

Programación Orientada a Objetos. Práctica 2.2

Tema 2. Clases, Objetos y Listas

Características de la práctica

En esta práctica el alumno aprenderá a utilizar la composición de objetos mediante la utilización de listas 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 listas (*añadir*, *buscar*, *eliminar*).

Ejercicio 1. (proyecto prLibreria, paquete libreria)

El diagrama de clases UML

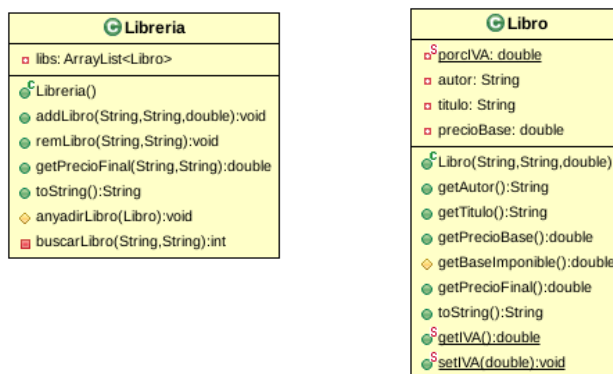


Figura 1: Diagrama de clases UML

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

Nota: para facilitar el desarrollo de clases derivadas en la siguiente práctica, algunos métodos se definirán con nivel de acceso **protegido**.

La clase Libro

La clase **Libro** (del paquete `libreria`) 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.

- `getBaseImponible(): double` // @Protegido

Es un método **protegido** que devuelve la base imponible del libro, que en este caso, es igual al precio base del libro.

- `getPrecioFinal(): double`

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

$$\text{precioFinal} = \text{baseImponible} + \text{baseImponible} \times \text{IVA} \div 100$$

Nota: este método invoca a `getBaseImponible` para obtener la base imponible sobre la que aplicar el IVA y calcular el precio final. Téngase en cuenta que el método `getBaseImponible` podría ser redefinido por subclases.

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

Devuelve la representación textual del objeto, según el formato del siguiente ejemplo (*autor; título; precioBase; porcentajeIva; precioFinal*):

(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 `libreria`) almacena múltiples instancias (objetos) de la clase `Libro` en una lista.

Nota 1: las comparaciones que se realicen tanto del nombre del autor como del título del libro se deberán realizar sin diferenciar mayúsculas de minúsculas.

Nota 2: 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 una lista de libros vacía.

- `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 protegido `anyadirLibro` para añadirlo a la lista de libros de la librería.

- `remLibro(String,String): void`

Si existe el libro correspondiente al autor y título recibidos como parámetros, entonces elimina el libro de la lista de libros de la librería. Si no hay ningún libro con ese autor y título, entonces lanza la excepción `RuntimeException` con el mensaje "Libro no encontrado (*autor, título*)".

Nota: este método invoca a `buscarLibro` para obtener el índice en la lista donde se encuentra el libro.

- `getPrecioFinal(String,String): double`

Devuelve el precio final del libro correspondiente al autor y título recibidos, en ese orden, como parámetros. Si no hay ningún libro con ese autor y título, entonces lanza la excepción `RuntimeException` con el mensaje "Libro no encontrado (*autor, título*)".

Nota: este método invoca a `buscarLibro` para obtener el índice en la lista donde se encuentra el libro.

- `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` // *@Protegido*

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 al final de la lista de libros de la librería.

Nota: este método invoca a `buscarLibro` para obtener el índice en la lista donde se encuentra el libro.

- `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 de la lista 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")
```

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

```
[(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; 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)]
```

Finalmente se mostrará el precio final de los siguientes libros:

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

Mostrando en consola el siguiente resultado:

```
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(Isaac Asimov, La Fundación): 8.03
PrecioFinal(william gibson, neuromante): 9.13
PrecioFinal(Isaac Asimov, Segunda Fundación): 8.91
Exception in thread "main" java.lang.RuntimeException: Libro no encontrado (Isaac Newton, Arithmetica
    at libreria.Libreria.getPrecioFinal(Libreria.java:??)
    at PruebaLibreria.main(PruebaLibreria.java:??)
```