

Programación Orientada a Objetos. Práctica 2.2

Tema 2. Clases, Objetos y Arrays

Características de la práctica

En esta práctica el alumno aprenderá a utilizar la composición de objetos mediante la utilización de arrays de objetos. Además, aprenderá las diferencias entre variables de instancia y variables de clase (también pueden ser constantes), así como la conveniencia de definir métodos privados, y de que los métodos de clase sólo pueden acceder a las variables de clase, ya que no pueden acceder a las variables de instancia. También practicará las operaciones básicas sobre arrays *incompletos* (*añadir*, *buscar*, *eliminar*).

Ejercicio 1. (proyecto prLibreria)

El diagrama de clases UML

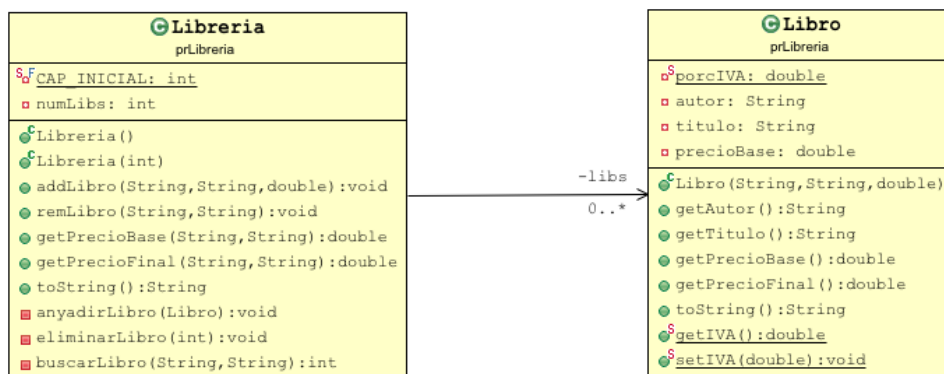


Figura 1: Diagrama de clases UML

Nota: se pueden añadir a las siguientes clases los métodos **privados** que se consideren necesarios.

Nota: para facilitar el desarrollo de clases derivadas, es posible que algunos métodos actualmente *privados* deban ser modificados a **protegidos** en la siguiente práctica.

La clase Libro

La clase **Libro** (del paquete **prLibreria**) contiene información sobre un determinado libro, tal como el nombre del autor, el título, y el precio base. Además, también posee información sobre el porcentaje de IVA que se aplica para calcular su precio final. Nótese que el porcentaje de IVA a aplicar es el mismo y es compartido por todos los libros, siendo su valor inicial el 10.0 %.

- **Libro(String,String,double)**

Construye un objeto **Libro**. Recibe como parámetros, en el siguiente orden, el nombre del autor, el título, y el precio base del libro.

- **getAutor():String**
getTitulo():String
getPrecioBase():double

Devuelven los valores correspondientes almacenados en el objeto.

- **getPrecioFinal():double**

Devuelve el precio final del libro, incluyendo el IVA, según la siguiente ecuación.

$$PF = PB + PB \times IVA \div 100$$

- `toString(): String // @Redefinición`

Devuelve la representación textual del objeto, según el formato del siguiente ejemplo:

(Isaac Asimov; La Fundación; 7.3; 10.0%; 8.03)

- `getIVA():double // @MétodoDeClase`

Devuelve el porcentaje del IVA asociado a la clase **Libro**.

- `setIVA(double):void // @MétodoDeClase`

Actualiza el valor del porcentaje del IVA asociado a la clase **Libro** al valor recibido como parámetro.

La clase Libreria

La clase **Libreria** (del paquete `prLibreria`) almacena múltiples instancias de la clase **Libro** en un array, así como el número total de libros que contiene almacenados. Además, también contiene una constante de clase que especifica la capacidad inicial por defecto del array (8).

Nota 1: todos los libros almacenados en la librería se encuentran *referenciados* desde las primeras posiciones del array, de forma consecutiva y sin huecos libres, y el resto de elementos del array, que no referencian a ningún libro, contienen el valor `null` (véase fig.2).

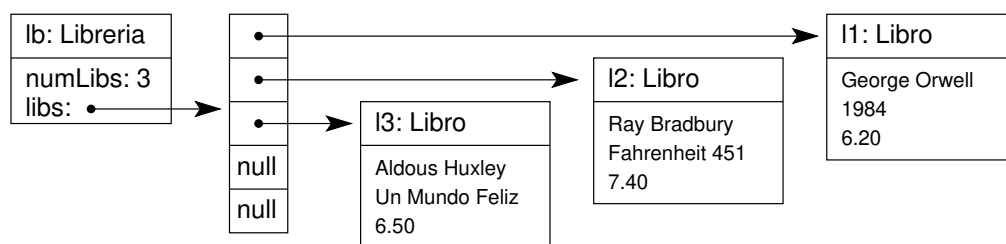


Figura 2: Esquema de la Librería

Nota 2: las comparaciones que se realicen tanto del nombre del autor como del título del libro se deberán realizar sin diferenciar mayúsculas de minúsculas.

Nota 3: se recomienda la definición de métodos privados que simplifiquen y permitan modularizar la solución de métodos complejos.

- `Libreria()`

Construye un objeto **Libreria** vacío (sin libros) con un array con una capacidad inicial de tamaño 8.

- `Libreria(int)`

Construye un objeto **Libreria** vacío (sin libros) con un array con una capacidad inicial del tamaño recibido como parámetro.

- `addLibro(String,String,double): void`

Crea un nuevo objeto **Libro** con el nombre del autor, el título, y el precio base recibidos como parámetros, e invoca al método privado `anyadirLibro` para añadirlo a la librería.

Invoca a: `anyadirLibro`.

- `remLibro(String,String): void`

Si existe el libro correspondiente al autor y título recibidos como parámetros, entonces elimina el libro de la librería invocando al método privado `eliminarLibro`. Si no hay ningún libro con ese autor y título, entonces no se hace nada.

Invoca a: `buscarLibro` y `eliminarLibro`.

- `getPrecioBase(String,String): double`

Devuelve el precio base del libro correspondiente al autor y título recibidos como parámetros. Si el libro no existe en la librería, entonces devuelve cero.

Invoca a: `buscarLibro`.

- `getPrecioFinal(String,String): double`

Devuelve el precio final del libro correspondiente al autor y título especificados. Si el libro no existe en la librería, entonces devuelve cero.

Invoca a: `buscarLibro`.

- `toString(): String // @Redefinición`

Devuelve la representación textual del objeto, según el formato del siguiente ejemplo (sin considerar los saltos de línea):

```
[(George Orwell; 1984; 6.2; 10.0%; 6.82),  
 (Philip K. Dick; ¿Sueñan los androides con ovejas eléctricas?; 3.5; 10.0%; 3.85),  
 (Isaac Asimov; Fundación e Imperio; 9.4; 10.0%; 10.34),  
 (Ray Bradbury; Fahrenheit 451; 7.4; 10.0%; 8.14),  
 (Aldous Huxley; Un Mundo Feliz; 6.5; 10.0%; 7.15),  
 (Isaac Asimov; La Fundación; 7.3; 10.0%; 8.03),  
 (William Gibson; Neuromante; 8.3; 10.0%; 9.13),  
 (Isaac Asimov; Segunda Fundación; 8.1; 10.0%; 8.91),  
 (Isaac Newton; Arithmetica Universalis; 10.5; 10.0%; 11.55)]
```

- `anyadirLibro(Libro):void // @Privado`

Si ya existe un libro de ese mismo autor, con el mismo título, entonces se reemplaza el libro anterior por el nuevo. En otro caso, añade el nuevo libro a la librería, en la primera posición disponible, según la *nota-1* especificada anteriormente, considerando que si el array está lleno se debe duplicar su capacidad. Así mismo, se debe actualizar adecuadamente el valor de la cuenta del número de libros.

Invoca a: `buscarLibro`, `Arrays.copyOf`

- `eliminarLibro(int):void // @Privado`

Elimina el libro que se encuentra en la posición recibida como parámetro. Para ello, moverá el último libro a la posición del elemento a eliminar, según la *nota-1* especificada anteriormente. Así mismo, se debe actualizar adecuadamente el valor de la cuenta del número de libros.

- `buscarLibro(String,String):int // @Privado`

Busca en la librería la posición donde se **encuentra** un libro del autor recibido como primer parámetro, con el título recibido como segundo parámetro. Si lo encuentra, entonces devuelve el índice del array donde se encuentra el libro buscado. En caso de no encontrarlo, devuelve -1.

La aplicación PruebaLibreria

Desarrolle una aplicación (en el paquete anónimo) que permita realizar una prueba de las clases anteriores. Así, deberá añadir a la librería los siguientes libros:

```
("george orwell", "1984", 8.20)  
("Philip K. Dick", "¿Sueñan los androides con ovejas eléctricas?", 3.50)  
("Isaac Asimov", "Fundación e Imperio", 9.40)  
("Ray Bradbury", "Fahrenheit 451", 7.40)  
("Aldous Huxley", "Un Mundo Feliz", 6.50)  
("Isaac Asimov", "La Fundación", 7.30)  
("William Gibson", "Neuromante", 8.30)
```

```

("Isaac Asimov", "Segunda Fundación", 8.10)
("Isaac Newton", "arithmetica universalis", 7.50)
("George Orwell", "1984", 6.20)
("Isaac Newton", "Arithmetica Universalis", 10.50)

```

De tal forma que al mostrar la representación textual de la librería mostrará (sin considerar los saltos de línea):

```

[(George Orwell; 1984; 6.2; 10.0%; 6.82),
 (Philip K. Dick; ¿Sueñan los androides con ovejas eléctricas?; 3.5; 10.0%; 3.85),
 (Isaac Asimov; Fundación e Imperio; 9.4; 10.0%; 10.34),
 (Ray Bradbury; Fahrenheit 451; 7.4; 10.0%; 8.14),
 (Aldous Huxley; Un Mundo Feliz; 6.5; 10.0%; 7.15),
 (Isaac Asimov; La Fundación; 7.3; 10.0%; 8.03),
 (William Gibson; Neuromante; 8.3; 10.0%; 9.13),
 (Isaac Asimov; Segunda Fundación; 8.1; 10.0%; 8.91),
 (Isaac Newton; Arithmetica Universalis; 10.5; 10.0%; 11.55)]

```

A continuación se eliminarán los siguientes libros:

```

("George Orwell", "1984")
("Aldous Huxley", "Un Mundo Feliz")
("Isaac Newton", "Arithmetica Universalis")
("James Gosling", "The Java Language Specification")

```

De tal forma que al mostrar la representación textual de la librería mostrará (sin considerar los saltos de línea):

```

[(William Gibson; Neuromante; 8.3; 10.0%; 9.13),
 (Philip K. Dick; ¿Sueñan los androides con ovejas eléctricas?; 3.5; 10.0%; 3.85),
 (Isaac Asimov; Fundación e Imperio; 9.4; 10.0%; 10.34),
 (Ray Bradbury; Fahrenheit 451; 7.4; 10.0%; 8.14),
 (Isaac Asimov; Segunda Fundación; 8.1; 10.0%; 8.91),
 (Isaac Asimov; La Fundación; 7.3; 10.0%; 8.03)]

```

Finalmente se mostrará el precio final de los siguientes libros:

```

("George Orwell", "1984")
("Philip K. Dick", "¿Sueñan los androides con ovejas eléctricas?")
("isaac asimov", "fundación e imperio")
("Ray Bradbury", "Fahrenheit 451")
("Aldous Huxley", "Un Mundo Feliz")
("Isaac Asimov", "La Fundación")
("william gibson", "neuromante")
("Isaac Asimov", "Segunda Fundación")
("Isaac Newton", "Arithmetica Universalis")

```

Mostrando en consola el siguiente resultado:

```

PrecioFinal(George Orwell, 1984): 0.0
PrecioFinal(Philip K. Dick, ¿Sueñan los androides con ovejas eléctricas?): 3.85
PrecioFinal(isaac asimov, fundación e imperio): 10.34
PrecioFinal(Ray Bradbury, Fahrenheit 451): 8.14
PrecioFinal(Aldous Huxley, Un Mundo Feliz): 0.0
PrecioFinal(Isaac Asimov, La Fundación): 8.03
PrecioFinal(william gibson, neuromante): 9.13
PrecioFinal(Isaac Asimov, Segunda Fundación): 8.91
PrecioFinal(Isaac Newton, Arithmetica Universalis): 0.0

```