

## Ejercicio1:

```

class Interbloqueos{
    2 usages
    private Object herramienta1 = new Object();
    2 usages
    private Object herramienta2 = new Object();

    //metodos
    1 usage
    public void quieroLaLLave(){
        try{
            synchronized (herramienta1){
                Thread.sleep( millis: 2000); //le dejo un tiempo
                synchronized (herramienta2){
                    Thread.sleep( millis: 2000); //le dejo un tiempo
                }
            }
        } catch (InterruptedException e){
            e.printStackTrace();
        }
    }

    1 usage
    public void yoQuieroLaLLave(){
        try{
            synchronized (herramienta2){
                Thread.sleep( millis: 2000); //le dejo un tiempo
                synchronized (herramienta1){
                    Thread.sleep( millis: 2000); //le dejo un tiempo
                }
            }
        } catch (InterruptedException e){
            e.printStackTrace();
        }
    }
} //fin class Interbloqueo

```

```

class Hilo implements Runnable{
    3 usages
    private Interbloqueos mecaBloq;
    3 usages
    private int herramienta;

    2 usages
    public Hilo(Interbloqueos mecaBloq, int herramienta){
        this.mecaBloq = mecaBloq;
        this.herramienta = herramienta;
    }

    @Override
    public void run() {
        while (herramienta != 0)
            switch (herramienta) {
                case 1:
                    System.out.println("Herramienta: llave inglesa");
                    mecaBloq.quieroLaLLave();
                    break;
                case 2:
                    System.out.println("Herramienta: carraca");
                    mecaBloq.yoQuieroLaLLave();
                    break;
            }

        /*
        //herramienta 1 = inglesa
        //herramienta 2 = carraca

        if (this.herramienta == 1)
            mecaBloq.quieroLaLLave();
        else
            mecaBloq.yoQuieroLaLLave();
        */
    }
}

} //fin class Hilo

```

```
public class Mecanicos{  
  
    public static void main (String argv[]){  
  
        Interbloqueos mecanico = new Interbloqueos();  
  
        Thread h1 = new Thread(new Hilo(mecanico, herramienta: 1));  
        Thread h2 = new Thread(new Hilo(mecanico, herramienta: 2));  
  
        h1.start();  
        h2.start();  
  
        try{  
  
            //hago join para que no me se terminen los dos tando h1 como h2  
            h1.join();  
            h2.join();  
  
        }catch(InterruptedException e){  
            e.printStackTrace();  
        }  
    }  
  
} //fin class mecanicos
```

ejecucion:



```
Mecanicos x Principal x  
C:\Users\tarde\.jdk\openjdk-19\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Communit  
Herramienta: llave inglesa  
Herramienta: carraca
```

## Ejercicio2:

```
class Personas {  
    2 usages  
    private long per1 = 0;  
    private long per2 = 0;  
    1 usage  
    private long per3 = 0;  
    private long per4 = 0;  
    1 usage  
    private long per5 = 0;  
  
    2 usages  
    private final Object bloqueoCuba1 = new Object();  
    private final Object bloqueoCuba2 = new Object();  
  
    public void cuba1(){  
        synchronized (bloqueoCuba1){  
            per1++;  
            per3++;  
            per5++;  
        }  
    }  
  
    public long getCont1(){  
        synchronized (bloqueoCuba1){  
            return per1;  
        }  
    }  
}
```