```java
package Puntuacion;

import java.util.ArrayList;

public class Ordenar{
    public static ArrayList<Tiempo>tiemporestado;
    public static ArrayList<Ciclista>tiemposGeneralE1;
    public static ArrayList<Ciclista>Abandonos;

    public static void calcularT(ArrayList<Ciclista>lista,Tiempo salida) {// este metodo lo que hace es recibir una lista de dorsal y tiempo (ciclistas) y crea otro con los tiempos ya restados al tiempo de salida.
        tiemporestado=new ArrayList<Tiempo>();

        for(Ciclista i :lista) {
            i.getUnTiempo().restaTiempo(salida);
            tiemporestado.add(i.getUnTiempo());
        }
    }
    public static void calcularTContra(ArrayList<Ciclista>lista,ArrayList<Ciclista>tiempo){
        Puntos.ordenNumDorsal(lista);
        Puntos.ordenNumDorsal(tiempo);
        tiemporestado=new ArrayList<Tiempo>();

        for(int i=0;i<lista.size();i++) {
            lista.get(i).getUnTiempo().restaTiempo(tiempo.get(i).getUnTiempo());
            tiemporestado.add(lista.get(i).getUnTiempo());
        }
    }
    public static ArrayList<Ciclista> dorsalTiempoGeneral(ArrayList<Ciclista>uno){
        tiemposGeneralE1=new ArrayList<>();
        int j=0;
        for(Ciclista i:uno) {
            tiemposGeneralE1.add(new Ciclista(i.getDorsal(),tiemporestado.get(j)));
            j++;
        }
        return tiemposGeneralE1;
    }
    public static ArrayList<Ciclista> sumaTiempoArray(ArrayList<Ciclista>uno,ArrayList<Ciclista>dos){
        Puntos.ordenNumDorsal(uno);
        Puntos.ordenNumDorsal(dos);
        for(int i=0;i<dos.size();i++) {
            dos.get(i).getUnTiempo().sumaTiempo(uno.get(i).getUnTiempo());
        }
        return dos;
    }

    public static  String Puntuacion(ArrayList<Ciclista>lista,String categoria) {
        int num=1;
        String cadena="";
        String espacio=" ";
        cadena="|-----------------------------------------------------|\n";
        cadena=cadena+"| Puntuación:                                         |\n";
        for(@SuppressWarnings("unused") Ciclista i:lista) {
            if(num>=10) {
                espacio="";
            }else {espacio=" ";}
            if(num>3&&!categoria.equals("categoria1")) {break;}else if(num>4) {break;}
            cadena=cadena+"| "+espacio+num+". ("+lista.get(num-1).getDorsal()+") "+Cuadrar.añadirEspaciosNom(Buscar.obetenerAtributosPorDorsal(lista.get(num-1).getDorsal
```

```
   ()).getNombre())+"           "+tiemporestado.get(num-1) + "
   "+Puntos.numerosPuntuacion(categoria)[num-1]+" Puntos |\n";
 61            num++;
 62        }
 63        return cadena;
 64    }
 65    public static  String PuntuacionTotal(ArrayList<Ciclista>lista) {
 66        int num=1;
 67        int j=0;
 68        String espacio=" ";
 69        String cadena="";
 70        cadena="|-----------------------------------------------------|\n";
 71        for(Ciclista i : lista) {
 72            if(num>=10) {
 73                espacio="";
 74            }else {espacio=" ";}
 75            cadena=cadena+"| "+espacio+num+". ("+i.getDorsal()+")
   "+Cuadrar.añadirEspaciosNom(Buscar.obetenerAtributosPorDorsal(i.getDorsal()).getNombre
   ())+Buscar.obetenerAtributosPorDorsal(i.getDorsal()).getPais()+"
   "+tiemporestado.get(j) + " |\n";
 76            num++;
 77            j++;
 78        }
 79        return cadena;
 80    }
 81    public static  String PuntuacionGeneralTotal(ArrayList<Ciclista>lista) {
 82    int num=1;
 83    String espacio=" ";
 84    String cadena="";
 85    cadena="|-----------------------------------------------------|\n";
 86    for(Ciclista i : lista) {
 87        if(num>=10) {
 88            espacio="";
 89        }else {espacio=" ";}
 90        cadena=cadena+"| "+espacio+num+". ("+i.getDorsal()+")
   "+Cuadrar.añadirEspaciosNom(Buscar.obetenerAtributosPorDorsal(i.getDorsal()).getNombre
   ())+Buscar.obetenerAtributosPorDorsal(i.getDorsal()).getPais()+"           "+i.getUnTiempo
   () + " |\n";
 91        num++;
 92    }
 93    return cadena;
 94 }
 95    public static String ClasificacionM() {
 96        Puntos.losPrimerosM();
 97        int num=1;
 98        String espacio=" ";
 99        String espacio2=" ";
100        String cadena="";
101        cadena="|                                                    |\n";
102        cadena=cadena+"| Clasificación:                                    |\n";
103        for(Ciclista i :Puntos.listaCicPuntos) {
104            if(num>=10) {
105                espacio="";
106            }else {espacio=" ";}
107            if(i.getPuntosM()>=10) {
108                espacio2="";
109            }else {espacio2=" ";}
110            if(i.getPuntosM()!=0) {
111                cadena=cadena+"| "+espacio+num+". ("+i.getDorsal
   ()+")"+Cuadrar.añadirEspaciosNom(Buscar.obetenerAtributosPorDorsal(i.getDorsal()).getNombre
   ())+Cuadrar.añadirEspaciosEquipo(Buscar.obetenerAtributosPorDorsal(i.getDorsal()).getEquipo
   ())+" "+espacio2+i.getPuntosM()+" Puntos |\n";
```

```
112                 num++;
113             }
114         }
115         return cadena;
116     }
117     public static String ClasificacionP() {
118         Puntos.losPrimerosP();
119         int num=1;
120         String espacio=" ";
121         String espacio2=" ";
122         String cadena="";
123         cadena="|                                                    |\n";
124         cadena=cadena+"| Clasificación:                                    |\n";
125         for(Ciclista i :Puntos.listaCicPuntos) {
126             if(num>=10) {
127                 espacio="";
128             }else {espacio=" ";}
129             if(i.getPuntosP()>=10) {
130                 espacio2="";
131             }else {espacio2=" ";}
132             if(i.getPuntosP()!=0) {
133                 cadena=cadena+"| "+espacio+num+". ("+i.getDorsal
    ()+")"+Cuadrar.añadirEspaciosNom(Buscar.obetenerAtributosPorDorsal(i.getDorsal()).getNombre
    ())+Cuadrar.añadirEspaciosEquipo(Buscar.obetenerAtributosPorDorsal(i.getDorsal()).getEquipo
    ())+" "+espacio2+i.getPuntosP()+" Puntos |\n";
134                 num++;
135             }
136         }
137         return cadena;
138     }
139     public static String ClasificacionFinal1() {
140         String cadena="";
141         cadena="                                                    \n";
142         cadena=cadena+"------------------------------------------------------\n";
143         cadena=cadena+"|                   CLASIFICACIONES                  |\n";
144         cadena=cadena+"|----------------------------------------------------|\n";
145         cadena=cadena+"|GENERAL :                                           |";
146 
147         return cadena;
148 
149     }
150 
151     public static String ClasificacionFinal2() {
152         String cadena="";
153         cadena=cadena+"|----------------------------------------------------|\n";
154         cadena=cadena+"|METAS VOLANTES :                                    |\n";
155         cadena=cadena+ClasificacionM();
156         cadena=cadena+"|----------------------------------------------------|\n";
157         cadena=cadena+"|MONTAÑA :                                           |\n";
158         cadena=cadena+ClasificacionP();
159         cadena=cadena+"------------------------------------------------------\n";
160         return cadena;
161 
162     }
163     public static void listaAbandonos(ArrayList<Ciclista> salida, ArrayList<Ciclista> meta)
    { //buscamos el que si que esta y le decimos que no le guarde, asi sabemos cual tenemos que
    guardar.
164         Abandonos = new ArrayList<Ciclista>();
165         for (int i = 0; i < salida.size(); i++) {
166             boolean encontrado = false;
167             for (int j = 0; j < meta.size(); j++) {
168                 if (salida.get(i).getDorsal().equals(meta.get(j).getDorsal())) {
```

```
169                 encontrado = true;
170                 }
171             }
172             if (!encontrado) {
173                 Abandonos.add(salida.get(i));
174             }
175         }
176     }
177     public static void matarAbandonos() {
178         for (int i = 0; i < Puntos.listaCicPuntos.size(); i++) {
179             if (Puntos.listaCicPuntos.get(i).getDorsal().equals(Abandonos.get(0).getDorsal
    ())) {
180                 Puntos.listaCicPuntos.remove(Puntos.listaCicPuntos.get(i));
181             }
182         }
183     }
184     public static String mostrarAbandonos() {
185         String cadena="";
186         cadena=cadena+"| Abandonos:                                                    |\n";
187         for(Ciclista i : Abandonos) {
188             cadena=cadena+"| ("+Buscar.obetenerAtributosPorDorsal(i.getDorsal()).getDorsal
    ()+") "+Buscar.obetenerAtributosPorDorsal(i.getDorsal()).getNombre()+"\n";
189         }
190         return cadena;
191     }
192 
193     public static ArrayList<Ciclista> ordenarPorTiempo(ArrayList<Ciclista> uno) {
194         ArrayList<Ciclista>ordenados = new ArrayList<>();
195         for(Ciclista i: uno) {
196             for(int j =0; j<ordenados.size();j++) {
197                 if (!i.getUnTiempo().esMayor(ordenados.get(j).getUnTiempo())) {
198                     ordenados.add(j, i);
199                     break;
200                 }
201                 if(j==ordenados.size()-1) {
202                     ordenados.add(i);
203                     break;
204                 }
205             }
206             if(ordenados.size()==0) {
207                 ordenados.add(i);
208             }
209 
210         }
211         return ordenados;
212     }
213 }
214 
```