

1. ¿Por qué se pone un `InterruptedException` en el método `main`?
Porque es la forma de desbloquear un hilo si éste está suspendido o esperando a que ocurra algo.
2. ¿Para que sirve el método `Join`?
Sirve para mantener la ejecución del hilo que actualmente se ejecuta hasta que éste esté muerto o finalizado.
3. ¿Qué pasa si no le hacemos `Join` a los hilos?
Podemos inicial los hilos de forma secuencial pero no los podemos finalizar en un orden específico, si no usamos `join` entonces cada vez que los finalizamos, tendrán un resultado diferente.
4. Explica de manera consisa como usar Hilos extendiendo la clase `Thread`.
Extendemos una subclase de la clase `Thread` para después sobrescribir el método `run()` de la clase `Thread`, después creamos un objeto de la subclase y con el método `start()` del objeto instanciado inicia la ejecución del thread.
5. ¿Cuáles son las ventajas en implementar `Runnable` contra extender de `Thread`?
Cuando implementamos `Runnable` ahorramos mucho espacio para que nuestra clase pueda extender a cualquier otra clase en cualquier momento, cosa que con `Thread` no se puede. También con `Runnable` podemos compartir el mismo objetos con varios subprocesos.
6. ¿Se puede predecir el orden en el que se imprimira el mensaje de la clase `Hilos`?
No es posible ya que cada hilo termina su proceso de forma aleatoria.
7. En el archivo `Hilos2.java`, ¿Qué pasa si sacamos la instancia de la clase "h" de `t1`, es decir, poner `h` por ejemplo, antes de declarar `t1`?
No pasa nada, sigue imprimiendo lo mismo ya que tenemos el método `run` el cual realiza la llamada al método que imprime el mensaje.
8. Explica como podriamos tener comportamientos diferentes implementando `Runnable`.
Si implementamos `Runnable` entonces el comportamiento del hilo es muy generico por lo cual no es posible realizarlo de esta forma.

Imagenes de la modificacion a Hilos.java

```
1  /**
2   * Clase que ejemplifica los Hilos implementando Runnable
3   * @author Sunny Mirael modificado por Jose Marin
4   * @version 1.1
5   */
6  import java.util.ArrayList;
7
8  public class Hilos implements Runnable {
9
10     @Override
11     public void run() { //Sobrescribimos el metodo run
12         System.out.println("Hola soy el: " + Thread.currentThread().getName()); //Pedimos el nombre del hilo pidiendo primero que se seleccione el hilo
13     }
14
15     public static void main(String[] args) throws InterruptedException {
16         ArrayList lista = new ArrayList(); //Creamos la lista de los hilos
17         Hilos h = new Hilos(); //Se crea una instancia de la clase
18         Thread t1 = new Thread(h, "Hilo 1"); //Creamos un hilo, le pasamos de parametro la instancia de la clase y un nombre
19         Thread t2 = new Thread(h, "Hilo 2");
20         Thread t3 = new Thread(h, "Hilo 3");
21         Thread t4 = new Thread(h, "Hilo 4");
22         Thread t = new Thread(); //Creamos una instancia de Thread para usarla
23         //Agregamos los hilos existentes en la lista
24         lista.add(t1);
25         lista.add(t2);
26         lista.add(t3);
27         lista.add(t4);
28         for (int i=0; i<=10; i++) { //Ciclo for para crear, iniciar y agregar los nuevos hilos a la lista
29             t = new Thread(h, "Hilo "+i);
30             t.start();
31             t.join();
32             lista.add(t);
33         }
34
35         t1.start(); t2.start(); t3.start(); t4.start(); //Se inicializan los hilos para comenzar su ejecucion
36
37         t1.join(); t2.join(); t3.join(); t4.join(); //????
38     }
39 }
```

Codigo modificado del archivo Hilos.java

```
→ Practical javac Hilos.java
Note: Hilos.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
→ Practical java Hilos
Hola soy el: Hilo 0
Hola soy el: Hilo 1
Hola soy el: Hilo 2
Hola soy el: Hilo 3
Hola soy el: Hilo 4
Hola soy el: Hilo 5
Hola soy el: Hilo 6
Hola soy el: Hilo 7
Hola soy el: Hilo 8
Hola soy el: Hilo 9
Hola soy el: Hilo 10
Hola soy el: Hilo 1
Hola soy el: Hilo 2
Hola soy el: Hilo 3
Hola soy el: Hilo 4
```

Resultado de ejecutar el archivo Hilos.java

Referencias.

¿Cómo un hilo puede interrumpir otro hilo en Java? – Acervo Lima. (s/f). Acervolima.com. Recuperado el 16 de agosto de 2022, de <https://es.acervolima.com/como-un-hilo-puede-interrumpir-otro-hilo-en-java/>

El método join() en Java con el ejemplo. (2020, septiembre 30). Tecnologías Moviles. <https://tecnologiasmoviles.com/el-metodo-join-en-java-con-el-ejemplo/>

Thread (java platform SE 7). (2020, junio 24). Oracle.com. <https://docs.oracle.com/javase/7/docs/api/java/lang/Thread.html>