

Programación II

Ejercicio: Club del Lector

Dpto. LSIIS. Unidad de Programación

Enunciado

Este ejercicio se hace de forma individual. El alumno debe entregar los ficheros Libro.java y Lector.java.

Se pretende realizar una implementación simplificada de un sistema gestor para un club de lectura.

El club consta de un conjunto de libros y de socios 'Lectores' que cogen prestados libros, los leen y los devuelven. En este ejercicio nos vamos a centrar sólo en las clases Libro y Lector. A continuación, se indica qué atributos y servicios tiene cada uno.

Diagrama de clases

En la Figura 1 se muestra el diagrama de clases correspondientes a las clases del paquete **ee1**. En este diagrama no se incluye la clase auxiliar Fecha. En el diagrama tampoco se muestran los constructores, ni los gets o sets (que pudieran ser necesarios). El alumno también tendrá que decidir si son necesarios o no más atributos para poder implementar la solución.

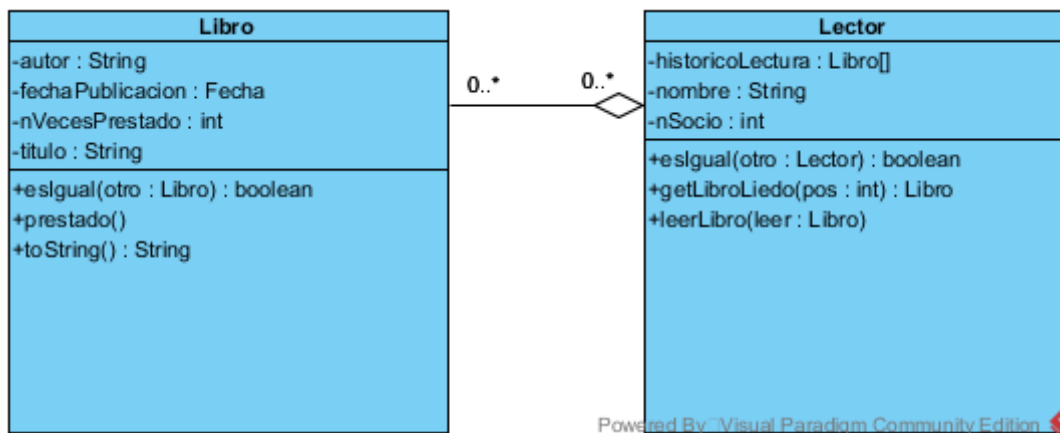


Figura 1 Diagrama de clases

Clase Libro

Esta clase representa un libro y se encuentra en el paquete ee1. La información que representa (estado) un libro es:

- **titulo:** de tipo String que contendrá el título del libro.
- **autor:** de tipo String que contendrá el nombre y apellidos del autor (para este problema se asume que cada libro sólo tiene un autor).
- **fechaPublicacion:** atributo de tipo fecha, que contendrá la fecha en que fue publicado.
- **nVecesPrestado:** número entero que lleva la cuenta de las veces que ha sido prestado.

Los servicios que ofrecen las instancias de la clase libro son:

- **Constructor** inicializa los atributos necesarios. El atributo fechaPublicacion se debe inicializar con una copia del parámetro de entrada para prevenir posibles efectos colaterales.
- Se ofrecerán *getters* de todos los atributos.
- Se ofrecerá un método **prestado** que incrementará en uno el número de veces que el libro ha sido prestado.
- Un método **esIgual** que indicará si el libro es igual a otro pasado como parámetro. Dos libros son iguales **si y sólo si coinciden el título, el autor y la fecha de publicación**.
- Un método **toString** que retorna un String con el siguiente formato: "Título, Autor, leído: nnn" donde nnn es el número de veces que el libro ha sido leído.

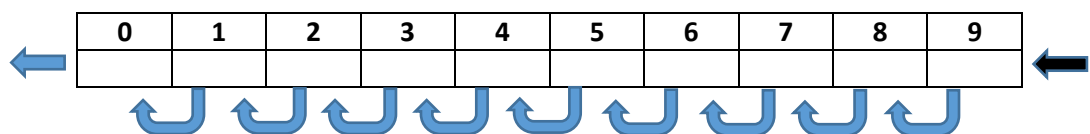
Clase Lector

Esta clase representa al lector. Esta clase se ubica en el paquete ee1. Estará formada por los siguientes atributos:

- **nSocio**: entero que representa el número de socio.
- **nombre**: contiene el nombre del socio.
- **historicoLectura**: vector (array) que mantiene un histórico de los últimos 10 libros que el lector ha leído en orden cronológico. Por tanto, en la posición cero está el libro que se leyó hace más tiempo y en la última posición ocupada el libro que se leyó más recientemente.

Los servicios que ofrecen las instancias de la clase Lector son:

- **Constructor**, inicializa los atributos necesarios. El constructor **no** recibirá información sobre el histórico de libros leídos, ya que este histórico se irá rellenando durante la vida del objeto.
- *Getters* para número de socio y nombre.
- **getLibroLeido**: Da el libro que se encuentra en el histórico en la posición que se recibe como parámetro. La posición **uno** hace referencia al primer libro que hay en el histórico. Si la posición está fuera del rango, se debe retornar **null**. Si la posición se encuentra entre **1 y el número de libros** en el histórico, entonces se retorna una referencia al libro que se encuentra en el histórico.
- **leerLibro**: Servicio que recibe una instancia de libro, lo inserta en el histórico y **le indica al libro que ha sido prestado**. Este método funciona de la siguiente forma:
 1. Si el número de libros leídos por el lector es menor que la capacidad del histórico, este libro se inserta después del último que hubiera en el histórico.
 2. Si el número de libros leídos **supera la capacidad del histórico** se descarta el libro que hizo más tiempo que se leyó (se desplazan los libros una posición hacia delante dejando un hueco al final) y en el hueco dejado al final se insertará el libro recibido como entrada.



- **esIgual**: Se considera que dos lectores son iguales cuando coincide el número de socio y los libros leídos que hay en el histórico son los mismos y están en el mismo orden.

Código de apoyo

Al alumno se le proporciona un zip que contiene los materiales necesarios para realizar el ejercicio, que deberá ser descargada (en el caso de que no se tenga aún) por el alumno. La estructura del zip es:

- src: Carpeta o directorio que contiene el código de apoyo. Esta carpeta a su vez tiene:
 - TestLectorAlum.java: Pruebas del lector implementadas con un main
 - TestLibroAlum.java: Pruebas del libro implementadas con un main
 - Fecha.jar: Librería que contiene la clase fecha. Se debe añadir al proyecto (Built Path -> Add External Libraries -> ...) para poder utilizar dicha clase.
 - ee1: Carpeta o directorio que define el paquete en el que están la clase Libro y Lector. Se proporcionan las clases Lector.java y Libro.java vacías. El alumno debe completarlas según lo especificado en el enunciado.
- docFecha: Carpeta o directorio que contiene la documentación de la librería fecha en formato javadoc (páginas html).

Consideraciones a la hora de probar el ejercicio

El alumno debe verificar que su código compila con los ficheros de prueba suministrados. Al alumno se le proporcionan dos programas de prueba uno para Libro (TestLibroAlum.java) y otro para lector (TestLectorAlum.java), estos dos programas se encuentran en el paquete por defecto. Las pruebas no son exhaustivas en el sentido de que no prueban todos los casos posibles que se le pueden presentar a la aplicación. Por tanto, no se deben modificar las pruebas que hay en estos archivos, pero sí se deben añadir nuevas pruebas después de la última que exista en cada uno de los archivos.

Se recomienda probar primero la clase Libro y cuando esta pase todas las pruebas, proceder con la clase Lector.