



OBJETIVOS

- Implementación de métodos de factoría.
- Implementación de propiedades poblacionales.

MÉTODOS DE FACTORÍA

Vamos a añadir varios métodos de factoría a nuestros tipos. Lo haremos en la clase de utilidad `Grados.java` proporcionada como material de apoyo en el boletín anterior. Escriba en esta clase los métodos de factoría que se indican para cada tipo. Nombre los métodos de la forma `createT`, siendo `T` el tipo correspondiente (por ejemplo, `createAlumno`). En los métodos creacionales a partir de un fichero, añada una 's' al final (por ejemplo, `createAlumnos`).

- **Alumno:** un método creacional por parámetros, un método creacional copia, un método creacional a partir de `String` y un método creacional a partir de un fichero.
- **Asignatura:** un método creacional por parámetros¹, un método creacional a partir de `String` y un método creacional a partir de un fichero.
- **Beca:** dos métodos creacionales por parámetros, un método creacional copia, un método creacional a partir de `String` y un método creacional a partir de un fichero.
- **Centro:** un método creacional por parámetros y un método creacional copia.
- **Departamento:** un método creacional por parámetros.
- **Despacho:** un método creacional por parámetros (que recibe nombre, capacidad y planta), un método creacional copia, un método creacional a partir de `String` y un método creacional a partir de un fichero.
- **Espacio:** un método creacional por parámetros, un método creacional copia, un método creacional a partir de `String` y un método creacional a partir de un fichero.
- **Grado:** un método creacional por parámetros.
- **Profesor:** un método creacional por parámetros² y un método creacional copia.

Para el tipo `Profesor`, los métodos creacionales deben permitir el uso de las dos implementaciones disponibles para el tipo, `ProfesorImpl1` y `ProfesorImpl2`. Para ello, añada a la factoría un método `setUserImplementacionMapProfesor` que permita elegir cuál de las dos clases se utilizará al construir objetos de tipo `Profesor`. Por defecto se utilizará `ProfesorImpl2`.

PROPIEDADES POBLACIONALES

Implemente, para todos los tipos excepto **Becario, Despacho, Expediente, Nota, Persona, Tutoría** y los tipos inmutables, las siguientes propiedades poblacionales:

- Número de objetos distintos creados.
- Conjunto con todos los objetos distintos creados. En el caso del tipo `Espacio`, este conjunto será ordenado.

¹ Utilice para este método el constructor de `AsignaturaImpl` que recibe un mayor número de parámetros.

² Utilice para este método el constructor de `ProfesorImpl` o `ProfesorImpl2` que recibe un mayor número de parámetros.

Además, implemente las siguientes propiedades poblacionales:

- **Asignatura:** códigos de todas las asignaturas creadas. La implementación de esta propiedad debe posibilitar el acceso a cualquier asignatura creada a partir de su código.
- **Beca:** número de becas creadas por cada tipo de beca.
- **Centro:** número máximo de plantas, número máximo de sótanos, número medio de plantas y número medio de sótanos. Si no se han creado centros, las dos primeras tendrán un valor `null` y las dos últimas un valor 0.
- **Espacio:** número de la planta máxima y número de la planta mínima donde se puede encontrar un espacio. Si no se han creado espacios, ambas propiedades tendrán un valor `null`. A efectos de estas propiedades y de las anteriormente descritas (número de espacios y conjunto ordenado de espacios), **tenga en cuenta que los despachos son también espacios.**
- **Grado:** número medio de asignaturas obligatorias, número medio de asignaturas optativas y número medio de asignaturas de cualquier tipo. Si no se han creado grados, las tres propiedades tendrán un valor 0.

Utilice los siguientes nombres para los métodos:

Tipo	Métodos
Alumno	<code>getNumAlumnosCreados</code> , <code>getAlumnosCreados</code>
Asignatura	<code>getNumAsignaturasCreadas</code> , <code>getAsignaturasCreadas</code> , <code>getCodigosAsignaturasCreadas</code> , <code>getAsignaturaCreada</code>
Beca	<code>getNumBecasCreadas</code> , <code>getNumBecasTipo</code> , <code>getBecasCreadas</code>
Centro	<code>getNumCentrosCreados</code> , <code>getCentrosCreados</code> , <code>getMaxPlantas</code> , <code>getMaxSotanos</code> , <code>getMediaPlantas</code> , <code>getMediaSotanos</code>
Departamento	<code>getNumDepartamentosCreados</code> , <code>getDepartamentosCreados</code>
Espacio	<code>getNumEspaciosCreados</code> , <code>getEspaciosCreados</code> , <code>getPlantaMayorEspacio</code> , <code>getPlantaMenorEspacio</code>
Grado	<code>getNumGradosCreados</code> , <code>getGradosCreados</code> , <code>getMediaAsignaturasGrados</code> , <code>getMediaAsignaturasObligatoriasGrados</code> , <code>getMediaAsignaturasOptativasGrados</code>
Profesor	<code>getNumProfesoresCreados</code> , <code>getProfesoresCreados</code> , <code>setUsarImplementacionMapProfesor</code>

TEST

Para probar los métodos creacionales, utilice las clases de test de cada tipo, sustituyendo las llamadas a los constructores por llamadas a los métodos de la factoría.

Para probar las propiedades de población, añada a cada clase de test un caso de prueba en el que construya varios objetos utilizando todos los métodos creacionales, y compruebe posteriormente los valores almacenados en cada una de las propiedades poblacionales.