

```

1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6  package Modelo;
7
8  import Interfaz.Escritor;
9  import static java.lang.Thread.sleep;
10 import java.util.Queue;
11 import java.util.concurrent.ConcurrentLinkedQueue;
12 import java.util.concurrent.CopyOnWriteArrayList;
13 import java.util.concurrent.Semaphore;
14 import java.util.logging.Level;
15 import java.util.logging.Logger;
16
17 /**
18  *
19  * @author marco
20  */
21 public class Merendero {
22     private ContadorBandejas bandLimpias, bandSucias;
23     private Escritor escritor;
24     private Paso paso;
25     private Queue<Ninno> colaMerendero = new ConcurrentLinkedQueue();
26     private CopyOnWriteArrayList<Ninno> ninnoMerendero = new CopyOnWriteArrayList<>();
27     private CopyOnWriteArrayList<Monitor> monMerendero = new CopyOnWriteArrayList<>();
28     private Semaphore semMerienda;
29
30     public Merendero(int p_nBandejas, int p_aforoMer, Escritor p_escritor, Paso p_paso){
31         bandLimpias = new ContadorBandejas(0);
32         bandSucias = new ContadorBandejas(p_nBandejas);
33         semMerienda = new Semaphore(p_aforoMer, true);
34         escritor = p_escritor;
35         paso = p_paso;
36     }
37
38     public void merendar (Ninno ninno){
39         paso.mirar();
40         colaMerendero.offer(ninno);
41         escritor.addMsg(ninno.getMiId() + " se pone a la cola de MERENDERO");
42         paso.mirar();
43         try {
44             semMerienda.acquire();
45             colaMerendero.remove(ninno);
46             ninnoMerendero.add(ninno);
47             paso.mirar();
48             bandLimpias.extraerBandeja();
49             paso.mirar();
50             escritor.addMsg(ninno.getMiId() + " se sienta a comer en el MERENDERO");
51             sleep(7000);
52             paso.mirar();
53             bandSucias.annadirBandeja();
54             paso.mirar();
55             ninnoMerendero.remove(ninno);
56             escritor.addMsg(ninno.getMiId() + " abandona el MERENDERO");
57         } catch (InterruptedException ex) {
58             Logger.getLogger(Merendero.class.getName()).log(Level.SEVERE, null, ex);
59         } finally {
60             semMerienda.release();
61         }
62     }
63
64     public void merendero (Monitor mon){
65         paso.mirar();
66         monMerendero.add(mon);
67         escritor.addMsg(mon.getMiId() + " llega al MERENDERO");
68         paso.mirar();
69         while(mon.getContadorActividades() > 0){

```

```

70         bandSucias.extraerBandeja();
71         paso.mirar();
72         try {
73             sleep((int) (3000 + 2000*Math.random()));
74         } catch (InterruptedException ex) {
75             Logger.getLogger(Merendero.class.getName()).log(Level.SEVERE, null, ex);
76         }
77         paso.mirar();
78         bandLimpias.annadirBandeja();
79         mon.subtractActividad();
80         escritor.addMsg(mon.getMiId() + " limpia una bandeja");
81     }
82     paso.mirar();
83     monMerendero.remove(mon);
84     escritor.addMsg(mon.getMiId() + " abandona el MERENDERO");
85 }
86
87 public int getBandLimpias() {
88     return bandLimpias.get();
89 }
90
91 public int getBandSucias() {
92     return bandSucias.get();
93 }
94
95 public String getCola() {
96     String msg = "";
97     for (Ninno ninno:colaMerendero){
98         msg += ninno.getMiId() + " ";
99     }
100    return msg;
101 }
102
103 public String getNinno() {
104     String msg = "";
105     for (Ninno ninno:ninnoMerendero){
106         msg += ninno.getMiId() + " ";
107     }
108     return msg;
109 }
110 public int cuantosNinnosMerienda(){
111     return ninnoMerendero.size();
112 }
113
114 public String getMon() {
115     String msg = "";
116     for (Monitor mon:monMerendero){
117         msg += mon.getMiId() + " ";
118     }
119     return msg;
120 }
121 }
122

```