

Examen Tema 5:

Programación Orientada a Objetos

Izan López Mora 1DAM

Clase KmSolidariosIESMutxamel:

```
package org.example.carrera_solidaria;

public class KmSolidariosIESMutxamel {
    public static void main(String[] args) {

        //Creación de 2 corredores: Uno con importe establecido y el otro con importe por defecto.
        Corredor corredor1 = new Corredor( nombre: "Pepe", importe_vuelta: 0.10);
        Corredor corredor2 = new Corredor( nombre: "Paco");
        System.out.println(corredor1);
        System.out.println(corredor2);
        //Agregación de patrocinadores al corredor 2
        corredor2.añadirPatrocinador();
        System.out.println(corredor2);

        //Creación de una organización.
        Organizador organizador1 = new Organizador( edicion: "IV");
        System.out.println(organizador1);
        //Agregación de 2 corredores a la organización.
        organizador1.agregarCorredores(corredor1);
        organizador1.agregarCorredores(corredor2);
        System.out.println(organizador1);
        //Muestra de corredores inscritos.
        organizador1.corredoresInscritos();
        //Establecimiento de vueltas completadas por el corredor 1.
        corredor1.setVueltas_completadas(10);
        System.out.println(corredor1);
        //Calculo de la donación total del corredor 1.
        organizador1.calcularDonacion(corredor1);

    }
}
```

Clase Organizador:

```
package org.example.carrera_solidaria;

import java.util.ArrayList;

public class Organizador { 2 usages  ± A719\1dam
    public final String ORG_DEF = "IES MUTXAMEL"; 1 usage

    private String nombre; 3 usages
    private String edicion; 3 usages
    private ArrayList<Corredor> listaCorredores; 7 usages

    public Organizador(String edicion){ 1 usage  ± A719\1dam
        nombre=ORG_DEF;
        this.edicion=edicion;
        listaCorredores = new ArrayList<>();
    }

    public String getNombre() { no usages  ± A719\1dam
        return nombre;
    }

    public String getEdicion() { no usages  ± A719\1dam
        return edicion;
    }

    public ArrayList<Corredor> getListaCorredores() { no usages  ± A719\1dam
        return listaCorredores;
    }

    public void agregarCorredores(Corredor corredor){ 2 usages  ± A719\1dam

        if(listaCorredores.size()>699){
            System.out.println("No se pueden agregar más corredores. Se ha excedido el cupo de 700.");
        } else {
            for (Corredor c : listaCorredores){
                if (c.getNombre().equals(corredor.getNombre()) && c.getDorsal()==corredor.getDorsal()){
                    System.out.println("El corredor ya está en la lista.");
                    return;
                }
            }
        }
        listaCorredores.add(corredor);
    }
}
```

```
    }

}

public void corredoresInscritos() { 1 usage ± A719\1dam
    for (Corredor c : listaCorredores) {
        System.out.println("[ "+c.getDorsal()+" ]. " + c.getNombre());
    }
}

public void calcularDonacion(Corredor corredor) { 1 usage ± A719\1dam
    double donacion_total = corredor.getImporte_vuelta()*corredor.getVueltas_completadas();
    System.out.println("El corredor " + corredor.getNombre() + " ha conseguido una donación de " + donacion_total+"€");
}

@Override ± A719\1dam
public String toString() {
    return "Organizador{" +
        "nombre=" + nombre + '\ ' +
        "edicion=" + edicion + '\ ' +
        "listaCorredores=" + listaCorredores +
        '}';
}
}
```

Clase Corredor:

```
package org.example.carrera_solidaria;

import java.util.ArrayList;
import java.util.Scanner;

public class Corredor { 10 usages  A719\1dam

    static Scanner entrada = new Scanner(System.in); 2 usages
    public static int numDorsal = 1; 4 usages

    private String nombre; 4 usages
    private int dorsal; 4 usages
    private ArrayList<String> listaPratocinadores; 5 usages
    private double importe_vuelta; 4 usages
    private int vueltas_completadas; 3 usages

    public Corredor(String nombre, double importe_vuelta){ 1 usage  A719\1dam
        this.nombre=nombre;
        dorsal=numDorsal;
        numDorsal++;
        this.importe_vuelta=importe_vuelta;
        listaPratocinadores = new ArrayList<>();
    }

    public Corredor(String nombre){ 1 usage  A719\1dam
        this.nombre=nombre;
        dorsal=numDorsal;
        numDorsal++;
        importe_vuelta=0.50;
        listaPratocinadores = new ArrayList<>();
    }

    public double getImporte_vuelta() { 1 usage  A719\1dam
        return importe_vuelta;
    }
```

```
        return importe_vuelta;
    }

    public int getVueltas_completadas() { 1 usage  ± A719\1dam
        return vueltas_completadas;
    }

    public ArrayList<String> getListaPatrocinadores() { no usages  ± A719\1dam
        return listaPatrocinadores;
    }

    public int getDorsal() { 3 usages  ± A719\1dam
        return dorsal;
    }

    public String getNombre() { 4 usages  ± A719\1dam
        return nombre;
    }

    public void setVueltas_completadas(int vueltas_completadas) { 1 usage  ± A719\1dam
        this.vueltas_completadas = vueltas_completadas;
    }

    public void añadirPatrocinador(){ 1 usage  ± A719\1dam

        boolean añadir = true;
        String nombre = "";
        String opc = "";
        do {
            if(añadir){
                System.out.println("Nombre del patrocinador a añadir:");
                nombre=entrada.next();
                listaPatrocinadores.add(nombre);
            }

            System.out.println("¿Quieres añadir más patrocinadores?");
            System.out.println("[S] / [N]");
            opc = entrada.next();
            switch(opc.toUpperCase()){
                case "S":
                    añadir = true;
            }
        } while (opc.equals("S"))
    }
```

```
        break;
    case "N":
        System.out.println("Saliendo del programa...");
        añadir=false;
        break;
    default:
        System.out.println("Opción no admitida, saliendo del programa...");
        añadir=false;
        break;
    }

}while (añadir);
}

@Override
public String toString() {
    return "Corredor{" +
        "dorsal=" + dorsal +
        ", listaPratocinadores=" + listaPratocinadores +
        ", importe_vuelta=" + importe_vuelta +
        ", vueltas_completadas=" + vueltas_completadas +
        ", nombre=" + nombre + '\n' +
        '}';
}
}
```