13/02/24

// Deadlock

```
classA{
    synchronized void foo(B,b){
    String name= thread.currentThread().getName();
    System.out.println(name +"entered A.foo");
    try{
         Thread.sleep(1000);
    } catch(Exception e){
          System.out.println("A Interrupted");
     }
     System.out.println(name +" trying to call B.last()");
     b.last();
     }
     void last(){
            System.out.println("Inside A.last");
     }
}
Class B{
      Synchronized void bar(A a){
      String name= Thread.current Thread().getName();
      System.out.println(name + "entered B.bar");
      try{
           Thread.sleep(1000);
      } catch(Exception e){
            System.out.println("B Interrupted");
       }
      System.out.println(name +"trying to call A.last()");
      a.last();
      }
      void last(){
            System.out.println("Inside A.last");
      }
}
```

```
class Deadlock implements Runnable.
{
    A a = new A();
    B b = new B();
    Deadlock(){
        Thread.currentThread().setName("MainThread");
        Thread t = new Thread(this, "Racing thread");
        t.start();
        a.foo(b);
        System.out.println("Back in main thread");
    }
    public void run(){
        b.bar(a);
        System.out.println("Back in other thread");
    }
    public static void main(String args[]){
        new Deadlock();
    }
}
```

output:

```
Main thread entered A.foo
Racing Thread entered B.bar
Main Thread trying to call B.last()
Inside A.last
Back in Main thread
Racing Thread trying to call A.last()
Inside A.last
Back in other thread
```

[illegible] 13/2/24