

15:55



≡ SimulacionHospital.java



```
1 import java.util.concurrent.ExecutorService;
2 import java.util.concurrent.Executors;
3 import java.util.concurrent.locks.ReentrantLock;
4
5 public class SimulacionHospital {
6
7     public static void main(String[] args) {
8         System.out.println("Iniciando acceso a la Sala de Cirugía...");
9
10        RecursoMedico salaCirugia = new RecursoMedico("Sala de Cirugia");
11
12        // Crear un pool de 4 hilos
13        ExecutorService executor = Executors.newFixedThreadPool(4);
14
15        // Crear tareas que representan a profesionales médicos
16        Runnable medico1 = new ProfesionalMedico("Dra. Sanchez", salaCirugia);
17        Runnable medico2 = new ProfesionalMedico("Dr. Gomez", salaCirugia);
18        Runnable medico3 = new ProfesionalMedico("Enfermera Lopez", salaCirugia);
19        Runnable medico4 = new ProfesionalMedico("Dr. Ramirez", salaCirugia);
20
21        // Ejecutar las tareas
22        executor.execute(medico1);
23        executor.execute(medico2);
24        executor.execute(medico3);
25        executor.execute(medico4);
26
27        // Cerrar el executor
```

```

27         // Cerrar el executor
28         executor.shutdown();
29     }
30 }
31
32 class RecursoMedico {
33     private String nombre;
34     private ReentrantLock lock = new ReentrantLock();
35
36     public RecursoMedico(String nombre) {
37         this.nombre = nombre;
38     }
39
40     public void usar(String profesional) {
41         lock.lock();
42         try {
43             System.out.println(profesional + " ha ingresado a " + nombre);
44             Thread.sleep(1500); // Simular uso del recurso
45             System.out.println(profesional + " ha salido de " + nombre);
46         } catch (InterruptedException e) {
47             e.printStackTrace();
48         } finally {
49             lock.unlock();
50         }
51     }
52 }
53

```

15:55



≡ SimulacionHospital.java



```
42     try {
43         System.out.println(profesional + " ha ingresado a " + nombre);
44         Thread.sleep(1500); // Simular uso del recurso
45         System.out.println(profesional + " ha salido de " + nombre);
46     } catch (InterruptedException e) {
47         e.printStackTrace();
48     } finally {
49         lock.unlock();
50     }
51 }
52 }
53
54 class ProfesionalMedico implements Runnable {
55     private String nombre;
56     private RecursoMedico recurso;
57
58     public ProfesionalMedico(String nombre, RecursoMedico recurso) {
59         this.nombre = nombre;
60         this.recurso = recurso;
61     }
62
63     public void run() {
64         recurso.usar(nombre);
65     }
66 }
67
```

```
Iniciando acceso a la Sala de Cirugía...
Dra. Sanchez ha ingresado a Sala de Cirugía
Dra. Sanchez ha salido de Sala de Cirugía
Dr. Gomez ha ingresado a Sala de Cirugía
Dr. Gomez ha salido de Sala de Cirugía
Enfermera Lopez ha ingresado a Sala de Cirugía
Enfermera Lopez ha salido de Sala de Cirugía
Dr. Ramirez ha ingresado a Sala de Cirugía
Dr. Ramirez ha salido de Sala de Cirugía
█
```

