Object-Oriented Programming. Exercise 2.2

Unit 2. Classes, Objects and Arrays

Description of the exercise

In this exercise we will learn to use the composition of objects by using object arrays. In addition, we will learn the differences between instance variables and class variables (they can also be constants), as well as the convenience of defining private methods, and that class methods can only access class variables, since they cannot access to instance variables. We will also practice basic operations on incomplete *arrays* (*add*, *search*, *delete*).

Exercise 1. (project prBookStore)

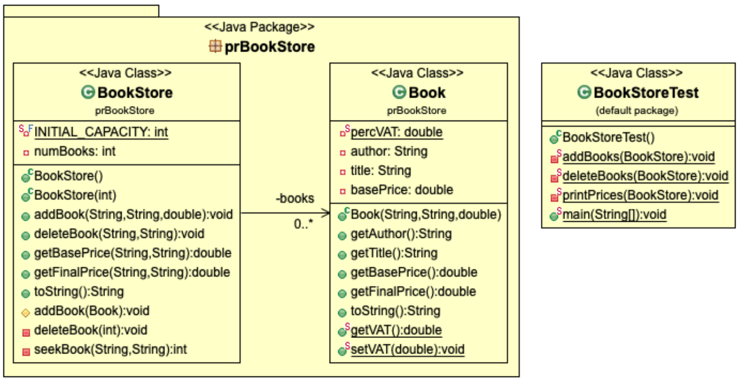


Figura 1: UML class diagram

Note 1: you can add as many **private** methods as you consider necessary.

Note 2: To facilitate the development of derived classes, it is possible that some currently *private* methods should be made **protected** in the following exercise.

The `Book` class

The `Book` class (from the `prBookStore` package) contains information about a particular book, such as the author's name, title, and base price. In addition, it also has information on the VAT percentage that is applied to calculate its final price. Note that the VAT percentage to be applied is the same and is shared by all books, with its initial value being 10.0 %.

- `Book(String, String, double)`

  Builds a `Book` object. It receives as parameters, in the following order, the author's name, its title, and the base price of the book.

- `getAuthor(): String`
- `getTitle(): String`
- `getBasePrice(): double`

  They return the corresponding values stored in the object.

- `FinalPrice(): double`

  Returns the final price of the book, including VAT, according to the following equation.

  $$FP = BP + BP \times VAT \div 100$$

- `toString(): String  // [@Redefinition]`

  Returns the textual representation of the object, according to the format of the following example:

  ```
  (Isaac Asimov; La Fundación; 7.30; 10.0%; 8.03)
  ```

- `getVAT(): double  // [@ClassMethod]`

  Returns the VAT percentage associated with the `Book` class.

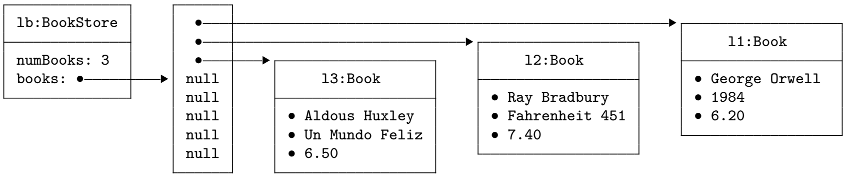
- `setVAT(double): void  // [@ClassMethod]`

  Updates the value of the VAT percentage associated with the `Book` class to the value received as a parameter.

The `BookStore` class

The `BookStore` class (from the `prBookStore` package) stores multiple instances of the `Book` class in an array, as well as the total number of books it contains. In addition, it also contains a class constant that specifies the initial default capacity of the array (8).

Note 3: All books stored in the library are *referenced* from the first positions of the array, consecutively and without free gaps, and the rest of the array elements, which do not reference any books, contain the value `null`.



Nota 4: To compare names and titles should not distinguish letter cases (*uppercase* or *lowercase*).

Nota 5: The definition of private methods that simplify and allow to modularize the solution of complex methods is recommended.

- `BookStore()`

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

- `BookStore(int)`

  Constructs an empty `BookStore` object (without books) with an array with an initial capacity of the size received as a parameter.

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

  Creates a new `Book` object with the author's name, title, and base price received as parameters, and invoke the private method `addBook` to add it to the library.

  Invokes: `addLibro`.

- `deleteBook(String, String): void`

  If the book corresponding to the author and title received as parameters exists, then the method deletes the book from the BookStore by invoking the private method `deleteBook`. If there is no book with such author and title then nothing is done.

  Invokes: `seekBook` y `deleteBook`.

- `getBasePrice(String, String): double`

Returns the base price of the book corresponding to the author and title received as parameters. If the book does not exist in the BookStore, then it returns zero.

Invokes: `seekBook`.

- `FinalPrice(String, String): double`

Returns the final price of the book corresponding to the specified author and title. If the book does not exist in the BookStore, then it returns zero.

Invokes: `seekBook`.

- `toString(): String // [@Redefinition]`

Returns the textual representation of the object, according to the format of the following example (without the line breaks):

```
[(George Orwell; 1984; 6.20; 10.0%; 6.82),
 (Philip K. Dick; ¿Sueñan los androides con ovejas eléctricas?; 3.50; 10.0%; 3.85),
 (Isaac Asimov; Fundación e Imperio; 9.40; 10.0%; 10.34),
 (Ray Bradbury; Fahrenheit 451; 7.40; 10.0%; 8.14),
 (Aldous Huxley; Un Mundo Feliz; 6.50; 10.0%; 7.15),
 (Isaac Asimov; La Fundación; 7.30; 10.0%; 8.03),
 (William Gibson; Neuromante; 8.30; 10.0%; 9.13),
 (Isaac Asimov; Segunda Fundación; 8.10; 10.0%; 8.91),
 (Isaac Newton; Arithmetica Universalis; 10.50; 10.0%; 11.55)]
```

- `addBook(Libro): void // [@Private]`

If a book by that same author already exists, with the same title, then the previous book is replaced by the new one. In another case, it adds the new book to the book store, in the first available position, according to *Note 1*, considering that if the array is full its capacity must be doubled. Likewise, number of books must be properly updated.

Invokes: `seekBook`, `Arrays.copyOf`

- `deleteBook(int): void // [@Private]`

Deletes the book that is in the position received as a parameter. To do this, it will move the last book to the position of the item to be deleted, according to *Note 1*. Likewise, the number of books must be properly updated.

- `seekBook(String, String): int // [@Private]`

Searches the book store for the position where a book with author and title as those received as parameters is **found**. If found, then it returns the index of the array where the searched book is located. If the book is not found, it returns `-1`.

The main class `BookStoreTest`

Develop an application (in the *anonymous package*) that allows you to test the previous classes. Thus, you must add the following books to the book store:

```
("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)
```

Then, when printing the string representation of the book store you will get in the terminal (without carry returns):

```
[(George Orwell; 1984; 6.20; 10.0%; 6.82),
 (Philip K. Dick; ¿Sueñan los androides con ovejas eléctricas?; 3.50; 10.0%; 3.85),
 (Isaac Asimov; Fundación e Imperio; 9.40; 10.0%; 10.34),
 (Ray Bradbury; Fahrenheit 451; 7.40; 10.0%; 8.14),
 (Aldous Huxley; Un Mundo Feliz; 6.50; 10.0%; 7.15),
 (Isaac Asimov; La Fundación; 7.30; 10.0%; 8.03),
 (William Gibson; Neuromante; 8.30; 10.0%; 9.13),
 (Isaac Asimov; Segunda Fundación; 8.10; 10.0%; 8.91),
 (Isaac Newton; Arithmetica Universalis; 10.50; 10.0%; 11.55)]
```

Then, delete the following books:

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

Then, when printing the string representation of the book store you will get in the terminal (without carry returns):

```
[(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)]
```

Finally, you'll show the final price of the following books:

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