

Autor: Francisco Yuste García
Fecha: 12 de mayo de 2025
Estudios: DAM1 485 UF2
Convocatoria: Ordinaria
Curso: 24-25

EJERCICIO DE CONVOCATORIA ORDINARIA.

Introducción.

Karl Konrad Koreander había trabajado cuarenta años en la librería del barrio y no soportaba que aquellos jóvenes lectores que requerían más tiempo sabotearan la agenda donde registraba las entradas y salidas de sus libros.

Necesitaba un sistema más confiable, que impidiera el préstamo de libros incunables.

También debe asegurarse de los periodos de penalización en días sin sacar libro, en caso de que un libro se entregase tarde. La regla es que un usuario sólo puede sacar un libro y cada día de retraso supone una penalización del doble de días sin sacar libros.

=====

Clase Biblioteca.

Para ello, contamos con un objeto Biblioteca, que contiene una colección de libros prestados y otra de libros en la estantería. Sólo existe un ejemplar de cada libro.

La clase Biblioteca ha de tener:

+ DATABASE, el nombre de la base de datos 'biblioteca.db'

+ DIARIO, el nombre de un archivo "diario.txt", donde se registran préstamos y recuperaciones de forma inmediata. Sólo se escribirá al final del mismo.

+ libros, un diccionario de libros donde a partir del ISBN se identifica al libro. Este diccionario almacena todos los libros con los que trabaja la biblioteca.

Un método prestar(libro, lector), donde se volcará en DIARIO el libro prestado al lector, en caso de que el lector no posea ya un libro prestado, el libro no esté ya prestado y además pueda prestarse por no ser un incunable.

Si la fecha a partir de la cual un usuario puede sacar un libro es igual o anterior a la fecha actual, se podrá prestar el libro.

Ejemplo si la fecha de fecha_posible_para_prestamo es a anterior o igual al tres de marzo. En este caso se registrará un mensaje del tipo "03/03/2025 09:15 Se presta el libro La historia interminable a Bástian Baltasar Bux". Este mensaje también se almacenará en el fichero de diario.

Ejemplo si la fecha de fecha_posible_para_prestamo es el 8 de marzo se registrará un mensaje del tipo: "03/03/2025 09:15 NO se presta el libro La historia interminable a Bástian Baltasar Bux porque le quedan 5 días de penalización."

Estos mensaje se almacenarán en el fichero DIARIO y no es incompatible con otros comentarios en el mismo fichero.

Si un libro se presta debe desaparecer de libros como objeto LibroAlmacenado y estar en libros como LibroPrestado.

Un LibroIncunable o un LibroPrestado no se puede prestar. "29/03/2025 09:15 Bástian Baltasar Bux jamás podrá sacar La Historia Interminable porque es un incunable." "29/03/2025 09:15 Bástian Baltasar Bux no podrá sacar La Historia Interminable porque ya está prestado."

Un método adquirir_nuevo_libro(libro) recibe un objeto de tipo LibroIncunable o LibroAlmacenado y lo añade a la lista libros, siempre que el ISBN no esté dado ya de alta. Además inserta en la base de datos los datos relativos al libro.

Un método registrar_nuevo_lector(lector) recibe un objeto de tipo Lector y lo inserta en la base de datos.

OBSERVA:

Sólo puede añadirse un lector con cierto DNI si no existe ya un lector con ese DNI.

Un método actualizar_libro(libro) recibe un objeto libro y actualiza su información en la base de datos.

Si se actualiza correctamente retorna true y false en otro caso.

OBSERVA:

En la base de datos debe existir un libro con el isbn del libro para actualizarlo.

Un método actualizar_lector(lector) recibe un objeto lector y actualiza su información en la base de datos.

Si se actualiza correctamente retorna true y false en otro caso.

OBSERVA:

En la base de datos debe existir un lector con el dni del lector para actualizarlo.

RECOMENDACIÓN:

Definir el método auxiliar registrar_en_diario(texto), que escribe en el fichero una cadena de caracteres con un retorno de carro al final (sólo uno).

Un método recibir(isbn, dni). Recibimos el ISBN. Establece el libro a disponible en estantería. Todo libro prestado posee una referencia (dni) del último lector al que se le prestó y la fecha en la que se prestó.

Si ya pasaron siete días, se escribirá en el diario:

"29/03/2025 09:15 Bástian Baltasar Bux devuelve el libro La historia interminable en fecha.".

Si pasaron cinco días, se escribirá en el diario:

"29/03/2025 09:15 Bástian Baltasar Bux devuelve tarde el libro La historia interminable 5 días.".

"29/03/2025 09:15 Bástian Baltasar Bux no podrá sacar libros hasta el 08/04/2025.".

Además, se actualizará la fecha a partir de la cual el lector podrá sacar libros.

Un LibroIncunable o un LibroAlmacenado no se puede recibir.

"29/03/2025 09:15 Bástian Baltasar Bux jamás debió tomar La Historia Interminable, porque es un incunable."

"29/03/2025 09:15 Bástian Baltasar Bux no podrá sacar La Historia Interminable porque ya está prestado."

OBSERVA:

La penalización es la fecha actual más el número de días de retraso multiplicada, por dos.

ADVERTENCIA:

Sólo deben cargarse en memoria un objeto lector cuando sea necesario, al prestar o recibir un libro. No puedes tener una lista de lectores cargada en memoria.

Si un libro se recibe debe desaparecer como objeto LibroPrestado y añadirse a libros como LibroAlmacenado.

La clase lector.

La clase Lector tiene las propiedades:

+ dni,
+ nombre,
+ apellido1,
+ apellido2
de tipo string.

+ fecha_posible_para_prestamo
de tipo Fecha.

En el constructor, cuando se crea un lector nuevo su fecha de penalización se establece al día anterior de su creación.

Esta fecha sólo se actualiza en el caso de que se devuelva un libro en una fecha posterior a la fecha indicada por la fecha de retorno esperada del libro entregado, tal y como se describe en el método recibir de la clase Biblioteca.

La clase libro.

La clase Libro, posee atributos

+ isbn,
+ titulo y
+ autor,
todos String.

No pueden existir objetos de la clase Libro.

Todos deben ser: LibroIncunable, LibroAlmacenado o LibroPrestado.

De la clase Libro hereda la clase LibroAlmacenado.

Define los atributos

+ categoría y
+ subcategoría, enteros.

Define el método ubicación() que devuelve una cadena de dos enteros separados por un punto.

Si la categoría Fantasía se corresponde con el código 3 y la subcategoría Medieval con el código 7, éste método devolvería la cadena '3.7'.

Redefine el constructor para recibir los atributos categoria y subcategoría, enteros.

Define el método asignar_lector(l), que recibe un objeto Lector y devuelve un booleano false si por algún motivo no puede prestarse el libro al lector y true en otro caso. Además registra en base de datos los campos de la fecha de retorno del libro, que es la actual más siete días, y el dni del lector asignado.

De la clase LibroAlmacenado hereda la clase LibroIncunable.

Redefine el método asignar_lector(l), que recibe un objeto Lector y devuelve un booleano false.

De la clase Libro hereda la clase LibroPrestado.

Redefine el método consultar_lector(), que devolverá un objeto Lector.

Redefine el método consultar_retorno(), que devolverá la fecha en la que el libro debe tornar.

OBSERVA:

Es posible que necesites definirte alguna clase auxiliar, como Fecha.

Evaluación.

Para evaluaros ejecutaré un script que se os facilitará, que deberá ejecutar las sentencias y volcar cierta salida esperada en DIARIO.

ATENCIÓN:

Deberás de crear la base de datos tal como hemos visto en clase.

Aunque tienes libertad para definir el esquema que consideres oportuno de base de datos, debes satisfacer todas las llamadas a métodos que se hacen en el script evaluativo, sin modificarlo.

ATENCIÓN:

En este enunciado existen métodos que pueden requerir el objeto self en python.

Dejo a vuestro criterio el incluirlo cuando consideréis necesario.

Pueden existir pequeñas diferencias respecto de los mensajes aportados como ejemplos, pero cuando se solicita cierta entrada en el fichero, ésta debe existir.

FORMATO DE ENTREGA.

Debes subir el fichero biblioteca.zip, que contendrá en la carpeta APELLIDO_NOMBRE los ficheros:

- + enunciado.py, aportado por el profesor.
- + clase_biblioteca.py, define Biblioteca.
- + clase_lector.py, define Lector.
- + clase_libro.py, define Libro.
- + clase_almacenado.py, define LibroAlmacenado.
- + clase_incunable.py, define LibroIncunable.
- + clase_prestado.py, define LibroPrestado.
- + cualquier otra clase que necesite vuestro código.

Estos ficheros son necesarios, pero no suficientes, para la evaluación.

Han de satisfacerse los requisitos descritos en el enunciado y los siguientes.

Ejemplo la carpeta de Amélia Álamo será ALAMO_AMELIA, en mayúsculas y sin acentos.
No respetar la estructura minorará puntos.

Sistema operativo de referencia: el GNU/Linux empleado en clase.

Si el proyecto depende de un IDE concreto para funcionar y no se puede ejecutar, se puntuará con cero.

La ausencia de comentarios supondrá también una minoración de la nota en la medida de la gravedad observada.