

NAME		GRADE	
		1ST - DAM / DAW	
SURNAMES		SUBJECT	ANNOUNCEMENT
		PROGRAMMING	2ND EVALUATION
DNI	DATE	GRADE	
	06/03/2024		

From a CSV file looking like this:

Tipo	Nombre	Apellidos	NIA	DNI	Edad	Mail	Curso	Ciclo
Alumno	Pedro	Lopez Sanz	111234244		18	plopez@alu.edu.gva.es	1	DAW
Alumno	Luis	Martinez Gonzalez	111424551		22	lmartinez@alu.edu.gva.es	2	DAM
Docente	Carlos	Martinez Ramos		67453398D	52	cmartinez@edu.gva.es		
Alumno	Carmen	Garcia Lujan	114223589		23	cgarcia@alu.edu.gva.es	2	DAW
Docente	Juan	Hueso Ramos		43382987F	36	jhueso@edu.gva.es		
Alumno	Lorena	Tarrega Navarro	114323451		19	ltarrega@alu.edu.gva.es	1	ASIR
Alumno	Ramon	Anton Navarro	122344213		22	ranton@alu.edu.gva.es	1	ASIR
Alumno	Jorge	Tarrega Lopez	112325216		18	jtarrega@alu.edu.gva.es	1	ASIR
Alumno	Sergio	Tarrega Nan	111251266		21	starrega@alu.edu.gva.es	1	ASIR
Alumno	Roberto	Jimenez Navarro	119399794		20	rjimenez@alu.edu.gva.es	1	ASIR

You must create an application to manage higher education institutions. In concrete, some of components like printing the students' cards. To do that:

### 1. CREATE THE DATA MODEL (4 POINTS)

- (1p) Implement an **abstract class** called **Persona** having following elements:
  - attributes: nombre (String), apellidos (String), edad (int) and email (String). All of them with a private visibility except nombre which can be accessed from the subclasses and classes in the same package.
  - constructor with all fields.
  - an abstract method called getID having no parameters and returning a String.
  - Implement getters for all the attributes.
  - Implement equals method knowing that a person is equals to another person if their IDs are the same.
  - Implements those methods needed **depending on the interfaces you think you need to implement** or any other method depending on the Collections you will need.
- (0,50p) Implement a **class** called **Docente** which inherits from Persona having:
  - An attribute called DNI (String). This DNI attribute is the Docente ID.
  - Constructor with all the needed attributes
  - Implement those methods you think are needed.

NAME & SURNAMES

- c. **(0.50p)** Implement a **class** called **Alumno** which inherits from **Persona** having:
- Attributes: a NIA (String), and a Titulo. This NIA attribute is the Alumno ID.
  - Constructor with all the needed attributes.
  - Implement getters for all the attributes.
  - Implement those methods you think are needed.
- d. **(0.50p)** Implement an **enum class** called **Ciclo** having following values: DAM, DAW, ASIR, SMR.
- e. **(0.50p)** Implement an **enum class** called **Curso** having following values: PRIMERO, SEGUNDO. This enum class will have an Integer attribute containing the year course, and therefore we need the constructor, the toString method and a static method called getCursoFromInt. This method will have an integer as parameter and returning a Curso value. Additionall, this method will throw an Exception (from Exception class) in case the integer parameter is not an appropriate value (1,2).
- f. **(0.50p)** Implement an enum class called **Titulo** having following values: DAM\_1, DAW\_1, ASIR\_1, SMR\_1, DAM\_2, DAW\_2, ASIR\_2, SMR\_2. This enum class will have two attributes: Curso and Ciclo. This class needs to have a constructor, the toString method and a static method called getTitulo having as parameters Curso and a Ciclo variables and returning a Titulo value.
- g. **(0.50p)** Implement an **Interface** called **Imprimible** having two methods which will be coded by all the classes implementing it:
- getFullName (returning a String, having no parameters)
  - getTipo (returning a String, having no parameters)

NAME & SURNAMES

2. **(6 POINTS)** Perform the following methods which theoretically will be implemented in a Main class:

- (0.75p)** A method (getPersonsSorted) to get a sorted by name collection of Imprimible objects from a set of Persona. This is the **natural sort** for a Person. Perform any change in the data model classes if needed.
- (0.75p)** A method (getAlumnosSortedByAge) to get a sorted by age list of Imprimible objects from a set of Persona. Perform any change in the data model classes if needed.
- (0.75p)** A method (getAlumnosTitulo) to get a map of Titulo and a list of Alumno from a set of Persona (Map<Titulo,List<Alumno>).
- (0.75p)** A method (imprimirCarnets) to print by console (so, it returns nothing) all the data from a collection of Imprimible. The result should be the type and the fullname, in that way:

```
[...]  
-----  
Alumno: Luis Martinez Gonzalez  
-----  
Docente: Carlos Martinez Ramos  
-----  
Alumno: Jorge Tarrega Lopez  
-----  
[...]
```

- (0.75p)** A method (saveAsObject) to save all the Persona objects in a file (called personas.dat). This method has no parameters and returns a set of Persona. All the potential exceptions are managed in this method. Perform any change in the data model classes if needed.
- (0.75p)** A method (loadObjectFile) to load all the Persona objects in a file (called personas.dat). This method has no parameters and returns a set of Persona. All the potential exceptions are managed in this method. Perform any change in the data model classes if needed.

NAME & SURNAMES

- g. **(0.75p)** A method (cargarDatos) to get a set of Personas from a file name (String). The file, a CVS one, will look like the one in the header of this exam:

```
Tipo,Nombre,Apellidos,NIA,DNI,Edad,Mail,Curso,Ciclo
Alumno,Pedro,Lopez Sanz,111234244,,18,plopez@alu.edu.gva.es,1,DAW
Docente,Carlos,Martinez Ramos,,67453398D,52,cmartinez@edu.gva.es,,
Alumno,Carmen,Garcia Lujan,114223589,,23,cgarcia@alu.gva.es,2,DAW
Docente,Juan,Hueso Ramos,,43382987F,36,jhueso@edu.gva.es,,
[...]
```

- h. **(0.75p)** A method (guardarDatosAlumno) to save a CSV file from a collection of Alumno. So, this method returns nothing, and the parameter is a collection of Alumno. File name is always fixed (alumnos.csv). The file will have a header as shown in the example below. Method will handle all the potential exceptions. Example:

```
Nombre,Apellidos,NIA,Edad,Mail,Curso,Ciclo
Roberto,Jimenez Navarro,119399794,20,rjimenez@alu.gva.es,1,ASIR
Ramon,Anton Navarro,122344213,22,ranton@alu.gva.es,1,ASIR
Lorena,Tarrega Navarro,114323451,19,ltarrega@alu.gva.es,1,ASIR
```