```java
package Modelo;

import Interfaz.Escritor;
import static java.lang.Thread.sleep;
import java.util.ArrayList;
import java.util.concurrent.ArrayBlockingQueue;
import java.util.concurrent.BrokenBarrierException;
import java.util.concurrent.CyclicBarrier;
import java.util.logging.Level;
import java.util.logging.Logger;

/**
 *
 * @author marco
 */
public class Soga {
    private int tamEquipo;
    private Escritor escritor;
    private Paso paso;
    private ArrayBlockingQueue<Ninno> colaSoga;
    private ArrayBlockingQueue<Ninno> colaSogaEquipoA;
    private ArrayBlockingQueue<Ninno> colaSogaEquipoB;
    private ArrayList<Monitor> monSoga = new ArrayList<>();
    private CyclicBarrier barreraSoga;

    public Soga(int p_tamEquipo, Escritor p_escritor, Paso p_paso){
        tamEquipo = p_tamEquipo;
        colaSoga = new ArrayBlockingQueue(tamEquipo*2);
        colaSogaEquipoA = new ArrayBlockingQueue(tamEquipo);
        colaSogaEquipoB = new ArrayBlockingQueue(tamEquipo);
        barreraSoga = new CyclicBarrier(1+tamEquipo*2);
        escritor = p_escritor;
        paso = p_paso;
    }

    public void soga(Ninno ninno){
        paso.mirar();
        if (colaSoga.size()+colaSogaEquipoA.size()+colaSogaEquipoB.size()<tamEquipo*2){
            try {
                colaSoga.put(ninno);
            } catch (InterruptedException ex) {
                Logger.getLogger(Soga.class.getName()).log(Level.SEVERE, null, ex);
            }
            escritor.addMsg(ninno.getMiId() + " se pone a la cola de SOGA");
            paso.mirar();
            try {
                barreraSoga.await();
            } catch (InterruptedException ex) {
                Logger.getLogger(Soga.class.getName()).log(Level.SEVERE, null, ex);
            } catch (BrokenBarrierException ex) {
                Logger.getLogger(Soga.class.getName()).log(Level.SEVERE, null, ex);
            }
        }
    }

    public void soga(Monitor mon) {
        paso.mirar();
        monSoga.add(mon);
        escritor.addMsg(mon.getMiId() + " llega a la SOGA");
        paso.mirar();
        while (mon.getContadorActividades() > 0) {
            for (int i = 0; i < tamEquipo * 2; i++) {
                paso.mirar();
                Ninno n;
                try {
                    n = colaSoga.take();
                    if (Math.random() < 0.5) {
                        if (!colaSogaEquipoA.offer(n)) {
                            colaSogaEquipoB.put(n);
                        }
                    } else {
                        if (!colaSogaEquipoB.offer(n)) {
                            colaSogaEquipoA.put(n);
                        }
```

```
                    }
                } catch (InterruptedException ex) {
                    Logger.getLogger(Soga.class.getName()).log(Level.SEVERE, null, ex);
                }
            }
            escritor.addMsg(mon.getMiId() + " inicia el enfrentamiento en SOGA");
            paso.mirar();
            try {
                sleep(7000);
            } catch (InterruptedException ex) {
                Logger.getLogger(Soga.class.getName()).log(Level.SEVERE, null, ex);
            }
            paso.mirar();
            if (Math.random() < 0.5) {
                escritor.addMsg(mon.getMiId() + " termina el enfrentamiento en SOGA a favor del EQUIPO A");
                for (Ninno ninno : colaSogaEquipoA) {
                    ninno.substractActividad(2);
                    colaSogaEquipoA.remove(ninno);
                    escritor.addMsg(ninno.getMiId() + " gana en SOGA");
                }
                for (Ninno ninno : colaSogaEquipoB) {
                    ninno.substractActividad(1);
                    colaSogaEquipoB.remove(ninno);
                    escritor.addMsg(ninno.getMiId() + " pierde en SOGA");
                }
            } else {
                escritor.addMsg(mon.getMiId() + " termina el enfrentamiento en SOGA a favor del EQUIPO B");
                for (Ninno ninno : colaSogaEquipoA) {
                    ninno.substractActividad(1);
                    colaSogaEquipoA.remove(ninno);
                    escritor.addMsg(ninno.getMiId() + " pierde en SOGA");
                }
                for (Ninno ninno : colaSogaEquipoB) {
                    ninno.substractActividad(2);
                    colaSogaEquipoB.remove(ninno);
                    escritor.addMsg(ninno.getMiId() + " gana en SOGA");
                }
            }
            paso.mirar();
            try {
                barreraSoga.await();
            } catch (InterruptedException ex) {
                Logger.getLogger(Soga.class.getName()).log(Level.SEVERE, null, ex);
            } catch (BrokenBarrierException ex) {
                Logger.getLogger(Soga.class.getName()).log(Level.SEVERE, null, ex);
            }
            mon.substractActividad();
        }
        monSoga.remove(mon);
        paso.mirar();
    }

    public String getCola(){
        String msg = "";
        for (Ninno ninno:colaSoga){
            msg += ninno.getMiId() + " ";
        }
        return msg;
    }
    public int getTamCola(){
        return colaSoga.size();
    }
    public int cuantosNinnosCola(){
        return colaSoga.size();
    }

    public String getColaEquipoA(){
        String msg = "";
        for (Ninno ninno:colaSogaEquipoA){
            msg += ninno.getMiId() + " ";
        }
        return msg;
    }

    public String getColaEquipoB(){
```

```
150         String msg = "";
151         for (Ninno ninno:colaSogaEquipoB){
152             msg += ninno.getMiId() + " ";
153         }
154         return msg;
155     }
156 
157     public String getMon(){
158         String msg = "";
159         for (Monitor mon:monSoga){
160             msg += mon.getMiId() + " ";
161         }
162         return msg;
163     }
164 }
```