Library-Management-System

Release v1.0.0

Madan Singh

Mar 18, 2024

CONTENTS:

CHAPTER

ONE

LIBRARY-MANAGEMENT-SYSTEM

1.1 book module

class Book(*title*, *author*, *isbn*)

Bases: `object`

Represents a book in the library.

delete(*library*)

Deletes the book from the library.

Parameters
library (Library) – The library object.

static search_by_author(*library*, *author*)

Searches for books by author in the library.

Parameters

- **library** (Library) – The library object.
- **author** (*str*) – The author to search for.

Returns
A list of books by the given author.

Return type
list

static search_by_isbn(*library*, *isbn*)

Searches for books by ISBN in the library.

Parameters

- **library** (Library) – The library object.
- **isbn** (*str*) – The ISBN to search for.

Returns
A list of books with the given ISBN.

Return type
list

static search_by_title(*library*, *title*)

Searches for books by title in the library.

Parameters

- **library** (Library) – The library object.
- **title** (*str*) – The title to search for.

Returns
A list of books with matching titles.

Return type
list

update(*title=None*, *author=None*, *isbn=None*)

Updates the book information.

Parameters

- **title** (*str, optional*) – The new title of the book.
- **author** (*str, optional*) – The new author of the book.
- **isbn** (*str, optional*) – The new ISBN of the book.

1.2 check module

class Check

Bases: object

Represents book check-in and check-out operations.

static check_in(*library*, *book*)

Checks in a book to the library.

Parameters

- **library** (Library) – The library object.
- **book** (Book) – The book to be checked in.

static check_out(*library*, *user*, *book*)

Checks out a book from the library.

Parameters

- **library** (Library) – The library object.
- **user** (User) – The user checking out the book.
- **book** (Book) – The book to be checked out.

Returns
True if the book was successfully checked out, False otherwise.

Return type
bool

1.3 library module

class Library

Bases: object

Represents a library management system.

add_book(*book*)

Adds a book to the library.

Parameters
book (Book) – The book object to add.

add_user(*user*)

Adds a user to the library.

Parameters
user (User) – The user object to add.

remove_book(*book*)

Removes a book from the library.

Parameters
book (Book) – The book object to remove.

remove_user(*user*)

Removes a user from the library.

Parameters
user (User) – The user object to remove.

1.4 main module

main()

print_books(*books*)

Prints a list of books.

print_users(*users*)

Prints a list of users.

1.5 models module

1.6 storage module

class Storage

Bases: object

Handles file-based storage operations for the library data.

static load_books(*filename*)

Loads a list of books from a JSON file.

Parameters
filename (*str*) – The name of the JSON file.

Returns
A list of book objects loaded from the file.

Return type
list

static load_users(*filename*)

Loads a list of users from a JSON file.

Parameters
filename (*str*) – The name of the JSON file.

Returns
A list of user objects loaded from the file.

Return type
list

static save_books(*books*, *filename*)

Saves a list of books to a JSON file.

Parameters

- **books** (*list*) – The list of book objects to save.
- **filename** (*str*) – The name of the JSON file.

static save_users(*users*, *filename*)

Saves a list of users to a JSON file.

Parameters

- **users** (*list*) – The list of user objects to save.
- **filename** (*str*) – The name of the JSON file.

1.7 testing module

class TestLibraryManagementSystem(*methodName='runTest'*)

Bases: TestCase

setUp()

Hook method for setting up the test fixture before exercising it.

tearDown()

Hook method for deconstructing the test fixture after testing it.

test_check_in()

test_check_out()

test_save_and_load_books()

test_save_and_load_users()

1.8 user module

class User(*name*, *user_id*)

Bases: object

Represents a user in the library management system.

delete(*library*)

Deletes the user from the library.

Parameters
library (Library) – The library object.

static search_by_id(*library*, *user_id*)

Searches for users by ID in the library.

Parameters

- **library** (Library) – The library object.
- **user_id** (*str*) – The ID to search for.

Returns
A list of users with the given ID.

Return type
list

static search_by_name(*library*, *name*)

Searches for users by name in the library.

Parameters

- **library** (Library) – The library object.
- **name** (*str*) – The name to search for.

Returns
A list of users with matching names.

Return type
list

update(*name=None*, *user_id=None*)

Updates the user information.

Parameters

- **name** (*str, optional*) – The new name of the user.
- **user_id** (*str, optional*) – The new ID of the user.

CHAPTER

TWO

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

INDEX