

TEMA 2 - HILOS EN JAVA

MÉTODOS BÁSICOS

EJERCICIO

Dado el siguiente código, crea un paquete, cópialo y analiza lo que hace.



Clase Cliente.java:

```
Dependienta.java  Cliente.java x  Main.java
7      public Cliente(String nombre, int[] compra) {
8          this.nombre = nombre;
9          this.compra = compra;
10     }
11
12     2 usages new *
13     public int[] getCompra() {
14         return compra;
15     }
16
17     no usages new *
18     public void setCompra(int[] compra) {
19         this.compra = compra;
20     }
21 }
```

Clase Dependienta:

```

1 package Tienda;
2
3 4 usages new *
4 public class Dependienta {
5     2 usages
6     String nombre;
7
8     2 usages new *
9     Dependienta(String nombre){
10         this.nombre=nombre;
11     }
12
13     2 usages new *
14 @ public void registrarCompra(Cliente cliente, long tiempo){
15     System.out.println("DEPENDIENTA: "+nombre+" atendiendo a:" + cliente.nombre);
16
17     for(int i=0; i<cliente.getCompra().length; i++) {
18         try{
19             //
20             Thread.sleep(cliente.getCompra()[i]);
21
22             //
23             System.out.println("PRODUCTO"+ (i+1) + " " +
24                 "pasa por caja en "+(System.currentTimeMillis()-tiempo)+"ms");
25         }catch (InterruptedException ex){
26             //
27             Thread.currentThread().interrupt();
28         }
29     }
30     System.out.println("\nCOMPRA DEL CLIENTE PROCESADA EN: "+(System.currentTimeMillis()-tiempo)+"ms");
31 }
32 }

```

Main.java

```
Dependiente.java  Cliente.java  Main.java x
1 package Tienda;
2
3 no usages new *
4 public class Main {
5     no usages new *
6     public static void main(String[] args) {
7
8         Cliente lola=new Cliente( nombre: "Lola",new int[] {200,300,400,200});
9         Cliente pepe=new Cliente( nombre: "Pepe",new int[] {200,200,500,200,300});
10
11         Dependienta de1=new Dependienta( nombre: "MARTA");
12         Dependienta de2=new Dependienta( nombre: "LORENA");
13
14         long tiempoInicio=System.currentTimeMillis();
15
16         de1.registrarCompra(lola,tiempoInicio);
17         de2.registrarCompra(pepe,tiempoInicio);
18     }
19 }
```

Cuestiones:

- ¿Para qué sirve el método interrupt()?

El método interrupt para la ejecución del hilo actual en caso de que se intentase para la ejecución del método sleep del try , ya que se supone que durante la ejecución del mismo el proceso está bloqueado y saltaría la excepción asociada.

- ¿Cómo podrías mejorar el código para ser más eficiente en la gestión del proceso de compra?

Se comprueba que se está procesando la compra del primer cliente por la primera dependienta y, secuencialmente, la del segundo cliente por la otra dependienta. Se ha detectado un error a la hora de mostrar el tiempo total de procesado de compra de cada cliente, ya que solo se considera el del último producto de su compra.

Dicho error se ha solventado con el método public long calculaTiempoCompra(ArrayList <Long> t) , que guarda los tiempos de compra de cada producto para cada cliente.

Lo que se ha implementado es que sencillamente se generen dos hilos de la clase Dependienta que procesen las compras de cada cliente de forma simultánea.

La clase Dependienta implementaría la clase Runnable, con su método run (), y se generarían hilos de cada objeto Dependienta generado y se lanzarían los mismos con start().

Se adjunta una captura de los tiempos de los procesados de compra del código original y la nueva versión del mismo y se puede comprobar que , mientras el tiempo de procesamiento del primer cliente es aproximadamente igual , el del segundo es prácticamente la mitad.

```
"C:\Program Files\Java\jdk-20\bin\java.exe" "-javaagent:C:\Program F
Depednienta MARTA atendiendo a lola
PRODUCTO 1 de cliente lola pasa por caja en 229 ms
PRODUCTO 2 de cliente lola pasa por caja en 643 ms
PRODUCTO 3 de cliente lola pasa por caja en 945 ms
PRODUCTO 4 de cliente lola pasa por caja en 1152 ms
COMPRA DEL CLIENTE lola PROCESADA EN 2969 ms
Depednienta LORENA atendiendo a pepe
PRODUCTO 1 de cliente pepe pasa por caja en 1374 ms
PRODUCTO 2 de cliente pepe pasa por caja en 1578 ms
PRODUCTO 3 de cliente pepe pasa por caja en 2079 ms
PRODUCTO 4 de cliente pepe pasa por caja en 2281 ms
PRODUCTO 5 de cliente pepe pasa por caja en 2595 ms
COMPRA DEL CLIENTE pepe PROCESADA EN 9907 ms
```

```
"C:\Program Files\Java\jdk-20\bin\java.exe" "-javaagent:C:\
Depednienta MARTA atendiendo a lola
Depednienta LORENA atendiendo a pepe
PRODUCTO 1 de cliente lola pasa por caja en 222 ms
PRODUCTO 1 de cliente pepe pasa por caja en 222 ms
PRODUCTO 2 de cliente pepe pasa por caja en 443 ms
PRODUCTO 2 de cliente lola pasa por caja en 648 ms
PRODUCTO 3 de cliente lola pasa por caja en 951 ms
PRODUCTO 3 de cliente pepe pasa por caja en 951 ms
PRODUCTO 4 de cliente pepe pasa por caja en 1156 ms
PRODUCTO 4 de cliente lola pasa por caja en 1156 ms
COMPRA DEL CLIENTE lola PROCESADA EN 2977 ms
PRODUCTO 5 de cliente pepe pasa por caja en 1469 ms
COMPRA DEL CLIENTE pepe PROCESADA EN 4241 ms
```

- Implementa la respuesta en java

Se adjunta el código de la clase Main y Dependienta que son las únicas que cambian.

```
public class Main {
    public static void main(String[] args) {

        Cliente lola=new Cliente("lola",new int[]
{200,400,300,200});
        Cliente pepe =new Cliente("pepe",new
int[]{200,200,500,200,300});

        long tiempoInicio=System.currentTimeMillis();

        //Se generan los objetos de tipo Runnable
        Runnable r1=new Dependienta("MARTA",lola,tiempoInicio);
        Runnable r2=new Dependienta("LORENA",pepe,tiempoInicio);

        //Se asignan a los objetos de tipo Thread
        Thread t1=new Thread(r1);
        Thread t2=new Thread(r2);

        //Arrancan los hilos
        t1.start();
        t2.start();

    }
}
```

```

public class Dependienta implements Runnable{

    String nombre;
    Cliente cliente;
    long tiempoInicio;
    ArrayList <Long> tiemposCompraProducto =new
ArrayList<Long> ();

    public Dependienta(String nombre, Cliente cliente, long
tiempoInicio){

        this.nombre=nombre;
        this.cliente=cliente;
        this.tiempoInicio=tiempoInicio;
    }

    public void run() {

        System.out.println("Depednienta "+nombre+"
atendiendo a "+cliente.nombre);

        for(int i=0;i<cliente.getCompra().length;i++){

            try {
                Thread.sleep(cliente.getCompra()[i]);

                //se guarda el tiempo de compra de cada
producto

                tiemposCompraProducto.add((System.currentTimeMillis() -
tiempoInicio));

                System.out.println("PRODUCTO " + (i + 1) + "
de cliente "+cliente.nombre+" pasa por caja en " +
tiemposCompraProducto.get(i) + " ms ");

                } catch (InterruptedException ex) {
                    Thread.currentThread().interrupt();
                }

            }

            System.out.println("COMPRA DEL CLIENTE
"+cliente.nombre+" PROCESADA EN
"+calculaTiempoCompra(tiemposCompraProducto)+" ms ");
        }

        public long calculaTiempoCompra(ArrayList <Long> t){
            long res=0;
            for (int i=0; i<t.size();i++){
                res=res+t.get(i);
            }
            return (res);
        }
    }

```

```
}  
  
}
```