

```

1 package Modelo;
2
3 import Interfaz.Escritor;
4 import static java.lang.Thread.sleep;
5 import java.util.ArrayList;
6 import java.util.Queue;
7 import java.util.concurrent.ConcurrentLinkedQueue;
8 import java.util.concurrent.locks.Condition;
9 import java.util.concurrent.locks.Lock;
10 import java.util.concurrent.locks.ReentrantLock;
11 import java.util.logging.Level;
12 import java.util.logging.Logger;
13
14 /**
15  *
16  * @author marco
17  */
18 public class Tirolina {
19     private int estadoTirolina = 0;
20     private int vecesUsado = 0;
21     private ArrayList<Ninno> ninnoTirolina = new ArrayList<>();
22     private ArrayList<Monitor> monTirolina = new ArrayList<>();
23     private Queue<Ninno> colaTirolina = new ConcurrentLinkedQueue();
24     private Escritor escritor;
25     private Paso paso;
26     private Lock lockTirolina = new ReentrantLock();
27     private Condition monitorTirolina = lockTirolina.newCondition();
28     private Condition primeroCola = lockTirolina.newCondition();
29     private Condition actividadesMonitor = lockTirolina.newCondition();
30
31     public Tirolina(Escritor p_escritor, Paso p_paso){
32         escritor = p_escritor;
33         paso = p_paso;
34     }
35
36     public void tirolina(Ninno ninno){
37         paso.mirar();
38         colaTirolina.offer(ninno);
39         escritor.addMsg(ninno.getMiId() + " se pone a la cola de la TIROLINA");
40         paso.mirar();
41         lockTirolina.lock();
42         try {
43             while (monTirolina.size()==0 || !ninno.equals(colaTirolina.peek())){
44                 if (monTirolina.size()==0) monitorTirolina.await();
45                 if (!colaTirolina.peek().equals(ninno))primeroCola.await();
46             }
47             colaTirolina.poll();
48             ninnoTirolina.add(ninno);
49             paso.mirar();
50             estadoTirolina++;
51             escritor.addMsg(ninno.getMiId() + " se empieza a preparar en la TIROLINA");
52             sleep(1000);
53             estadoTirolina++;
54             paso.mirar();
55             sleep(3000);
56             escritor.addMsg(ninno.getMiId() + " se monta en la TIROLINA");
57             paso.mirar();
58             estadoTirolina++;
59             escritor.addMsg(ninno.getMiId() + " llega al fin de la TIROLINA");
60             sleep(500);
61             paso.mirar();
62             estadoTirolina=0;
63             ninnoTirolina.remove(ninno);
64             vecesUsado++;
65             ninno.subtractActividad(1);
66             monTirolina.get(0).subtractActividad();
67             actividadesMonitor.signal();
68             primeroCola.signalAll();
69         } catch (InterruptedException ex) {

```

```

70         Logger.getLogger(Campamento.class.getName()).log(Level.SEVERE, null, ex);
71     } finally {
72         lockTirolina.unlock();
73     }
74     escritor.addMsg(ninno.getMiId() + " abandona la TIROLINA");
75 }
76
77 public void tirolina(Monitor mon) {
78     paso.mirar();
79     lockTirolina.lock();
80     try {
81         monTirolina.add(mon);
82         paso.mirar();
83         monitorTirolina.signalAll();
84         escritor.addMsg(mon.getMiId() + " llega a la TIROLINA");
85         while (mon.getContadorActividades()>0){
86             actividadesMonitor.await();
87         }
88         paso.mirar();
89         monTirolina.remove(mon);
90         escritor.addMsg(mon.getMiId() + " abandona la TIROLINA");
91     } catch (InterruptedException ex) {
92         Logger.getLogger(Tirolina.class.getName()).log(Level.SEVERE, null, ex);
93     } finally {
94         lockTirolina.unlock();
95     }
96 }
97
98 public String getCola(){
99     String msg = "";
100     for (Ninno ninno:colaTirolina){
101         msg += ninno.getMiId() + " ";
102     }
103     return msg;
104 }
105 public int getTamCola(){
106     return colaTirolina.size();
107 }
108
109 public String getNinno(){
110     String msg = "";
111     for (Ninno ninno:ninnoTirolina){
112         msg += ninno.getMiId() + " ";
113     }
114     return msg;
115 }
116 public int cuantosNinnosCola(){
117     return colaTirolina.size();
118 }
119
120 public String getMon(){
121     String msg = "";
122     for (Monitor mon:monTirolina){
123         msg += mon.getMiId() + " ";
124     }
125     return msg;
126 }
127
128 public int getEstadoTirolina(){
129     return estadoTirolina;
130 }
131
132 public int getVecesUsado() {
133     return vecesUsado;
134 }
135
136
137
138 }
139

```

