-0 Hilo A numero = 3
Nombre: Thread-0 Hilo A numero = 4
Nombre: Thread-0 Hilo A numero = 5
Nombre: Thread-0 Hilo A numero = 6
Nombre: Thread-0 Hilo A numero = 7
Nombre: Thread-0 Hilo A numero = 8
Nombre: Thread-0 Hilo A numero = 9
Nombre: Thread-0 Hilo A numero = 10
Nombre: Thread-1 Hilo B numero = 11
Nombre: Thread-1 Hilo B numero = 12
Nombre: Thread-1 Hilo B numero = 13
Nombre: Thread-1 Hilo B numero = 14
Nombre: Thread-1 Hilo B numero = 15
Nombre: Thread-1 Hilo B numero = 16
Nombre: Thread-1 Hilo B numero = 17
Nombre: Thread-1 Hilo B numero = 18
Nombre: Thread-1 Hilo B numero = 19
Nombre: Thread-1 Hilo B numero = 20
Nombre: Thread-1 Hilo B numero = 19
Nombre: Thread-1 Hilo B numero = 18
Nombre: Thread-1 Hilo B numero = 17
Nombre: Thread-1 Hilo B numero = 16
Nombre: Thread-1 Hilo B numero = 15
Nombre: Thread-1 Hilo B numero = 14
Nombre: Thread-1 Hilo B numero = 13
Nombre: Thread-1 Hilo B numero = 12
Nombre: Thread-1 Hilo B numero = 11
Nombre: Thread-1 Hilo B numero = 10
Nombre: Thread-0 Hilo A numero = 9
Nombre: Thread-0 Hilo A numero = 8
Nombre: Thread-0 Hilo A numero = 7
Nombre: Thread-0 Hilo A numero = 6
Nombre: Thread-0 Hilo A numero = 5
Nombre: Thread-0 Hilo A numero = 4
Nombre: Thread-0 Hilo A numero = 3

```

```
Nombre: Thread-0 Hilo A numero = 2
Nombre: Thread-0 Hilo A numero = 1
Nombre: Thread-0 Hilo A numero = 0
```

Salida Runnable:

```
C:\Tools\Java\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2025.2\lib\idea_rt.jar=52227" -Dfile.encoding=UTF-8 -jar C:\Users\josem\OneDrive\Documentos\NetBeansProjects\Hilos\dist\Hilos-1.0-SNAPSHOT.jar
Nombre: Thread-0 hiloA numero = 1
Nombre: Thread-0 hiloA numero = 2
Nombre: Thread-0 hiloA numero = 3
Nombre: Thread-0 hiloA numero = 4
Nombre: Thread-0 hiloA numero = 5
Nombre: Thread-0 hiloA numero = 6
Nombre: Thread-0 hiloA numero = 7
Nombre: Thread-0 hiloA numero = 8
Nombre: Thread-0 hiloA numero = 9
Nombre: Thread-0 hiloA numero = 10
Nombre: Thread-1 hiloB numero = 11
Nombre: Thread-1 hiloB numero = 12
Nombre: Thread-1 hiloB numero = 13
Nombre: Thread-1 hiloB numero = 14
Nombre: Thread-1 hiloB numero = 15
Nombre: Thread-1 hiloB numero = 16
Nombre: Thread-1 hiloB numero = 17
Nombre: Thread-1 hiloB numero = 18
Nombre: Thread-1 hiloB numero = 19
Nombre: Thread-1 hiloB numero = 20
Nombre: Thread-0 hiloA numero = 19
Nombre: Thread-0 hiloA numero = 18
Nombre: Thread-0 hiloA numero = 17
Nombre: Thread-0 hiloA numero = 16
Nombre: Thread-0 hiloA numero = 15
Nombre: Thread-0 hiloA numero = 14
Nombre: Thread-0 hiloA numero = 13
Nombre: Thread-0 hiloA numero = 12
Nombre: Thread-0 hiloA numero = 11
Nombre: Thread-0 hiloA numero = 10
Nombre: Thread-1 hiloB numero = 9
Nombre: Thread-1 hiloB numero = 8
Nombre: Thread-1 hiloB numero = 7
Nombre: Thread-1 hiloB numero = 6
Nombre: Thread-1 hiloB numero = 5
Nombre: Thread-1 hiloB numero = 4
Nombre: Thread-1 hiloB numero = 3
Nombre: Thread-1 hiloB numero = 2
Nombre: Thread-1 hiloB numero = 1
Nombre: Thread-1 hiloB numero = 0
```

Actividad 3: Autorun

CODIGO: ACTIVIDAD 3

```
class AutoThread implements Runnable {  
    private Thread mio; // atributo para manejar el hilo  
    private String nombre;  
  
    public AutoThread() {  
        // Se crea un hilo que ejecuta este mismo objeto  
        mio = new Thread(this);  
        nombre = "mio";  
        mio.start(); // <<<--- AUTORUN: el hilo inicia automáticamente  
    }  
  
    // Constructor con parámetro (se le pasa un nombre)  
    public AutoThread(String nom) {  
        nombre = nom;  
        mio = new Thread(this);  
        mio.start(); // <<<--- AUTORUN: también arranca solo  
    }  
  
    public void run() {  
        // Verifica que el hilo que está corriendo sea el mismo que creamos en el objeto  
        if (mio == Thread.currentThread()) {  
            for (int i = 0; i < 10; i++) {  
                System.out.println("Dentro del Autorun() i=" + i + " " + mio.getName());  
            }  
        }  
    }  
  
    public static void main(String[] args) throws Exception {  
        // Ambos arrancan automáticamente por el "autorun" en el constructor  
        AutoThread miThred = new AutoThread("Autorun");  
        AutoThread miThred1 = new AutoThread();  
  
        Thread.sleep(10);  
  
        // Se crea un tercer hilo (Thread w) a partir del primer objeto (miThred)  
        Thread w = new Thread(miThred);  
        w.start();  
  
        for (int i = 0; i < 10; i++) {  
            System.out.println("Dentro del main " + w.getName() + " " + miThred.nombre);  
            System.out.println("Dentro del main " + w.getName() + " " + miThred1.nombre);  
        }  
    }  
}
```

Salida Autorun: