



OBJETIVOS

- Implementación de relaciones entre objetos mediante el tipo Map.
- Construcción y uso de objetos de tipo Map y SortedMap.

NUEVA IMPLEMENTACIÓN DEL TIPO PROFESOR

Como recordará, un profesor tiene como propiedades las asignaturas que imparte y los créditos que tiene asignados a cada una de ellas. Estas propiedades se almacenan por separado, y las operaciones que las manipulan deben garantizar que ambas se mantienen sincronizadas.

En este boletín vamos a realizar una nueva implementación de estas propiedades, utilizando un Map. Esta implementación es más apropiada que la anterior ya que al tener toda la información en un único atributo en lugar de dos, resulta más fácil su tratamiento.

Para realizar esta nueva implementación, siga los pasos que se indican:

1. Haga una copia de la clase `ProfesorImpl` en el mismo paquete, y llámela `ProfesorImpl2`¹.
2. Realice las siguientes modificaciones en el código de la clase `ProfesorImpl2`:
 - a. Elimine los atributos de tipo `List<Asignatura>` y `List<Double>` que almacenan las asignaturas y los créditos impartidos en ellas, respectivamente.
 - b. Añada un nuevo atributo de tipo `Map<Asignatura, Double>`.
 - c. Modifique los constructores para inicializar el atributo de tipo Map a vacío.
 - d. Modifique todos los métodos donde se utilizan los atributos eliminados para que pasen a utilizar el nuevo atributo de tipo Map: `getAsignaturas`, `getCreditos`, `imparteAsignatura`, `eliminaAsignatura` y `dedicacionAsignatura`.

NUEVAS PROPIEDADES DE TIPO MAP Y SORTEDMAP

Vamos a añadir a algunos tipos nuevas operaciones que devuelvan objetos de tipo Map y SortedMap. Estas operaciones permitirán implementar relaciones entre objetos de distintos tipos: despachos que ocupan los profesores, profesores que imparten una asignatura, etc. Para ello, añada las siguientes operaciones a los tipos que se indican:

Tipo Centro:

- `SortedMap<String, Despacho> getDespachosPorProfesor()`. Devuelve un `SortedMap<String, Despacho>` que hace corresponder a cada profesor con el despacho que ocupa en el centro. Las claves son las representaciones como cadena de los profesores. Para construir el Map, comience por recorrer los despachos del centro y, para cada uno, recorra a su vez los profesores que lo ocupan y añada al Map la pareja formada por el profesor y el despacho.

¹ Antes de hacerlo, asegúrese de que su clase `ProfesorImpl` está bien construida y pasa todos los test.



Tipo **Departamento**:

- `SortedMap<Asignatura, SortedSet<Profesor>> getProfesoresPorAsignatura()`. Devuelve un `SortedMap<Asignatura, SortedSet<Profesor>>` que hace corresponder a cada asignatura con el conjunto de profesores que la imparten. Para construir el `SortedMap`, comience por recorrer los profesores del departamento y, para cada uno, recorra a su vez las asignaturas que imparte y añada al `SortedMap` la pareja formada por la asignatura y el profesor. Antes de hacerlo, tenga en cuenta si se trata de una nueva asignatura, en cuyo caso deberá crear antes el conjunto de profesores de la asignatura.
- `SortedMap<String, SortedSet<Tutoria>> getTutoriasPorProfesor()`. Devuelve un `SortedMap<String, SortedSet<Tutoria>>` que hace corresponder a cada profesor con el conjunto de tutorías que tiene. Las claves son las representaciones como cadena de los profesores. Para construir el `SortedMap`, comience por recorrer los profesores del departamento y, para cada uno, añada al `SortedMap` la pareja formada por el profesor y su conjunto de tutorías.

Tipo **Grado**:

- `SortedMap<Asignatura, Double> getCreditosPorAsignatura()`. Devuelve un `SortedMap<Asignatura, Double>` que hace corresponder a cada asignatura con su número de créditos. Para construir el `Map`, comience por recorrer todas las asignaturas del centro, tanto obligatorias como optativas, y, para cada una, añada al `SortedMap` la pareja formada por la asignatura y su número de créditos.

Tipo **Alumno**:

- `SortedMap<Asignatura, Calificacion> getCalificacionPorAsignatura()`. Devuelve un `SortedMap<Asignatura, Calificacion>` que hace corresponder a cada asignatura que aparece en el expediente del alumno con la calificación máxima obtenida en ella. Para construir el `SortedMap`, comience por recorrer las notas del expediente del alumno y, para cada una, añada al `SortedMap` la pareja formada por la asignatura y su calificación. Antes de hacerlo, compruebe si el `SortedMap` ya contiene una calificación para dicha asignatura, en cuyo caso deberá sustituirla por la nueva si ésta supera a la existente.

TEST

Añada casos de prueba para los nuevos métodos en las clases de test de los respectivos tipos. Para la clase `ProfesorImp12`, reutilice el test de la clase `ProfesorImp1` cambiando la llamada al constructor.