Bloque 3. Paradigmas de la Programación

Tarea 1: Programa que simule el problema del sensor: Primera versión

A tener en cuenta:

- Usar únicamente locks explícitos.
- Es un productor-consumidor.
- Coordinación pura de comunicación: necesitaremos de variables compartidas. En este caso usaremos un ArrayList<Integer>.
- Tendremos 4 clases: Main, 2 hilos (el productor y el consumidor) y por último la clase compartida Temperatura.
- Main: crea el objeto compartido, el productor y el consumidor, y los lanza .
- Temperatura: objeto compartido con operaciones de leerTemperatura y mostrarTemperatura.
- Productor: genera una temperatura y la inserta en el ArrayList.
- Consumidor: consume una temperatura y la elimina del ArrayList.

Observaciones:

Dado que tenemos un productor-consumidor, para que el programa funcione hay que asegurar que el productor genera algo antes de que el consumidor lo quiera obtener. Como no podemos hacer uso de conditions (estos serían los encargados de comprobar que podemos consumir siempre y cuando haya elementos disponibles) haremos uso de un join, de forma que el productor sea el primero que empiece y después lo siga el consumidor.

Si no lo hacemos de esta manera, e iniciamos ambos hilos a la vez, el programa nos devolverá un error *Exception in thread "Thread-0" java.lang.IndexOutOfBoundsException: Index: 0, Size: 0* indicando que el consumidor no puede acceder al elemento del ArrayList y eliminarlo dado que no hay nada, el productor no ha tenido tiempo de generar la temperatura e insertarla.

Código Clase Main:

```
public class Main {

   public static void main(String[] args) {
      Temperatura temp = new Temperatura(10); //Clase común al productor y consumidor
      Consumidor c = new Consumidor(temp);
      Productor p = new Productor(temp);

      p.start();

      try {

         p.join();
         c.start();
      } catch (Exception e) {
      }
   }

}
```

Código Clase Productor:

```
public class Productor extends Thread {

   private Temperatura temp;

   public Productor(Temperatura temp) {
      this.temp = temp;
   }

   public void run() {
      int temperatura;
      for (int i = 0; i <= 20; i++) { //Vamos a leer 20 temperaturas
         try {
            //Vamos a generar la temperatura de forma aleatoria entre 1 y 100ªC
            temperatura = (100 + (int) (-100 * Math.random()));
            //Metodo leer temperatura de la clase compartida
            temp.leerTemperatura(temperatura);
         } catch (InterruptedException e) {
         }
      }

   }

}
```

Código Clase Consumidor:

```
public class Consumidor extends Thread {

   private Temperatura temp;

   public Consumidor(Temperatura temp) {
      this.temp = temp;
   }

   public void run() {
      int temperatura;
      for (int i = 0; i <= 20; i++) {
         try {
            temperatura = (int) temp.mostrarTemperatura();
            System.out.println("La temperatura mostrada es es: " + temperatura); //Imprimiremos 20 lecturas
de temperatura
            sleep(400);
         } catch (InterruptedException e) {
         }
      }
   }
}
```

Código Clase Temperatura:

```
import java.util.ArrayList;
import java.util.concurrent.locks.Lock;
import java.util.concurrent.locks.ReentrantLock;

public class Temperatura {

    private ArrayList<Integer> arl; //Array de temperaturas

    Lock cerrojo = new ReentrantLock();

    public Temperatura(int max) {
        this.arl = new ArrayList<>();

    }

    public void leerTemperatura(Integer temperatura) throws InterruptedException {
        cerrojo.lock();
        try {
            //Vamos añadiendo las temperaturas aleatorias de la clase productor
            arl.add(temperatura);
            //System.out.println(arl.get(0));
        } finally {
            cerrojo.unlock();
        }
    }

    public Integer mostrarTemperatura() throws InterruptedException {
        cerrojo.lock();
        try {
            int temperatura = arl.get(0);
            arl.remove(0);
            return (temperatura);
            //return (999);
        } finally {
            cerrojo.unlock();
        }
    }
}
```

Resultado:

```
run:
La temperatura mostrada es es: 13
La temperatura mostrada es es: 18
La temperatura mostrada es es: 54
La temperatura mostrada es es: 38
La temperatura mostrada es es: 33
```

```
La temperatura mostrada es es: 84
La temperatura mostrada es es: 2
La temperatura mostrada es es: 47
La temperatura mostrada es es: 64
La temperatura mostrada es es: 19
La temperatura mostrada es es: 51
La temperatura mostrada es es: 43
La temperatura mostrada es es: 91
La temperatura mostrada es es: 95
La temperatura mostrada es es: 53
La temperatura mostrada es es: 71
La temperatura mostrada es es: 41
La temperatura mostrada es es: 15
La temperatura mostrada es es: 12
La temperatura mostrada es es: 93
La temperatura mostrada es es: 46
BUILD SUCCESSFUL (total time: 8 seconds)
```