

ABNER GALINDO GONZALEZ 0242596compraventa

```
public class PruebaBuferCompartido {
    public static void main(String [] args) {
        //crear el objeto compartido utilizando por los subprocessos
        Bufer ubicacionCompartida=new BuferNoSincronizado();

        //crear objetos productor y consumidor
        Productor productor=new Productor( ubicacionCompartida);
        Consumidor consumidor =new Consumidor( ubicacionCompartida);

        productor.start();//iniciar subprocesso productor
        consumidor.start();//iniciar subprocesso consumidor
    }
}

public class BuferNoSincronizado implements Bufer
{
    private int bufer=-1;//compartido por los subprocesso productor y cosntatres

    //colocar valor en bufer
    public void establecer(int valor)
    {
        System.err.println(Thread.currentThread().getName()+" Escribe:"+valor);
        bufer=valor;
    }
    //devolver valor de bufer
    public int obtener()
    {
        System.err.println(Thread.currentThread().getName()+" Lee:" + bufer);

        return bufer;
    }
} //fin de lace bufernosintrocnicado

public interface Bufer{
    public void establecer(int valor);//coloacr valro en bufer
    public int obtener();
}

public class Consumidor extends Thread {
    private Bufer ubicacionCompartida;//referencia al objeto compartir

    //constructor
    public Consumidor(Bufer compartido)
    {
        super("Consumidor");
    }
}
```

```

ubicacionCompartida=compartido;
}

//leer el valor de ubicacionCompartida cuatro veces y sumar los valores
public void run()
{
    int suma=0;
    for(int cuenta=1;cuenta<=4;cuenta++)
    {
        //estar inactivo de 0 s3 , leer un valor de Bufer y agregarlo
        try {
            Thread.sleep((int)(Math.random()*3001));
            suma += ubicacionCompartida.obtener();
        }

        //si se interrumpe el subproceso inactivo, imprimir rastreo de la pila
        catch (InterruptedException excepcion){
            excepcion.printStackTrace();
        }
    }

    System.err.println(getName()+" Leyo valores, dado un total de:"+ suma
+".\nTerminado"+ getName()+".");
} //fin del metodo run
} //fin de la clase Consumidor

```

```

public class Productor extends Thread
{
    private Bufer ubicacionCompartida;//referencia al objeto compartido
    //constructor

    /** Creates a new instance of Producto */
    public Productor(Bufer compartido)
    {
        super("Productor");
        ubicacionCompartida=compartido;
    }

    //almacenar valores de 1 a 4 en ubicacionCompartida
    public void run()
    {
        for(int cuenta=1;cuenta<=4;cuenta++){
            //estar inactivo de 0 a 3 segundo y luego colocar valor en Bufer
            try{
                Thread.sleep((int)(Math.random()*3001));
                ubicacionCompartida.establecer(cuenta);
            }
            //si se interrumpe el subproceso inactivo, imprimir rastreo de pila
            catch(InterruptedException excepcion){
                excepcion.printStackTrace();
            }
        }
    }
}

```

```
    }//fin de intriccion for
    System.err.println(getName()+"Termino de
producir."+"\nTerminado"+getName()+".");
} //fin de metodo run
} //fin de clase productor
```

La ejecución del programa es:

```
Consumidor Lee:-1
Productor Escribe:1
Consumidor Lee:1
Consumidor Lee:1
Consumidor Lee:1
Consumidor Leyo valores, dado un total de:2.
TerminadoConsumidor.
Productor Escribe:2
Productor Escribe:3
Productor Escribe:4
ProductorTermino de producir.
TerminadoProductor.
```

```
Productor Escribe:1
Productor Escribe:2
Consumidor Lee:2
Productor Escribe:3
Consumidor Lee:3
Productor Escribe:4
ProductorTermino de producir.
TerminadoProductor.
Consumidor Lee:4
Consumidor Lee:4
Consumidor Leyo valores, dado un total de:13.
TerminadoConsumidor.
```

En el proceso se alternan la secuencia de la lectura del bufer entre el productor y consumidor con posibilidad de obtener cuatro veces y valores entre 0 y 4.

El proceso se establece para el que obtenga su primer resultado. Con la característica de mandar a dormir proceso del productor y consumidor con el método sleep(). Activándolo con Start().