1. Implementar un programa en java que use hilos independientes, en clases diferentes. Un hilo imprime números pares del 1 al 10 y la suma de dichos números, y otro hilo que imprime números impares del 1 al 10. Cree dos instancias (hilos), uno de cada clase, y muestre la salida.

La clave para la **sincronización en Java** es el concepto de **monitor**, que controla el acceso a un objeto. Un monitor funciona implementando el concepto de **bloqueo** (lock). Cuando un objeto está bloqueado por un hilo, ningún otro hilo puede obtener acceso al objeto. Cuando el hilo sale, el objeto está desbloqueado y está disponible para ser utilizado por otro hilo.

**Todos los objetos en Java tienen un monitor**. Esta característica está integrada en el lenguaje Java en sí. Por lo tanto, **todos los objetos se pueden sincronizar.** La sincronización está respaldada por la palabra clave **synchronized.**

Puede sincronizar el acceso a un método modificándolo con la palabra clave synchronized. Cuando se llama a ese método, el hilo de llamada entra en el monitor del objeto, que luego bloquea el objeto.

* Mientras está bloqueado, ningún otro hilo puede ingresar al método, o ingresar cualquier otro método sincronizado definido por la clase del objeto.
* Cuando el hilo retorna del método, el monitor desbloquea el objeto, permitiendo que sea utilizado por el siguiente hilo. Por lo tanto, la sincronización se logra con prácticamente ningún esfuerzo de programación de tu parte.

1. Define que hace el siguiente programa:

import java.io.BufferedReader;

import java.io.File;

import java.io.FileReader;

import java.io.IOException;

public class BuscarPalabra implements Runnable {

private String palabra;

private Thread hilo;

private int cant;

public BuscarPalabra(String palabra) {

this.palabra = palabra;

hilo = new Thread(this);

hilo.start();

while (hilo.isAlive());

System.out.println("La palabra " + palabra + " se encuentra contenida en " + cant + " archivos");

}

@Override

public void run() {

File ar = new File("C:\\documentos\\");

String[] directorio = ar.list();

for (String arch : directorio) {

if (tiene(arch))

cant++;

}

}

private synchronized boolean tiene(String archivo) {

boolean existe = false;

try {

FileReader fr = new FileReader("c:\\documentos\\" + archivo);

BufferedReader br = new BufferedReader(fr);

String linea = br.readLine();

while (linea != null) {

if (linea.indexOf(palabra) != -1)

existe = true;

linea = br.readLine();

}

br.close();

fr.close();

} catch (IOException e) {

System.out.println(e.toString());

}

return existe;

}

public static void main(String[] ar) {

new BuscarPalabra("rojo");

new BuscarPalabra("verde");

}

}

1. Crear una clase llamada Counter que contenga dos métodos, uno con el modificador synchronized y otro sin este. El método solamente mostrará los números entre 0 y n.

Después, crear una clase **HiloContador** que herede de **Thread**. Tendrá como atributos un id para identificar el hilo, un número n que indica hasta dónde debe mostrar por pantalla y una instancia de la clase **Counter**.

Posteriormente en la clase Principal, crearemos dos hilos y los ejecutaremos para ver las diferencias entre el método sincronizado y el que no.

1. Crea una clase sumArray que contenga un método synchronized sumArray que sume los elementos de un array, una clase MiHilo con el constructor y el método run y una clase sincronización en la cual crear dos hilos para probar la suma.