

```

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  /**
9   *
10  * @author marco
11  */
12  public class Ninno extends Thread implements Comparable<Ninno>{
13      //ATRIBUTOS (privados)
14      private String id;
15      private int contActividades, totalActividades;
16      private Campamento campamento;
17
18      public Ninno(int p_id, int p_contActividades, Campamento p_campamento){
19          id = ""+100000+p_id;
20          id = "N"+id.substring(1);
21          contActividades = p_contActividades;
22          totalActividades = p_contActividades;
23          campamento = p_campamento;
24      }
25
26      public String getMiId() {
27          return id;
28      }
29
30      public void entrarCamp(){
31          boolean entrada = Math.random()<0.5;
32          if (entrada){
33              campamento.entrarPuerta1(this);
34          }
35          else{
36              campamento.entrarPuerta2(this);
37          }
38      }
39
40      public void seleccionarActividad(){
41          boolean meriendaDisp = contActividades<=12;
42          int k=0;
43          if (meriendaDisp) k=1;
44          int actividad = (int) ((2 + k) * Math.random());
45          switch (actividad){
46              case 0 -> campamento.usarTirolina(this);
47              case 1 -> campamento.usarSoga(this);
48              case 2 -> campamento.usarMerendero(this);
49          }
50          campamento.usarZonaComun(this);
51      }
52
53      public void salirCamp(){
54          campamento.salirCampamento(this);
55          campamento.calificar(this);
56      }
57
58      public void subtractActividad(int num){
59          contActividades-=num;
60      }
61
62      public int actividadesRealizadas(){
63          return totalActividades - contActividades;
64      }
65
66      public boolean equals(Ninno ninno) {
67          return Integer.valueOf(id.substring(1)).equals(Integer.valueOf(ninno.getMiId().substring(1)));
68      }
69      public boolean equals(String idNinno) {
70          return Integer.valueOf(id.substring(1)).equals(Integer.valueOf(idNinno.substring(1)));
71      }
72
73      @Override
74      public int compareTo(Ninno ninno) {
75          //a negative integer, zero, or a positive integer as this object is less than, equal to, or greater than the specified object.
76          return Integer.valueOf(id.substring(1)).compareTo(Integer.valueOf(ninno.getMiId().substring(1)));
77      }
78      public int compareTo(String idNinno) {
79          //a negative integer, zero, or a positive integer as this object is less than, equal to, or greater than the specified object.
80          return Integer.valueOf(id.substring(1)).compareTo(Integer.valueOf(idNinno.substring(1)));
81      }
82
83      public void run(){
84          entrarCamp();
85          while (contActividades>0){
86              seleccionarActividad();
87          }
88          salirCamp();
89      }
90  }
91

```