

## PROGRAMACIÓN JAVA

Nivel: **Intermedio**

Puntos: **4**

Se pide crear **Filtro** e **Almacen** e **ArrayListAlmacen** no package **storage** e a clase **AlmacenArrayListTest** no package **tests**

Un **Filtro** é un obxecto capaz de indicar si resulta “aceptable” ou non un obxecto de un tipo calquera mediante o método **boolean match(obxecto)**, que retorna **true** si o **obxecto** resulta “aceptable”, ou **false** en outro caso. A definición de **Filtro** debe facer uso de tipos xenéricos (Generics) para poder ser empregado con calquera clase de obxecto.

Un **Almacen** é un obxecto capaz de xestionar o almacenamento de calquera clase de obxectos, para o que fai uso de tipos xenéricos (Generics) para a súa correcta definición. Os obxectos **Almacen** deben permitir percorrer os elementos almacenados mediante un Iterator, un for ou un forEach.

Un **Almacen** debe ter os seguintes métodos:

- **engadir** – engade ao Almacen o obxecto recibido como parámetro, retornando **true** si se engade correctamente ou **false** en outro caso. Si o obxecto xa se atopaba no **Almacen** (no sentido de que xa existe outro obxecto que é “equals” co que queremos engadir), non se engade e se retorna **false**. Si se produce un erro grave, lanza unha **Exception**.
- **cargar** – Recibe un **Filtro** e un array onde gardar a información, e retorna un array con todos os obxectos do **Almacen** que son “validos” segundo o **Filtro** recibido. Si se produce un erro grave, lanza unha **Exception**
- **cargar** – Recibe un array onde gardar a información e retorna un array con todos os elementos gardados no **Almacen**. Si se produce un erro grave, lanza unha **Exception**.
- **eliminar** – Recibe un **Filtro** e elimina do almacén todos os obxectos que son “validos” segundo o **Filtro** recibido retornando o número de elementos eliminados. Si se produce un erro grave, lanza unha **Exception**
- **eliminar** – Recibe un obxecto e o elimina do **Almacen**. Retorna 1 si elimina o obxecto ou 0 si non existe. Si se produce un erro grave, lanza unha **Exception**
- **modificar** – Recibe un obxecto e substitúe a información do obxecto existente no **Almacen** pola do obxecto recibido. Retorna **false** si o obxecto non existe, e **true** si se modifica correctamente. Si se produce un erro grave, lanza unha **Exception**
- **gardar** – Recibe un obxecto e o engade ao **Almacen** si non existe, ou o modifica si xa existe. Retorna **true** si se garda/modifica e **false** en outro caso. Si se produce un erro grave, lanza unha **Exception**

Un Almacen debe poder ser implantado de moitos xeitos facendo uso de distintas estruturas (persistentes ou non) e servir para almacenar obxectos de distintas clases, polo que os seus métodos non son implantables. Queremos facer unha implantación completa de Almacen facendo uso dun ArrayList como estrutura para gardar os datos definindo a clase ArrayListAlmacen no package storage que será capaz de almacenar obxectos da clase asociada, polo que se deseñará facendo uso de tipos xenéricos (Generics)

Para comprobar a corrección de **ArrayListAlmacen**:

- Creamos no package **library** a clase **Publicacion** que ten como mínimo os atributos privados e non modificables de tipo String **isbn** e **título** cos **getter** correspondentes. As **Publicacion** son **Comparable** por título e consideramos dúas **Publicacion** iguais si os seus **isbn** coinciden. **Publicacion** ten como representación String **isbn=<valor isbn>, título=<valor título>**
- Definimos no package **test** a clase **AlmacenArrayListTest** que levará a cabo as seguintes accións no seu método **main()**:
  1. Creamos un obxecto **ArrayListAlmacen** capaz de almacenar obxectos de tipo **Publicacion**
  2. Engadimos ao almacén os seguintes obxectos **Publicacion**:
    - **Publicacion**("978-1-4842-4827-0", "Building Microservices Applications")
    - **Publicacion**("84-7897-429-6", "Java 2")
    - **Publicacion**("84-9103-064-5", "Antoni Gaudí. Biografía Ilustrada")
    - **Publicacion**("84-2067-420-9", "El Guardián Entre el Centeno")
  3. Listamos todos os elementos do almacén facendo uso dun **Iterator** ou mediante un **for / forEach**
  4. Listamos todas as publicacións que teñen a palabra “Gaudí” no seu título (ver método **contains()** da clase **String**)
  5. Modificamos o libro "84-7897-429-6" poñendo “Aprendendo Java” como Título e listamos todos os elementos do almacén ordeados por título.

O resultado da execución debe ser o seguinte:

```
Creando AlmacenArrayList e almacenando publicacións

Listado de Publicacións no almacen:
isbn=978-1-4842-4827-0, titulo=Building Microservices Applications
isbn=84-7897-429-6, titulo=Java 2
isbn=84-9103-064-5, titulo=Antoni Gaudí. Biografía Ilustrada
isbn=84-2067-420-9, titulo=El Guardián Entre el Centeno

Listado de Publicacións con 'Gaudí' no título:
isbn=84-9103-064-5, titulo=Antoni Gaudí. Biografía Ilustrada

Modificación de 'Java 2' por 'Aprendendo Java' e listado dos libros ordeados por título:
isbn=84-9103-064-5, titulo=Antoni Gaudí. Biografía Ilustrada
isbn=84-7897-429-6, titulo=Aprendendo Java
isbn=978-1-4842-4827-0, titulo=Building Microservices Applications
isbn=84-2067-420-9, titulo=El Guardián Entre el Centeno
```

#### Criterios de Avaliación

- |   |      |
|---|------|
| • <b>Filtro</b> está correctamente definido               | 0.25 |
| • <b>Almacen</b> está correctamente definido              | 0.75 |
| • A clase AlmacenArrayList está correctamente definida    | 0.25 |
| • <b>engadir, modificar, gardar</b> son correctos         | 0.50 |
| • os métodos <b>cargar</b> son correctos                  | 0.50 |
| • os métodos <b>eliminar</b> son correctos                | 0.50 |
| • <b>AlmacenArrayList</b> retorna un Iterator correcto    | 0.25 |
| • Publicacion está correctamente e completamente definida | 0.50 |
| • A aplicación AlmacenArrayListTest e correcta            | 0.50 |