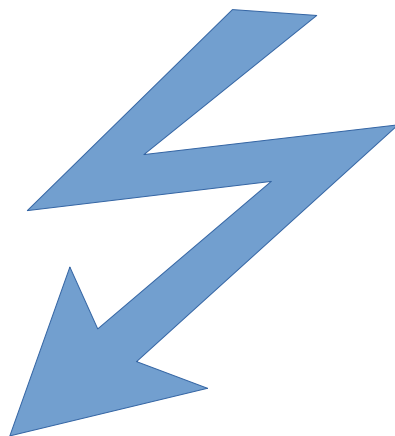


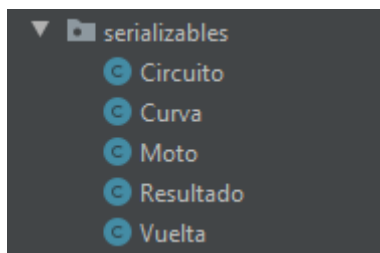
ACCESO A DATOS → PRÁCTICA UD1

Iván Mejía
3ºDAM Nocturno

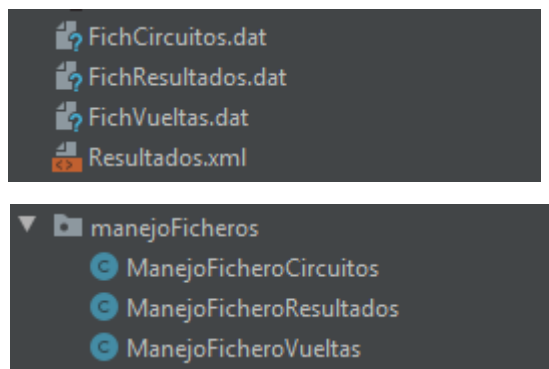


La aplicación codificada para la presente práctica pretende controlar, a través del acceso y manipulación de ficheros, una serie de inputs obtenidos a través de sensores ubicados en una motocicleta de competición a lo largo de diferentes circuitos y vueltas.

Se han implementado en total 5 clases serializables:



Se han planteado 4 tipos de archivos los cuales son manejados a través de 3 clases:



La aplicación empieza haciendo una simulación en base a unos parámetros establecidos:

```
int cantMotos = 2;
//int cantVueltas = ((int) (Math.random() * (16 - 22 + 1)) + 16);
int cantVueltas = 10;
int cantidadCircuitos = 5;

for (int m = 0; m < cantMotos; m++) {
    for (int c = 0; c < cantidadCircuitos; c++) {
        for (int v = 0; v < cantVueltas; v++) {
            Moto moto = cargarDatosMoto((m + 1), (c + 1), (v + 1));
            mostrarMoto(moto);
            System.out.println("-----");
            Resultado r = cargarDatosResultado(moto, v);
            mostrarResultado(r);
            System.out.println("--- FIN VUELTA " + (v + 1) + " ---");
        }
        System.out.println("--- FIN CIRCUITO " + (c + 1) + " MOTO " + (m + 1) + " ---" + "\n");
    }
    System.out.println("--- FIN CAMPEONATO MOTO " + (m + 1) + " ---" + "\n");
}
```

El método cargarDatosMoto() realiza la carga secuenciada de todos los valores requeridos para la simulación:

```

//cargarDatosCircuito
private static Circuito cargarDatosCircuito(int codigoCircuito) {...}

//cargarDatosVuelta
private static Vuelta cargarDatosVuelta(int numeroVuelta, Circuito c, Moto m) {...}

//cargarDatosMoto
private static Moto cargarDatosMoto(int codigoMoto, int codigoCircuito, int numeroVuelta) {...}

public static Resultado cargarDatosResultado(Moto m, int codigoResultado) {...}

private static Map<Curva, Float> calcularHP(Moto m) {...}

private static Map<Curva, Boolean> calcularDerrapada(Moto m) {...}

private static float calcularVelMedia(Moto m) {...}

private static void mostrarMoto(Moto m) {...}

private static void mostrarResultado(Resultado r) {...}

private static void mostrarCircuitoFromFile(){...}

private static void mostrarVueltasFromFile(){...}


private static void mostrarResultadoFromFile(){...}


private static void mostrarResultadoFromXML(){...}

```

Este método es el que inicia toda la simulación debido a que todo gira entorno a la entidad principal moto.

Los métodos que manejan los ficheros implementan la lectura y escritura de los distintos ficheros .dat y en el caso de la clase ManejoFicheroResultados → esta implementa además dos métodos a parte para escribir y leer desde un archivo .xml.

```
public class ManejoFicheroCircuitos {  
      
    public void escribirFicheroCircuitos(Circuito c){...}  
  
    public List<Circuito> leerFicheroCircuitos(){...}  
}
```

```
public class ManejoFicheroResultados {  
      
    public void escribirFicheroResultado(Resultado r) {...}  
  
    public List<Resultado> leerFicheroResultado() {...}  
  
    public void escribirFicheroResultadoXML() {...}  
  
    public List<Resultado> leerFicheroResultadoXML() {...}  
}
```

El manejo específico de la escritura y lectura de los distintos ficheros se ha realizado a través de objetos y listas de objetos:

(Notese que al metodo se le pasa un objeto y este es escrito en el fichero)

```
public void escribirFicheroVueltas(Vuelta v){
    //objeto file
    File fichero = new File ( pathname: "FichVueltas.dat");
    //Flujo de salida
    try {
        FileOutputStream fileout = new FileOutputStream(fichero, append: true);
        //Conecta el flujo de bytes al flujo de datos
        ObjectOutputStream dataOS = new ObjectOutputStream(fileout);

        dataOS.writeObject(v);
        dataOS.close();
    } catch (IOException e) {
        System.out.println(" - Fin fichero Vueltas - ");
        //e.printStackTrace();
    }
}
```

(Notese que el método que lee los archivos devuelve listas de objetos)

```
public List<Vuelta> leerFicheroVueltas(){
    Vuelta v = new Vuelta();
    List<Vuelta> vueltas = new ArrayList<>();
    //Objeto File
    File fichero = new File ( pathname: "FichVueltas.dat");
    //Flujo de entrada
    try {
        FileInputStream filein = new FileInputStream(fichero);
        //Conecta el flujo de bytes al flujo de datos

        try {
            while (true) {
                ObjectInputStream dataIS = new ObjectInputStream(filein);
                v = (Vuelta) dataIS.readObject();
                vueltas.add(v);
            }
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        }
    } catch (IOException e) {
        e.printStackTrace();
    }

    return vueltas;
}
```

La gestión del archivo .xml se realiza en base al archivo .dat correspondiente:

```
public void escribirFicheroResultadoXML() {  
    File fichero = new File( pathname: "FichResultados.dat");  
  
    ObjectInputStream dataIS = null;  
    FileInputStream filein = null;  
    try {  
        filein = new FileInputStream(fichero);  
    } catch (IOException e) {  
        System.out.println(" - Fin fichero Resultados (XML) - ");  
        //e.printStackTrace();  
    }  
}
```

Al leer desde el archivo .xml este también devuelve listas de objetos:

```
public List<Resultado> leerFicheroResultadoXML() {  
    List<Resultado> lr = new ArrayList<>();  
  
    XStream xstream = new XStream();  
  
    xstream.alias( name: "ListaResultados", ListaResultados.class);  
    xstream.alias( name: "Resultado", Resultado.class);  
    //xstream.addImplicitCollection(ListaResultados.class, "Lista");  
  
    try {  
        FileInputStream fichero = new FileInputStream( name: "Resultados.xml");  
        ListaResultados listaResultados = (ListaResultados) xstream.fromXML(fichero);  
        lr = listaResultados.getListaResultados();  
    } catch (FileNotFoundException e) {  
        e.printStackTrace();  
    }  
    return lr;  
}
```

Ejemplos de la salida que se obtiene de la aplicación:

MOTO → CIRCUITO → CURVAS

```
Código Moto -> 1
Codigo Circuito -> 1, Logitud total -> 1708 mts, Curvas Totales -> 26
Número Curva -> 1, Lado Curva -> DERECHA, Torque -> 59 NM, RPM -> 8318, velRuedaDel -> 310 KM/H, velRuedaTras -> 310 KM/H
Número Curva -> 2, Lado Curva -> IZQUIERDA, Torque -> 47 NM, RPM -> 9506, velRuedaDel -> 263 KM/H, velRuedaTras -> 243 KM/H
Número Curva -> 3, Lado Curva -> IZQUIERDA, Torque -> 58 NM, RPM -> 5446, velRuedaDel -> 304 KM/H, velRuedaTras -> 304 KM/H
Número Curva -> 4, Lado Curva -> IZQUIERDA, Torque -> 18 NM, RPM -> 13081, velRuedaDel -> 342 KM/H, velRuedaTras -> 342 KM/H
Número Curva -> 5, Lado Curva -> IZQUIERDA, Torque -> 28 NM, RPM -> 9584, velRuedaDel -> 253 KM/H, velRuedaTras -> 233 KM/H
Número Curva -> 6, Lado Curva -> IZQUIERDA, Torque -> 57 NM, RPM -> 2008, velRuedaDel -> 133 KM/H, velRuedaTras -> 113 KM/H
Número Curva -> 7, Lado Curva -> DERECHA, Torque -> 43 NM, RPM -> 9352, velRuedaDel -> 260 KM/H, velRuedaTras -> 260 KM/H
Número Curva -> 8, Lado Curva -> DERECHA, Torque -> 2 NM, RPM -> 15107, velRuedaDel -> 320 KM/H, velRuedaTras -> 320 KM/H
Número Curva -> 9, Lado Curva -> IZQUIERDA, Torque -> 17 NM, RPM -> 10876, velRuedaDel -> 213 KM/H, velRuedaTras -> 193 KM/H
Número Curva -> 10, Lado Curva -> IZQUIERDA, Torque -> 45 NM, RPM -> 15148, velRuedaDel -> 174 KM/H, velRuedaTras -> 174 KM/H
Número Curva -> 11, Lado Curva -> DERECHA, Torque -> 5 NM, RPM -> 17998, velRuedaDel -> 93 KM/H, velRuedaTras -> 73 KM/H
Número Curva -> 12, Lado Curva -> DERECHA, Torque -> 1 NM, RPM -> 11627, velRuedaDel -> 198 KM/H, velRuedaTras -> 198 KM/H
Número Curva -> 13, Lado Curva -> IZQUIERDA, Torque -> 56 NM, RPM -> 10089, velRuedaDel -> 233 KM/H, velRuedaTras -> 213 KM/H
Número Curva -> 14, Lado Curva -> DERECHA, Torque -> 33 NM, RPM -> 13445, velRuedaDel -> 15 KM/H, velRuedaTras -> -5 KM/H
Número Curva -> 15, Lado Curva -> IZQUIERDA, Torque -> 45 NM, RPM -> 17076, velRuedaDel -> 15 KM/H, velRuedaTras -> -5 KM/H
Número Curva -> 16, Lado Curva -> IZQUIERDA, Torque -> 23 NM, RPM -> 7146, velRuedaDel -> 136 KM/H, velRuedaTras -> 136 KM/H
Número Curva -> 17, Lado Curva -> DERECHA, Torque -> 17 NM, RPM -> 14637, velRuedaDel -> 341 KM/H, velRuedaTras -> 321 KM/H
Número Curva -> 18, Lado Curva -> DERECHA, Torque -> 31 NM, RPM -> 14148, velRuedaDel -> 279 KM/H, velRuedaTras -> 259 KM/H
Número Curva -> 19, Lado Curva -> IZQUIERDA, Torque -> 43 NM, RPM -> 11950, velRuedaDel -> 295 KM/H, velRuedaTras -> 275 KM/H
Número Curva -> 20, Lado Curva -> IZQUIERDA, Torque -> 15 NM, RPM -> 8660, velRuedaDel -> 273 KM/H, velRuedaTras -> 253 KM/H
Número Curva -> 21, Lado Curva -> DERECHA, Torque -> 49 NM, RPM -> 8582, velRuedaDel -> 133 KM/H, velRuedaTras -> 113 KM/H
Número Curva -> 22, Lado Curva -> DERECHA, Torque -> 11 NM, RPM -> 17047, velRuedaDel -> 44 KM/H, velRuedaTras -> 44 KM/H
Número Curva -> 23, Lado Curva -> DERECHA, Torque -> 30 NM, RPM -> 17350, velRuedaDel -> 163 KM/H, velRuedaTras -> 143 KM/H
Número Curva -> 24, Lado Curva -> DERECHA, Torque -> 56 NM, RPM -> 15754, velRuedaDel -> 186 KM/H, velRuedaTras -> 186 KM/H
```

RESULTADO → CURVAS:

```
Codigo resultado -> 0
Velocidad Media -> 12.351888
Número Curva -> 3, HP -> 60.14242, ¿Derrapa? -> false
Número Curva -> 13, HP -> 107.57502, ¿Derrapa? -> true
Número Curva -> 9, HP -> 35.204113, ¿Derrapa? -> true
Número Curva -> 5, HP -> 51.095203, ¿Derrapa? -> true
Número Curva -> 6, HP -> 21.792841, ¿Derrapa? -> true
Número Curva -> 24, HP -> 167.97867, ¿Derrapa? -> false
Número Curva -> 2, HP -> 85.068924, ¿Derrapa? -> true
Número Curva -> 12, HP -> 2.2138233, ¿Derrapa? -> false
Número Curva -> 1, HP -> 93.44288, ¿Derrapa? -> false
Número Curva -> 8, HP -> 5.7528563, ¿Derrapa? -> false
Número Curva -> 21, HP -> 80.06816, ¿Derrapa? -> true
Número Curva -> 15, HP -> 146.30998, ¿Derrapa? -> true
Número Curva -> 17, HP -> 47.377953, ¿Derrapa? -> true
Número Curva -> 18, HP -> 83.50876, ¿Derrapa? -> true
Número Curva -> 4, HP -> 44.832066, ¿Derrapa? -> false
Número Curva -> 20, HP -> 24.733435, ¿Derrapa? -> true
Número Curva -> 7, HP -> 76.56816, ¿Derrapa? -> false
Número Curva -> 10, HP -> 129.79056, ¿Derrapa? -> false
Número Curva -> 16, HP -> 31.294365, ¿Derrapa? -> false
Número Curva -> 23, HP -> 99.1051, ¿Derrapa? -> true
Número Curva -> 25, HP -> 90.77209, ¿Derrapa? -> true
Número Curva -> 19, HP -> 97.83892, ¿Derrapa? -> true
Número Curva -> 22, HP -> 35.703922, ¿Derrapa? -> false
Número Curva -> 14, HP -> 84.47925, ¿Derrapa? -> true
```