



Tenemos un conjunto de datos sobre los exámenes realizados la E.T.S. de Ingeniería Informática. Para cada examen se tiene la siguiente información:

- **Asignatura:** acrónimo de la asignatura a la que corresponde el examen, de tipo cadena.
- **Curso:** curso al que pertenece la asignatura, de tipo entero.
- **Fecha:** fecha del examen, de tipo fecha.
- **Hora:** hora de comienzo del examen, de tipo hora.
- **Duración:** duración del examen en minutos, de tipo entero.
- **Tipo:** tipo de examen, que puede tomar los valores Teórico y Práctico.
- **Asistentes:** número de asistentes al examen, de tipo entero.
- **Inscripción:** indica si el examen requiere inscripción previa, de tipo lógico.
- **Aulas:** lista con las aulas asignadas al examen. Para cada aula se tiene su nombre, de tipo cadena, y su capacidad, de tipo entero

Por ejemplo, sean las dos líneas del fichero:

```
FP,1,29/5/2024,9:00,150,Practico,40,Si,F1.30-24;F1.31-25
```

```
FP,1,30/5/2024,9:00,150,Practico,55,Si,I2.31-18;I2.33-22;I2.35-22
```

La primera línea nos indica que el día 29 de mayo de 2024 a las 9:00 tiene lugar el examen de la asignatura FP, de primer curso. Es un examen práctico, con una duración de 150 minutos, y a él asisten 40 alumnos, que deben inscribirse previamente. Para realizar el examen se utilizan las aulas F1.30 y F1.31, con una capacidad de 24 y 25 puestos, respectivamente.

Abra el proyecto e implemente en los paquetes fp.examen y fp.examen.test los tipos que se piden a continuación.

Ejercicio 1: tipo Aula (0,5 pts)

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

Propiedades:

- **Nombre:** nombre del aula, de tipo String, consultable.
- **Capacidad:** número de puestos disponibles en el aula, 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.

Criterio de igualdad: dos aulas son iguales si lo son su nombre y su capacidad.

Restricciones:

- R1: el nombre debe comenzar por una letra.
- R2: la capacidad debe ser mayor que cero.

Ejercicio 2: tipo Examen (1,5 pts)

Implemente el tipo Examen mediante una clase, de acuerdo con la siguiente descripción:

Propiedades:



- **Asignatura**, de tipo String, consultable.
- **Curso**, de tipo Integer, consultable.
- **Fecha y hora**, de tipo LocalDateTime, consultable.
- **Duración**, de tipo Duration, consultable y modificable.
- **Tipo**, de tipo TipoExamen, consultable. Puede tomar los valores TEORICO y PRACTICO.
- **Asistentes**, de tipo Integer, consultable y modificable.
- **Inscripción**, de tipo Boolean, consultable.
- **Aulas**, de tipo List<Aula>, consultable. Lista de objetos de tipo Aula.
- **Puestos**, de tipo List<Integer>, consultable. Lista con las capacidades de las aulas del examen, que se calcula a partir de la lista de aulas.
- **Capacidad máxima**, de tipo Integer, consultable. Es el número máximo de puestos disponibles para el examen, que se calcula como la suma de las capacidades de todas las aulas asignadas al examen.
- **Porcentaje de asistentes**, de tipo Double, consultable. Se calcula como la relación entre el número de asistentes y la capacidad máxima del examen en tanto por ciento.

Constructores:

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

Restricciones:

- El número de asistentes debe ser mayor que 0.
- La duración debe ser como mínimo de una hora.

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

Criterio de igualdad: dos exámenes son iguales si lo son sus fechas y horas, sus cursos y sus asignaturas.

Criterio de ordenación: los exámenes se ordenan por fecha y hora, a igualdad de fecha y hora por curso, y a igualdad de curso por asignatura.

Otras operaciones:

- *Boolean usaAula(String nombreAula):* devuelve *true* si el examen utiliza el aula con el nombre dado como parámetro.

Ejercicio 3: Factoría (1 pto)

En la clase FactoriaExámenes, que se le da parcialmente implementada, implemente el método:

- *Examen parsearExamen(String lineaCSV):* crea un objeto de tipo Examen a partir de una cadena de caracteres. La cadena de caracteres debe tener el mismo formato que las líneas del fichero CSV.

Ejercicio 4: Tratamientos secuenciales (7 ptos)

El tipo CalendarioExámenes tiene la siguiente descripción:

Propiedades:

- **Exámenes:** lista de exámenes, de tipo List<Examen>, consultable.
- **Número de exámenes:** número de exámenes que hay en la lista, de tipo Integer, consultable.

Constructores:

- C1: recibe un parámetro de tipo Stream<Examen>.

Representación como cadena: una cadena con todos los exámenes, separados por saltos de línea.



Implemente el tipo `CalendarioExámenes` en el paquete `fp.examenes` y añada también los siguientes tratamientos secuenciales al tipo `CalendarioExámenes`. Debe resolver todos los métodos **mediante streams**, salvo que se le indique expresamente que debe utilizar bucles:

1. `Map<String, Set<Examen>> getExamenesPorAula()`: obtiene un `Map` que relaciona cada nombre de aula con un conjunto de los exámenes que utilizan esa aula. **Implemente este método con bucles.** (1 pto)
2. `Examen getExamenMayorPorcentajeAsistentes(LocalTime t, String nombreAula)`: dadas una hora y un nombre de aula, obtiene el examen con mayor porcentaje de asistentes de entre todos los exámenes cuya hora de comienzo sea posterior esa hora y se realicen en esa aula. Si no se puede obtener, devolverá un valor `null`. (1 pto)
3. `SortedSet<String> getAulasExamenesTipo(TipoExamen tipo)`: dado un tipo de examen, obtiene un conjunto ordenado con los nombres de las aulas utilizadas en los exámenes de un tipo, sin repetir ninguno. (1 pto)
4. `String getAulaMasOcupada(LocalDate fecha)`: dada una fecha, obtiene el nombre del aula con mayor ocupación en esa fecha. Si no se puede obtener, lanzará la excepción `NoSuchElementException`. Nota: la ocupación del aula es la suma de las duraciones de los exámenes que se realizan en esa aula. Puede utilizar el método `getExamenesPorAula`. (2 ptos)
5. `List<LocalDate> getFechasConMasAulasDe(Integer umbral)`: dado un valor umbral que representa un número de aulas, obtiene una lista con las fechas en las que se han utilizado más aulas de las indicadas por ese valor umbral, en orden creciente de fechas. (2 ptos)

Escriba una clase `TestCalendarioExamenes` en el paquete `fp.examenes.test`. En el test se leerán los datos del fichero CSV y se probarán todos los tratamientos secuenciales, definiendo un método de test por cada tratamiento a probar. No se obtendrá la puntuación máxima del ejercicio si no se realiza el test y este ejecuta.

Los resultados esperados para el dataset proporcionado, con los valores indicados en los tests, son:

EJERCICIO 4.1=====

Exámenes por aula (por motivos de espacio, solo se muestran las asignaturas):

Aula F1.31: [PRG, SEG, HUM, M1]

Aula F1.30: [M2, IA, IA, IPO, HUM, M1]

Aula A1.10: [M2, M1, FP, M2, IA, IA, PRG, M2, IPO, MA, HUM, HUM]

Aula I2.33: [M1, SEG, IA, IA, SEG, HUM, M2, IPO, MA, FI, HUM, M1]

Aula I2.35: [M2, M1, PRG, FI, M2, SEG, SEG, PRG, MA, HUM, M1]

Aula I2.31: [M2, FP, M2, IA, IA, M2, IPO, MA, HUM, M1]

EJERCICIO 4.2=====

Examen con mayor porcentaje de asistentes realizado en el aula F1.30 y con hora de comienzo posterior a las 08:30: Examen [asignatura=IA, curso=3, fechaHora=2024-05-25T16:00, duracion=PT1H26M, tipo=TEORICO, asistentes=91, inscripcion=false, aulas=[Aula[nombre=A1.10, capacidad=50], Aula[nombre=I2.31, capacidad=18], Aula[nombre=F1.30, capacidad=24], Aula[nombre=I2.33, capacidad=22]]]

Examen con mayor porcentaje de asistentes realizado en el aula I2.31 y con hora de comienzo posterior a las 15:30: Examen [asignatura=M2, curso=2, fechaHora=2024-05-25T16:30, duracion=PT1H28M, tipo=PRACTICO, asistentes=88, inscripcion=false, aulas=[Aula[nombre=A1.10, capacidad=50], Aula[nombre=I2.31, capacidad=18], Aula[nombre=I2.33, capacidad=22]]]



FUNDAMENTOS DE PROGRAMACIÓN. Curso 2023/24

SEGUNDO EXAMEN PARCIAL. 29 de mayo de 2024

EJERCICIO 4.3=====

Aulas utilizadas en exámenes de tipo PRACTICO: [A1.10, F1.30, F1.31, I2.31, I2.33, I2.35]

Aulas utilizadas en exámenes de tipo TEORICO: [A1.10, F1.30, F1.31, I2.31, I2.33, I2.35]

EJERCICIO 4.4=====

Aula con mayor ocupación en la fecha 2024-05-25: I2.33

Aula con mayor ocupación en la fecha 2024-06-07: F1.31

EJERCICIO 4.5=====

Fechas con más de 5 aulas: [2024-05-25, 2024-06-07, 2024-06-19]

Fechas con más de 8 aulas: [2024-06-19]