



Se dispone de un fichero de datos de un grupo de estudiantes matriculados en el Grado en Tecnologías Informáticas de la Universidad de Sevilla. De cada estudiante se registra la siguiente información:

- **Nombre:** nombre del estudiante.
- **Fecha de nacimiento:** fecha de nacimiento del estudiante.
- **Repetidor:** indica si el estudiante es o no repetidor.
- **Nota media:** nota media de las asignaturas cursadas.
- **Asignaturas:** conjunto de asignaturas cursadas por el estudiante. Cada asignatura incluye su identificación mediante siglas, número de créditos y curso dentro del plan de estudio.

Por ejemplo, la siguiente línea de texto del fichero

ALEX ORTIZ MOLINA;11/1/1998;no;5.7657;[CIMSI-6-3, FP-12-1, SSII-6-4]

representa al estudiante ALEX ORTIZ MOLINA, nacido el 11 de enero de 1998, no repetidor y con 5.7657 de nota media en las asignaturas cursadas, siendo estas CIMSI de tercer curso con 6 créditos, FP de primer curso con 12 créditos y SSII de cuarto curso con 6 créditos.

Implemente los tipos que se piden a continuación.

Ejercicio 1: Tipo Asignatura (0.5 ptos)

Implemente en el paquete `fp.estudiantes` el tipo `Asignatura` con un record con la siguiente especificación:

Propiedades:

- **id:** siglas de la asignatura, de tipo `String`, consultable.
- **créditos:** número de créditos de la asignatura, de tipo `Integer`, consultable.
- **curso:** curso en el que se encuentra la asignatura dentro del plan de estudios, de tipo `Integer`, consultable.

Constructores:

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

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

Restricciones:

- R1: Los cursos deben estar en el rango [1..4].

Ejercicio 2: Tipo Estudiante (1.5 ptos)

Implemente en el paquete `fp.estudiantes` el tipo `Estudiante` con una clase con la siguiente especificación:

Propiedades:

- **nombre:** nombre del estudiante, de tipo `String`, consultable.
- **fechaNacimiento:** fecha de nacimiento del estudiante, de tipo `LocalDate`, consultable.
- **esRepetidor:** indica si el estudiante es repetidor o no, de tipo `Boolean`, consultable.
- **notaMedia:** nota media de las asignaturas cursadas, de tipo `Double`, consultable.
- **asignaturas:** asignaturas cursadas por el estudiante, de tipo `Set<Asignatura>`, consultable.
- **edad:** edad del estudiante, de tipo `Integer`, consultable.

Constructores:

- C1: recibe un parámetro por cada propiedad básica del tipo.

Restricciones:

- R1: La nota media no puede ser negativa ni superar el valor 10.
- R2: El número de asignaturas cursadas debe variar en el rango [0..10].

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

Criterio de igualdad: dos estudiantes son iguales si lo son sus nombres y fechas de nacimiento.

Criterio de ordenación: dos estudiantes se ordenan por nombre y, a igualdad de nombres, por fecha de nacimiento.

Ejercicio 3: Tipo FactoríaGrupo (1 pto)

Para facilitar la lectura del fichero de datos, se suministra la clase FactoriaGrupo parcialmente implementada en el paquete fp.estudiantes. Complete su implementación con el método:

- *Estudiante parsearEstudiante(String linea):* método que crea un objeto de tipo Estudiante a partir de una línea del fichero de datos.

NOTA: implemente otros métodos auxiliares si los necesita.

Ejercicio 4: Tipo Grupo (6.5 ptos)

Con el propósito de programar un contenedor de objetos, se pide la implementación en el paquete fp.estudiantes de un tipo llamado Grupo con la siguiente especificación:

Propiedades:

- **estudiantes:** estudiantes del grupo, de tipo List<Estudiante>, consultable.

Constructores:

- C1: recibe un parámetro de tipo Stream<Estudiante>

Representación como cadena: una cadena con todos los estudiantes del grupo separados por retornos de carro (“\n”).

Criterio de igualdad: dos grupos son iguales si lo son sus estudiantes.

Añada los siguientes tratamientos secuenciales al tipo Grupo. Debe resolver todos los métodos **mediante streams**, salvo que se le indique expresamente que debe utilizar bucles. En ningún caso está permitido el uso de forEach en los métodos del ejercicio 4:

1. *SortedSet<Estudiante> estudiantesQueCursanNAsignaturasAlMenos(Integer curso, Integer n):*
Obtiene los estudiantes que cursan al menos n asignaturas, con alguna de ellas pertenecientes al curso dado como parámetro. El conjunto resultante estará ordenado por el orden natural de los estudiantes. (1 pto)
2. *Double edadMediaEstudiantesRepetidores(Set<String> idsAsignaturas):*
Obtiene la edad media de los estudiantes repetidores que cursan al menos las asignaturas cuyos ids se dan como parámetro. Pueden haber cursado más asignaturas, pero al menos las de idsAsignaturas. Si la edad media no puede calcularse por no haber ningún estudiante en esas condiciones, devuelva null.
Implemente este método con bucles. (1 pto)
3. *Map<String, Map<Integer, List<Asignatura>>> asociarEstudiantesConAsignaturasPorCurso():*
Asocia el nombre de cada estudiante con las asignaturas que cursa agrupadas por curso. Para ello, este método devolverá un Map tal que asocia a cada nombre de alumno como clave, un valor que es otro Map que asocia cada curso como clave con la lista de asignaturas de ese curso. Suponga que no existen dos estudiantes con el mismo nombre. (1.5 ptos)

4. *String estudianteRepetidorConMasCreditos()*:

Obtiene el nombre del estudiante repetidor con más créditos cursados. Si no se puede calcular, devuelve null. (1.5 ptos)

5. *Map<Integer, List<String>> asignaturasPorCursoOrdenadas()*:

Obtiene un Map en el que, a cada curso como clave, le hace corresponder una lista con los identificadores de las asignaturas de ese curso, ordenadas por número de créditos de menor a mayor, y, a igualdad de créditos, por id. Tenga en cuenta que en la lista resultado no debe haber asignaturas duplicadas. (1.5 ptos)

Ejercicio 5: Test (0.5 ptos)

Escriba un programa de prueba del tipo contenedor en el paquete fp.estudiantes.test. En dicho programa se leerán los datos del fichero y se probarán todos los tratamientos secuenciales, definiendo un método de test por cada tratamiento secuencial a probar. Un ejemplo de resultado de prueba del tipo contenedor se incluye en un fichero llamado test.txt en la carpeta out. Un extracto de este archivo con la información principal se muestra a continuación:

EJ.4.1: Estudiantes que cursan al menos 5 asignaturas con alguna de estas pertenecientes al curso 3:

```
Estudiante [nombre=ALEJANDRA CAMPOS MEDINA, fechaNacimiento=1995-08-11, esRepetidor=false,
notaMedia=7.091318919375115, asignaturas=[Asignatura[id=CSI, creditos=6, curso=3],
Asignatura[id=SI, creditos=6, curso=3], Asignatura[id=MASI, creditos=6, curso=3],
Asignatura[id=AI, creditos=6, curso=4], Asignatura[id=PL, creditos=6, curso=3], Asignatura[id=IPO,
creditos=6, curso=4], Asignatura[id=TIS, creditos=6, curso=4], Asignatura[id=S0, creditos=6,
curso=2]]]
```

```
Estudiante [nombre=ANGELA PEÑA MARTIN, fechaNacimiento=1998-10-17, esRepetidor=false,
notaMedia=0.30003390085281745, asignaturas=[Asignatura[id=AR, creditos=6, curso=2],
Asignatura[id=T, creditos=6, curso=4], Asignatura[id=CM, creditos=6, curso=4], Asignatura[id=MCC,
creditos=6, curso=4], Asignatura[id=PL, creditos=6, curso=3]]]
```

...LOS RESULTADOS INTERMEDIOS SE OMITEN POR CLARIDAD. SE PUEDEN CONSULTAR EN EL ARCHIVO out/test.txt

```
Estudiante [nombre=ROSA MARIA CABRERA RAMIREZ, fechaNacimiento=1998-03-27,
esRepetidor=false, notaMedia=4.105453757098639, asignaturas=[Asignatura[id=AIA, creditos=6,
curso=3], Asignatura[id=SOS, creditos=6, curso=3], Asignatura[id=IMD, creditos=6, curso=1],
Asignatura[id=AR, creditos=6, curso=2], Asignatura[id=SI, creditos=6, curso=3],
Asignatura[id=IISSI2, creditos=6, curso=2], Asignatura[id=SSII, creditos=6, curso=4],
Asignatura[id=LI, creditos=6, curso=2], Asignatura[id=IPO, creditos=6, curso=4],
Asignatura[id=ADDA, creditos=12, curso=2]]]
```

```
Estudiante [nombre=SANDRA GARRIDO HIDALGO, fechaNacimiento=2000-04-29, esRepetidor=false,
notaMedia=0.1229126097389599, asignaturas=[Asignatura[id=AR, creditos=6, curso=2],
Asignatura[id=EdC, creditos=6, curso=1], Asignatura[id=SI, creditos=6, curso=3], Asignatura[id=MCC,
creditos=6, curso=4], Asignatura[id=SSII, creditos=6, curso=4], Asignatura[id=LI, creditos=6,
curso=2], Asignatura[id=CIMSI, creditos=6, curso=3]]]
```

EJ. 4.2: Edad media de los estudiantes repetidores que cursan [ADDA, FP]:

21.5

EJ. 4.3: Asociar el nombre de cada estudiante con las asignaturas que cursa agrupadas por curso:

```
MARC ROMERO MORA->{1=[Asignatura[id=AE, creditos=6, curso=1], Asignatura[id=CED, creditos=6,
curso=1]], 2=[Asignatura[id=AR, creditos=6, curso=2]], 3=[Asignatura[id=CSI, creditos=6, curso=3],
Asignatura[id=PL, creditos=6, curso=3]], 4=[Asignatura[id=MCC, creditos=6, curso=4],
Asignatura[id=SP, creditos=6, curso=4]]}
```

JOSE MARIA HERNANDEZ GOMEZ->{1=[Asignatura[id=E, creditos=6, curso=1]], 2=[Asignatura[id=AC, creditos=6, curso=2], Asignatura[id=AR, creditos=6, curso=2], Asignatura[id=ADDA, creditos=12, curso=2]], 4=[Asignatura[id=ISI, creditos=6, curso=4], Asignatura[id=CM, creditos=6, curso=4], Asignatura[id=MCC, creditos=6, curso=4], Asignatura[id=IPO, creditos=6, curso=4]]}

...LOS RESULTADOS INTERMEDIOS SE OMITEN POR CLARIDAD. SE PUEDEN CONSULTAR EN EL ARCHIVO out/test.txt

ROSA MARIA CABRERA RAMIREZ->{1=[Asignatura[id=IMD, creditos=6, curso=1]], 2=[Asignatura[id=AR, creditos=6, curso=2], Asignatura[id=IISSI2, creditos=6, curso=2], Asignatura[id=LI, creditos=6, curso=2], Asignatura[id=ADDA, creditos=12, curso=2]], 3=[Asignatura[id=AIA, creditos=6, curso=3], Asignatura[id=SOS, creditos=6, curso=3], Asignatura[id=SI, creditos=6, curso=3]], 4=[Asignatura[id=SSII, creditos=6, curso=4], Asignatura[id=IPO, creditos=6, curso=4]]}

TERESA GOMEZ CORTES->{2=[Asignatura[id=ADDA, creditos=12, curso=2]], 3=[Asignatura[id=PL, creditos=6, curso=3]], 4=[Asignatura[id=T, creditos=6, curso=4], Asignatura[id=MCC, creditos=6, curso=4]]}

EJ. 4.4: Nombre del estudiante repetidor con mayor número de créditos cursados:

PAULA GARCIA PRIETO

EJ. 4.5: Asignaturas por curso ordenadas por créditos y por id:

1->[AE, CED, CIN, E, EdC, FFI, IMD, FP]

2->[AC, AR, IISSI1, IISSI2, LI, MD, RC, SO, ADDA]

3->[AIA, CIMSI, CSI, GEE, IA, MASI, PD, PL, SI, SIE, SOS]

4->[AII, CM, IPO, ISI, MCC, SP, SSII, T, TIS]