



--	--	--	--	--

APELLIDOS: _____ NOMBRE: _____

DNI: _____ TITULACIÓN: ☐ IC ☐ IS ☐ TI ☐ DG GRUPO: ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

Para cada profesor de un departamento universitario se tiene la siguiente información:

- **Nombre:** nombre del profesor, de tipo String, consultable.
- **Fecha ingreso:** fecha de ingreso del profesor en el departamento, de tipo LocalDate, consultable.
- **Asignaturas cursos anteriores:** asignaturas impartidas en cursos anteriores, de tipo Set<Asignatura>, consultable.
- **Asignaturas primer cuatrimestre:** asignaturas impartidas en el primer cuatrimestre del curso actual, de tipo Set<Asignatura>, consultable.
- **Asignaturas segundo cuatrimestre:** asignaturas impartidas en el segundo cuatrimestre del curso actual, de tipo Set<Asignatura>, consultable.
- **Créditos teóricos primer cuatrimestre:** créditos de docencia teórica asignados en el primer cuatrimestre, de tipo Double, consultable.
- **Créditos teóricos segundo cuatrimestre:** créditos de docencia teórica asignados en el segundo cuatrimestre, de tipo Double, consultable.
- **Créditos laboratorio primer cuatrimestre:** créditos de docencia práctica asignados en el primer cuatrimestre, de tipo Double, consultable.
- **Créditos laboratorio segundo cuatrimestre:** créditos de docencia práctica asignados en el segundo cuatrimestre, de tipo Double, consultable.
- **Capacidad:** capacidad docente en el curso actual en número de créditos, de tipo Double, consultable.
- **Asignaturas coordinadas:** asignaturas que coordina en el curso actual, de tipo Set<Asignatura>, consultable.
- **Experiencia:** número de años que lleva en el departamento, de tipo Integer, consultable. Se calcula a partir de la fecha de ingreso.
- **Ocioso:** un profesor se considera ocioso si y sólo si sus créditos asignados en el curso actual no superan la mitad de su capacidad docente, de tipo Boolean, consultable.
- **Asignaturas impartidas:** todas las asignaturas impartidas por el profesor en el curso actual (considerando los dos cuatrimestres), de tipo Set<Asignatura>, consultable.
- **Coordinador:** propiedad cierta si y sólo si el profesor es coordinador de alguna asignatura, de tipo Boolean, consultable.
- **Créditos teóricos:** número total de créditos teóricos asignados al profesor (considerando los dos cuatrimestres), de tipo Double, consultable.
- **Créditos asignados:** número total de créditos asignados al profesor (considerando tanto teoría como laboratorio en los dos cuatrimestres), de tipo Double, consultable.

Las asignaturas del departamento (Asignatura) se formalizan como un enumerado de acrónimos: FP, ADDA, IISSI1, IISSI2, AII, PSG1, PSG2, DP1, DP2, IR, IS, SO, PL, GSI, SIE, SOS

La información de cada profesor se deriva de un fichero CSV con líneas conteniendo los siguientes datos: nombre, fecha ingreso, asignaturas cursos anteriores, asignaturas primer cuatrimestre, asignaturas segundo cuatrimestre, créditos teóricos primer cuatrimestre, créditos teóricos segundo cuatrimestre, créditos laboratorio primer cuatrimestre, créditos laboratorio segundo cuatrimestre, capacidad y asignaturas coordinadas. A continuación, se muestran algunas líneas de ejemplo del fichero CSV.

```
Jesus Garcia Gamez, 12/01/2006, {FP},{FP},{FP},3.6,3.6,2.4,2.4,24,{}  
Fernando Suarez Ruiz, 12/01/2005, {SIE; FP},{IISSI1; FP}, {FP; SIE},3.6,3.6,12,2.4,24,{IISSI2}  
Vicente Colomo Mol, 12/01/1994, {FP;AII}, {AII},{AII;IISSI2;SOS},6,6,4.8,4.8,24,{AII;SOS}
```

Cree en el proyecto los paquetes fp.departamento, fp.departamento.test y fp.utiles, y dentro de ellos implemente los tipos que se piden a continuación.

Ejercicio 1: Tipo base (1,5puntos)

Implemente el tipo Profesor mediante un *record*, de acuerdo con la siguiente información:

Constructores:

- C1: recibe un parámetro por cada propiedad básica del tipo, en el mismo orden en el que están definidas.

Restricciones:

- R1: la fecha de ingreso debe ser anterior a la fecha actual.
- R2: la suma de créditos de docencia asignados debe ser mayor o igual que 0.
- R3: la capacidad docente debe ser mayor o igual que 0.

Criterio de igualdad: dos profesores son iguales si tienen el mismo nombre.

Orden natural: los profesores se ordenan lexicográficamente por su nombre.

Representación como cadena: una cadena con todas las propiedades básicas del tipo.

Ejercicio 2: Tipo contenedor (1 punto)

Implemente el tipo Departamento de acuerdo con la siguiente descripción:

Propiedades:

- profesores: conjunto de profesores del departamento, de tipo Set<Profesor>, consultable.

Constructores:

- C1: no recibe parámetros. Crea un conjunto de profesores inicialmente vacío.
- C2: recibe un parámetro de tipo Stream<Profesor> para inicializar el conjunto de profesores.

Criterio de igualdad: dos departamentos son iguales si lo son el conjunto de sus profesores.

Representación como cadena: una cadena con todas las propiedades del tipo.

Otras operaciones:

- *void añadirProfesor(Profesor p):* añade el profesor *p* al conjunto profesores.
- *void eliminarProfesor(String n):* elimina el profesor con nombre *n* del conjunto profesores. No tiene efecto si éste no existe.

Ejercicio 3: Factoría (1 puntos)

Cree una factoría *FactoriaDepartamento* que permita leer los datos de un fichero CSV y almacenarlos en el tipo contenedor. La factoría debe contener los siguientes métodos:

- *Departamento leerDepartamento(String rutaFichero):* lee un fichero con los profesores del departamento y construye un objeto de tipo Departamento.
- *Profesor parsearProfesor(String lineaCSV):* crea un objeto de tipo Profesor a partir de una cadena de caracteres. La cadena de caracteres debe tener el mismo formato que las líneas del fichero CSV. Cree un método auxiliar *Set<Asignatura> parsearConjAsignaturas(String campo)*.

Ejercicio 4: Tratamientos secuenciales (6,5 puntos)

Añada los siguientes tratamientos secuenciales al tipo contenedor. Debe resolver todos los métodos **mediante streams**, salvo que se le indique expresamente lo contrario:

1. *Map<Asignatura,List<Profesor>> profesoresPorAsignatura():* Relacionar cada asignatura con los profesores que la imparten. Resuélvalo sin usar streams. **(1 punto)**
2. *Set<Profesor> profesoresQueSololoImpartenAsignaturasQueCoordinan():* Determinar el conjunto de profesores que sólo imparten clases en las asignaturas que coordinan. **(1 punto)**
3. *Boolean departamentoResponsable():* Decidir si el departamento es responsable. Un departamento es responsable si todo profesor que coordina asignatura/s tiene una experiencia mínima de 5 años e imparte tal/es asignatura/s en el curso actual (primer o segundo cuatrimestre). **(1 punto)**
4. *List<String> ordenarProfesoresPorNumeroCreditosTeoricos():* Ordenar de forma decreciente los nombres de los profesores que dan teoría por el número de créditos teóricos asignados en el curso actual. **(1 punto)**
5. *Integer añoIncorporacionMasCoordinadores():* Obtener el año en el que se incorporaron más profesores que actualmente son coordinadores. Si no se puede calcular se eleva *NoSuchElementException*. **(1,5 puntos)**

Escriba un test para el tipo contenedor. En este test se leerán los datos del fichero CSV y se probarán todos los métodos del ejercicio 4. Cada método del ejercicio 4 tiene que tener su correspondiente método de test (como solemos hacer en clase). **(1 punto)**