LU20.A03 - Bibliotheksverwaltung erweitern - JSON speichern und laden

Erweitern Sie die Bibliotheksverwaltung aus LU20.A01, damit die Buchliste bei veränderungen ins File `books.json` gespeichert. Wenn die Applikation geladen wird, soll `init_books` das JSON laden.

Allgemeine Infos

Vorlage

Im Vorlage-Repository ist die Musterlösung zu der Aufgabe LU20.A02, erweitern Sie diese Lösung um die benötigten Funktionen und passen Sie bestehende Funktionen gemäss Beschreibung an.

Vorgehen

Wie immer beim Programmieren gilt:

1. Wählen Sie einen Programmteil aus, dessen Vorgänger korrekt funktioniert.
2. Codieren Sie den neuen Programmteil.
3. Testen Sie den neuen Programmteil,
    1. Falls die Tests nicht erfolgreich sind, gehen Sie zurück zu Schritt 2.
4. Führen Sie alle Tests mit `pytest -vv` aus.
5. Falls mehr Tests erfolgreich sind als zuvor: Commit & Push.

Nr	Programmteil	Vorgänger	Tests
1	init_books	-	library_test.py > `test_init_books`

Testing

Die einzelnen Programmteile (Funktionen, Attribute, Property) werden mittels Unit Tests überprüft.

Beschreibungen

Der Code stellt ein terminalbasiertes Bibliotheksverwaltungssystem zur Verwaltung von Buchausleihen dar. Es wird mit einer Reihe von Büchern („Herr der Ringe“-Trilogie) initialisiert und ermöglicht es den Benutzern, Ausleihdatensätze hinzuzufügen, indem sie Buchnamen und Ausleihdetails (Ausleih- und Rückgabedaten, Anzahl der Ausleihtage) eingeben. Das System berechnet und zeigt die Mietkosten an, einschließlich der Strafen für überfällige Rückgaben. Die Benutzer interagieren über ein einfaches Textmenü, in dem sie Optionen zum Hinzufügen von Vermietungen, zum Anzeigen von Salden oder zum Beenden des Programms auswählen. Die

Schnittstelle besteht hauptsächlich aus dem Lesen von Benutzereingaben und dem Ausdrucken von Informationen direkt im Terminal.

Modul library.py

save_books()

Argumente	Dictionary mit Büchern
Rückgabewerte	keine

Logik

Die Funktion wandelt das Dictionary mit Büchern in ein JSON um und speichert das JSON als File `books.json`.

Das Umwandeln von `datetime`-Objekten in JSON ist nicht ganz einfach.

Schauen Sie sich dazu unter anderem folgende Webseiten an:

- https://lidatong.github.io/dataclasses-json/
- https://dev.to/kfuquay/extending-pythons-json-encoder-7k0

⇒ *GitHub Repo für externe Besucher*

GitHub Repository https://github.com/templates-python/m319-lu20-a03-library-json-read-write

*Lernende am BZZ **müssen** den Link zum GitHub Classroom Assignment verwenden*

m310-LU20

Kevin Maurizi, Marcel Suter

From:
https://wiki.bzz.ch/ - **BZZ - Modulwiki**

Permanent link:
https://wiki.bzz.ch/de/modul/m319/learningunits/lu20/aufgaben/bibliothekerweitertspeichern

Last update: **2025/06/23 07:45**