



Nombre del proyecto: Gestión de Biblioteca

Una biblioteca desea administrar los préstamos de sus libros.

Cada libro tiene una ficha de préstamos en donde se asientan todos los préstamos realizados del libro. Cada socio tiene una lista de préstamos donde se asientan los préstamos en su poder. Los socios pueden ser estudiantes o docentes. Un socio estudiante puede pedir prestado un libro sólo si no tiene más de 3 libros en su poder y si no tiene ningún libro vencido (20 días después de la fecha de retiro). Un socio docente no tiene límite en cuanto a la cantidad de libros prestados, pero solo puede pedir prestado un libro si no tiene alguna devolución vencida (superados **los días de préstamos que tiene cada docente**, desde la fecha de retiro). Inicialmente todos los docentes tienen 5 días de préstamo, pero si es responsable se le van agregando días de préstamo, individualmente.

Al pedir un préstamo, éste se crea con fecha de retiro solamente; la biblioteca asienta el préstamo en la ficha del libro y en la ficha del socio. Cuando se devuelve el libro, en el préstamo se indica la fecha de devolución.

La administración de la biblioteca necesita informes que indiquen:

- Qué cantidad de socios de un determinado tipo tienen (según un tipo de socio dado).
- Qué libros están prestados, cuyo plazo de devolución se encuentre vencido.
- Qué socio tiene en su poder determinado Libro.
- Cuáles son los Títulos que tiene la Biblioteca (Sin repetir, en caso de haber más de 1 ejemplar)
- Qué docentes devolvieron los libros siempre antes de su vencimiento (o el mismo día).

El diseño de la solución está representado en el diagrama de clases al final de este documento.

A continuación, una descripción de los métodos de las distintas clases:

Clase Biblioteca

- **prestarLibro(Calendar, Socio, Libro):** Crea el préstamo, y lo agrega en el libro y el socio.
- **devolverLibro(Libro):** Asigna la fecha de devolución del préstamo con la fecha actual. Si el libro no se encuentra prestado lanzar una excepción de tipo **LibroNoPrestadoException** con el mensaje descriptivo "*El libro <<título>> no se puede devolver ya que se encuentra en la biblioteca*".
- **cantidadDeSociosPorTipo(String):** devuelve la cantidad de socios del tipo pasado como parámetro.
- **prestamosVencidos():** devuelve una colección con los préstamos vencidos al día de la fecha.
- **docentesResponsables():** devuelve una colección con los docentes responsables.
- **quienTieneElLibro(Libro):** devuelve el nombre del Socio que tiene el libro con el título ingresado, y si no se encuentra prestado lanza una excepción de tipo **LibroNoPrestadoException** con el mensaje descriptivo "*El libro se encuentra en la biblioteca*".
- **buscarSocio(int):** devuelve el Socio que tiene el dni pasado como parámetro, o null si no lo encuentra.
- **listaDeSocios():** devuelve un String según formato1.
- **listaDeLibros():** devuelve un String según formato2.
- **listaDeTitulos():** devuelve un String con la lista de los títulos con los que cuenta la Biblioteca.
- **listaDeDocentesResponsables():** devuelve un String según formato3.

Clase Libro

- **ultimoPrestamo():** retorna el último préstamo del libro.
- **prestado():** devuelve true si el libro se encuentra prestado.
- **toString():** devuelve el siguiente String: **Título:** <<título>>

Ejemplo: **Título:** JAVA. Como Programar



Clase Socio

- **cantLibrosPrestados()**: devuelve la cantidad de libros en poder del Socio.
- **toString()**: devuelve el siguiente String:

D.N.I.: <<dni>> || <<nombre y apellido>> ([Docente|Estudiante])

Ejemplo:

D.N.I.: 14524782 || Juan Perez (Docente) || Libros Prestados: 6

- **puedePedir()**: devuelve true si el socio no tiene ningún préstamo vencido.

Clase Docente

- **esResponsable()**: devuelve true si el Docente nunca tuvo ni tiene un préstamo vencido.
- **soyDeLaClase()**: devuelve el String "Docente".
- **cambiarDiasDePrestamos(int)**: adiciona días de préstamo al docente. Es un premio a la responsabilidad.

Clase Estudiante

- **soyDeLaClase()**: devuelve el String "Estudiante".
- **puedePedir()**: devuelve true si el Estudiante no tiene ningún préstamo vencido y si no tiene más de 3 libros prestados (en su poder).

Clase Prestamo

- **vencido(Calendar)**: devuelve true si la fecha pasada como parámetro es mayor que la fecha de vencimiento (fecha de retiro más los días de préstamos asignados).
- **toString()**: devuelve el siguiente String:

Retiro: <<fecha de retiro>> - **Devolución:** <<fecha de devolución>>

Libro: <<título del libro>>

Socio: <<nombre del socio>>

Ejemplo:

Retiro: 2014/10/15 - **Devolución:** 2014/10/28

Libro: JAVA. Como Programar

Socio: Juan Perez

Formatos de listados

Formato1: Listado de Socios

Lista de Socios:

- 1) **D.N.I.:** <<dni>> || <<nombre y apellido>> (<<tipo>>) || **Libros Prestados:** <<cant. prést. actuales>>
- 2) **D.N.I.:** <<dni>> || <<nombre y apellido>> (<<tipo>>) || **Libros Prestados:** <<cant. prést. actuales>>
- 3) **D.N.I.:** <<dni>> || <<nombre y apellido>> (<<tipo>>) || **Libros Prestados:** <<cant. prést. actuales>>

Ejemplo:

Lista de Socios:

- 1) **D.N.I.:** 14524782 || Juan Perez (Docente) || **Libros Prestados:** 6
- 2) **D.N.I.:** 17982110 || Juan Fernández (Docente) || **Libros Prestados:** 1
- 3) **D.N.I.:** 10912002 || María Alegre (Docente) || **Libros Prestados:** 0
- 4) **D.N.I.:** 28987498 || Francisco Paenza (Estudiante) || **Libros Prestados:** 2
- 5) **D.N.I.:** 31987123 || Cesar Milstein (Estudiante) || **Libros Prestados:** 3

Cantidad de Socios del tipo Estudiante: 2

Cantidad de Socios del tipo Docente: 3



Formato2: Listado de Libros

Lista de Libros:

- 1) **Titulo:** <<titulo>> || **Prestado:** (<<Si/No>>)
- 2) **Titulo:** <<titulo>> || **Prestado:** (<<Si/No>>)
- 3) **Titulo:** <<titulo>> || **Prestado:** (<<Si/No>>)

Ejemplo:

Lista de Libros:

- 1) **Titulo:** JAVA. Como Programar || **Prestado:** (No)
- 2) **Titulo:** Longman. Diccionario Pocket || **Prestado:** (Si)
- 3) **Titulo:** Vivir para contarla || **Prestado:** (No)

Formato3: Listado de Docentes Responsables

Lista de Socios:

- * **D.N.I.:** <<dni>> || <<nombre y apellido>> (<<tipo>>) || **Libros Prestados:** <<cant. prést. actuales >>
- * **D.N.I.:** <<dni>> || <<nombre y apellido>> (<<tipo>>) || **Libros Prestados:** <<cant. prést. actuales >>

Ejemplo:

Lista de Docentes Responsables:

- * **D.N.I.:** 14524782 || Juan Perez (Docente) || **Libros Prestados:** 6
- * **D.N.I.:** 17982110 || Juan Fernández (Docente) || **Libros Prestados:** 1

Se solicita:

1. Implementar en java todas las clases del diseño provisto
2. Declarar la clase **GestionBiblioteca**, donde se deben instanciar objetos de las clases del diseño y obtener los listados descriptos.

Usar los métodos apropiados para responder a las siguientes preguntas:

- ¿Qué cantidad de socios de tipo Estudiante hay?
- ¿Cuál es la lista de docentes que nunca han adeudado ni adeudan libros?
- ¿Cuál es la lista de libros? ¿Y la de socios?
- ¿Qué socio tiene prestado el libro “Programando con JAVA”?

NOTA:

- En todos los casos se deberá usar colecciones genéricas
- Todas las clases deben estar dentro del paquete “biblioteca”.

