

Actividad 2.7.2 PSP.

En una hipotética cuenta bancaria. Crea una clase de nombre Saldo, con un atributo que nos indica el saldo, el constructor que da un valor inicial al saldo. Crea varios métodos uno para obtener el saldo y otro para dar valor al saldo, este método debe informar de quien añade cantidad al saldo, la cantidad que añade, el estado inicial del saldo (antes de añadir la cantidad) y el estado final del saldo después de añadir la cantidad. Define los parámetros necesarios que debe de recibir este método y defínelo como synchronized.

Crea una clase que extienda Thread, desde el método run() hemos de usar el método de la clase Saldo que añade la cantidad al saldo. Averigua los parámetros que se necesita en el constructor. No debe visualizar nada en pantalla.

Crea en el método main() un objeto Saldo asignándole un valor inicial. Visualiza el saldo inicial. Crea varios hilos que compartan ese objeto Saldo. A cada hilo le debes dar un nombre y le asignamos una cantidad. Lanzamos los hilos y esperamos a que finalicen para visualizar el saldo final del objeto Saldo. Comprueba los resultados quitando synchronized del método de la clase.

Código Clase Saldo:

```
public class Saldo {  
    private int saldo;  
  
    //Constructor de la clase Saldo, inicializo a 0 la clase Saldo  
    public Saldo(){  
        this.saldo=0;  
    }  
  
    //Retorno el valor del atributo saldo  
    public synchronized int retornarSaldo(){  
        return this.saldo;  
    }  
  
    //Aqui hago el depósito e incremento el saldo y le pongo el nombre para saber quien hizo ese depósito  
    public synchronized void darValorSaldo(String nombre,int mas){
```

```
        saldo=saldo+mas;

        System.out.println(nombre+" ha añadido a la cuenta "+mas+" y el valor del Saldo después
del depósito es= "+saldo);
    }
}
```

Código clase Hilo:

```
public class Hilo extends Thread{

    private Saldo s;

    private String nombre;

    public Hilo(String nombre,Saldo s){

        this.nombre=nombre;

        this.s=s;

    }

    public void run() {

        //Ponemos en el synchronized el objeto

        System.out.println("Saldo inicial= "+s.retornarSaldo());;

        s.darValorSaldo(this.nombre, 200);

        System.out.println("Saldo después del depósito= "+s.retornarSaldo());;

    }

}
```

Código clase Hilo2:

```
public class Hilo2 extends Thread{

    private Saldo s;

    private String nombre;


    public Hilo2(String nombre,Saldo s){

        this.nombre=nombre;

        this.s=s;

    }

    public void run() {
```

//Le pongo un sleep al segundo hilo así se ve mejor a la hora de visualizar lo que pasa por consola

```
try {
    Thread.sleep(1000);
} catch (InterruptedException ex) {
    Logger.getLogger(Hilo2.class.getName()).log(Level.SEVERE, null, ex);
}

//Retorno el saldo
System.out.println("Saldo inicial= "+s.retornarSaldo());

//Le añado por parámetro el valor que hago de depósito
s.darValorSaldo(this.nombre, 100);

//Visualizo el nuevo resultado del saldo
System.out.println("Saldo después del depósito= "+s.retornarSaldo());
}
}
```

Código clase Main:

```
public class Main {
    public static void main(String[] args) throws InterruptedException {
        Saldo s=new Saldo();
        Hilo hilo1=new Hilo("Lucia",s);
        Hilo2 hilo2=new Hilo2("Miriam",s);
        hilo1.start();

        //Le pongo un sleep al segundo hilo así se ve mejor a la hora de visualizar lo que pasa por consola
        Thread.sleep(1000);
        hilo2.start();
    }
}
```

Si ejecutamos, con el synchronized en los métodos sale lo siguiente:

```
Output - Actividad2.7.2 (run)

run:
Saldo inicial= 0
Saldo inicial= 0
Lucia ha añadido a la cuenta 200 y el valor del Saldo después del depósito es= 200
Miriam ha añadido a la cuenta 100 y el valor del Saldo después del depósito es= 300
Saldo después del depósito= 300
Saldo después del depósito= 300
BUILD SUCCESSFUL (total time: 0 seconds)
```

Si ejecutamos, sin el synchronized en los métodos sale lo siguiente:

```
Output - Actividad2.7.2 (run)

run:
Saldo inicial= 0
Lucia ha añadido a la cuenta 200 y el valor del Saldo después del depósito es= 200
Saldo inicial= 0
Miriam ha añadido a la cuenta 100 y el valor del Saldo después del depósito es= 300
Saldo después del depósito= 300
Saldo después del depósito= 200
BUILD SUCCESSFUL (total time: 0 seconds)
```

Como no se ve muy bien he puesto un sleep para que se pueda ver mejor:

```
Output - Actividad2.7.2 (run)

run:
Saldo inicial= 0
Lucia ha añadido a la cuenta 200 y el valor del Saldo después del depósito es= 200
Saldo después del depósito= 200
Saldo inicial= 200
Miriam ha añadido a la cuenta 100 y el valor del Saldo después del depósito es= 300
Saldo después del depósito= 300
BUILD SUCCESSFUL (total time: 1 second)
```