Aufgabe 5 - Applikation nach Vorgabe erstellen

Ziel

Sie können Objekte nach Vorgabe eines Klassen- und Sequenzdiagramms erstellen und kommunizieren lassen.

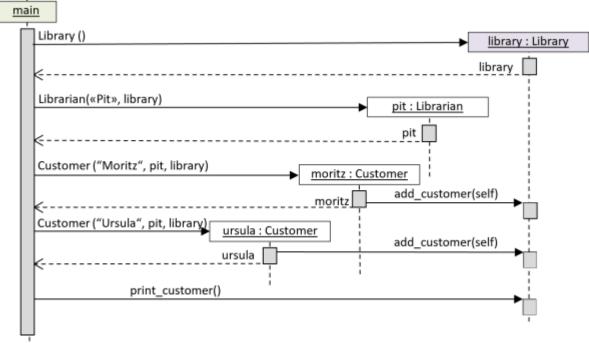
Hinweise

- Lösen Sie die Aufgabe jeweils step by step und führen Sie dann den Code aus, um zu prüfen, ob Ihre Implementation erfolgreich ist.
- Die Klassen Book und Library werden nicht bearbeitet. Lesen Sie aber dennoch die Beschreibungen der Methoden, damit Sie wissen, wie diese anzuwenden sind.
- Eine OO-Anwendung startet damit, dass in der main-Methode alle wichtigen Klassen instanziiert und deren Beziehungen sichergestellt werden.
- Das Klassendiagramm gibt den statischen Aufbau der Anwendung wieder. Main + main() Library - booklist : Book[] - customers : Customer[] Librarian + add_customer(customer : Customer) - name : String + search_customer(name : String) : Customer - library: Library + print_customer() + Librarian(name : String, library : Library) + add_book(a_book : Book) : String + buy_new_book(title : String, isbn: String) + remove_book(book : Book) + print_inventory() + borrow_a_book_by_title(title : String) : Book + get_a_book_from_customer(borrowed_book : Book) + search_book_by_title(title : String) : Book + remove_book(title : String) + borrow_book(location : String) : Book + remind customer(name: String) + put_back_book(borrowed_book : Book) Customer Book - name : String - title : String - reminded : boolean (= false) - isbn : String - librarian : Librarian - location : String - book : Book (= None) + Book(title: String, isbn: String) + Customer(name :String, librarian : Librarian, library : Library) + print() + get_title(): String +get_name(): String + get_isbn(): String + get_borrowed_book() : Book + get_location(): String + borrow_a_book_by_title(title : String) + set_location(location : String) + bring_bak_a_book() + remind() + is_reminded(): boolean

Auftrag

Teil 1: Die Objekte werden erzeugt.

- 1. Akzeptieren Sie das Assignment in GitHub Classroom und klonen Sie das Repo in Ihre Entwicklungsumgebung.
- 2. Implementieren Sie den Konstruktor der Klassen Customer.
 - Achtung: Beim Erzeugen eines Customer-Objektes meldet sich dieses selber (proaktiv) bei der Library an. Sie können das im Ablauf des Konstruktors erkennen, da dort der Aufruf add customer ausgeführt wird.
- 3. Führen Sie nun in test library.py die folgende Tests aus, welche fehlerfrei sein müssen:
 - ∘ test add and print customers
 - ∘ test_search_customer
 - ∘ test search customer failed
- 4. Pushen Sie die aktuelle Lösung auf GitHub.
- 5. Implementieren Sie den Konstruktor der Klasse Librarian.
 - Hinweis: Es gibt in diesem Zustand der Klasse keine Tests, die pr
 üfen, ob der Konstruktor erfolgreich angelegt wurde. Wenn Sie unsicher sind, ob Sie die Aufgabe richtig gelöst haben, frage Sie bei Ihrer Lehrperson nach.
- 6. Implementieren Sie nun in der main-Methode (main.py) die Instanziierung der Objekte sowie die Methodenaufrufe gemäss dem gezeigten Sequenzdiagramm.
 - Hinweis: Halten Sie sich an die Anweisungen im Quellcode.



7. Führen Sie dann das Programm aus. Es muss fehlerfrei laufen. Die Ausgabe soll wie folgt aussehen:



- 8. Testen Sie das Programm mit der Methode test_main_part1 in test_main.py. Wie immer muss auch dieser Test fehlerfrei ablaufen.
- 9. Pushen Sie den Teilauftrag 1 auf GitHub

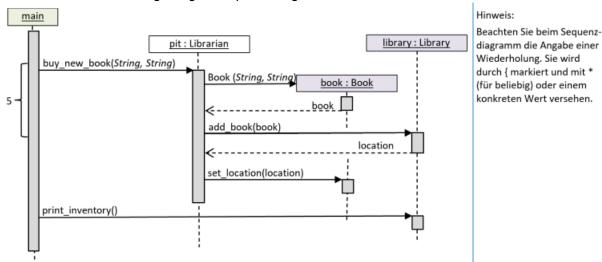
https://wiki.bzz.ch/ Printed on 2025/11/18 16:08

Teil 2: Der Bibliothek werden 5 Bücher zugeführt.

- 1. Erzeugen Sie für die 2. Teilaufgabe einen neuen Branch. So ist sichergestellt, dass Sie im Notfall jederzeit auf eine lauffähige Version zurückgreifen können.
- 2. Ergänzen Sie den Code der Methode buy_new_book in der Klasse Librarian. Hinweise:
 - Halten Sie sich an die Anweisungen im Code der Datei librarian.py
 - Beachten Sie bitte die Beschreibung der benötigten Methoden in den Klassen Library und Book.
 - Der Wert des Attributs location ist ein Zufallswert der in der Methode add_book()
 generiert wird.
 - Halten Sie sich an das gezeigte Sequenzdiagramm.
- 3. Führen Sie den Test test_buy_new_book in der Datei test_librarian.py aus. Er muss fehlerfrei ablaufen.
- 4. Pushen Sie die aktuelle Lösung auf GitHub.
- 5. Ergänzen Sie den Code in der main-Methode. Halten Sie sich an die Anweisungen im Code der Datei main.py.

Hinweise:

- Übernehmen Sie die Buchtitel und ISBN-Nummern genau so, wie im Printout angegeben.
- Halten Sie sich an das gezeigte Sequenzdiagramm.



6. Führen Sie das Programm aus. Es muss fehlerfrei laufen.

Die Ausgabe soll wie folgt aussehen:

```
Inventar:

ISBN : 3-345-678 / Titel: Ich bin dann mal weg / Ablage : D8

ISBN : 6-444-856 / Titel: im West nichts neues / Ablage : B2

ISBN : 3-3345-678-X / Titel: Das Omen / Ablage : C10

ISBN : 3-4321-334 / Titel: Harry Potter, die neue Welt / Ablage : A7

ISBN : 3-2123-554 / Titel: die schönsten Zugreisen / Ablage : B6
---
```

Hinweis:

- · Der Ablageort wird durch die Library generiert und kann darum variieren!
- 7. Pushen Sie die lauffähige Teilaufgabe 2 auf GitHub.

Teil 3: Ursula und Moritz leihen sich je ein Buch aus.

Ursula wählt das Buch "Das Omen" und Moritz das Buch "Ich bin dann mal weg".

1. Erstellen Sie - basierend auf dem Branch aus Teilaufgabe 2 - einen neuen Branch für die Teilaufgabe 3.

- Ergänzen Sie in der Klasse Librarian die Methode borrow_a_book_by_title. Hinweise:
 - · Halten Sie sich an die Anweisungen im Code der Datei librarian.py
 - · Beachten Sie bitte die Beschreibung der benötigten Methoden in den Klassen Library und Book.
 - · Halten Sie sich an das gezeigte Sequenzdiagramm. Der Ablauf ist etwas komplex gehalten, dies aber bewusst, um möglichst viel "Kommunikation" zu erzielen.
- 3. Führen Sie den Test test_borrow_a_book_by_title in der Datei test_librarian.py aus. Er muss fehlerfrei ablaufen.
- 4. Pushen Sie diese Lösung auf GitHub.
- 5. Führen Sie nun den Test test_borrow_a_book_by_unknown_title in der Datei test_librarian.py aus. Er muss fehlerfrei ablaufen.
- 6. Pushen Sie diese Lösung auf GitHub.
- 7. Ergänzen Sie nun in der Klasse Customer die Methode borrow_a_book_by_title.

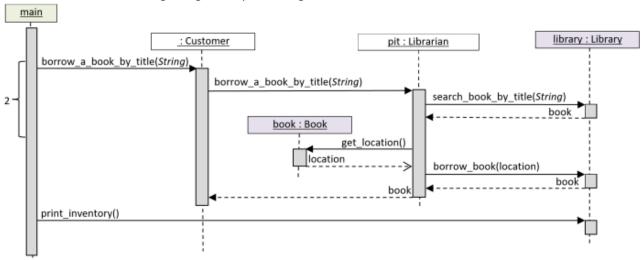
 Achtung! Sie benötigen hier die Referenz self._book auf ein Book-Objekt, um den Rückgabewert aus dem Methodenaufruf zu speichern. Das lässt sich aus dem Sequenzdiagramm so nicht entnehmen. Dieses Wissen müssen Sie als Fachperson hier einbringen. Fehlt die Deklaration dieses Attributs im Konstruktor, müssen Sie das jetzt nachholen.

Hinweis:

- · initialisieren Sie self. book mit dem Wert None
- 8. Führen Sie nun den Test test_borrow_a_book_by_title in der Datei test_customer.py aus. Er muss fehlerfrei ablaufen.
- 9. Pushen Sie die lauffähige Lösung auf GitHub.
- Ergänzen Sie den Code in der main-Methode. Halten Sie sich an die Anweisungen im Code der Datei main.py.

Hinweis:

· Halten Sie sich an das gezeigte Sequenzdiagramm.



11. Führen Sie das Programm aus. Es muss fehlerfrei laufen. Die Ausgabe soll in etwa wie folgt aussehen:

https://wiki.bzz.ch/ Printed on 2025/11/18 16:08

```
Ursula hat das Buch 'Das Omen' erhalten.

Moritz hat das Buch 'Ich bin dann mal weg' erhalten.

Inventar:

ISBN : 6-444-856 / Titel: im West nichts neues / Ablage : D10

ISBN : 3-4321-334 / Titel: Harry Potter, die neue Welt / Ablage : C2

ISBN : 3-2123-554 / Titel: die schönsten Zugreisen / Ablage : B4
```

12. Pushen Sie die lauffähige Teilaufgabe 3 auf GitHub.

Hinweis:

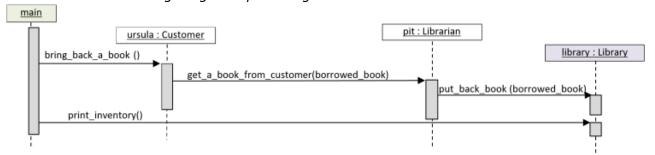
 \cdot die main-Methode wird nicht mehr mit einem eigenen Test geprüft, da alle Methoden die benötigt werden, schon getestete sind. Sie müssen hier in Eigenverantwortung schauen, ob das angezeigte Ergebnis korrekt ist.

Teil 4: Ursula bringt ihr Buch zurück.

- Ergänzen Sie den Code der Methode get_a_book_from_customer in der Datei librarian.py.
- 2. Testen Sie den Code mit der Methode test_get_a_book_from_customer in der Datei test_librarian.py.
- 3. Pushen Sie die Lösung auf GitHub.
- 4. Ergänzen Sie den Code der Methode bring_back_a_book in der Datei customer.py.
- 5. Testen Sie den Code mit der Methode test_bring_back_a_book in der Datei test customer.py.
- 6. Pushen Sie die Lösung auf GitHub.
- 7. Ergänzen Sie den Code in der main-Methode.

Hinweis:

- · Halten Sie sich an die Anweisungen im Code der Dateien Librarian.py, Customer.py und main.py.
- · Halten Sie sich an das gezeigte Seguenzdiagramm.



8. Führen Sie das Programm nun aus. Es muss fehlerfrei laufen. Die Ausgabe soll in etwa wie folgt aussehen:

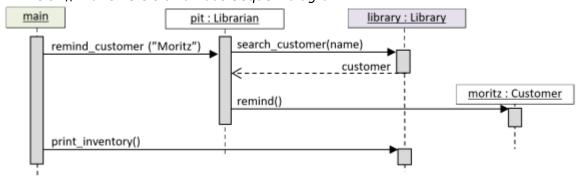
```
Ursula hat das Buch 'Das Omen' zurückgebracht
Inventar:
ISBN : 6-444-856 / Titel: im West nichts neues / Ablage : D10
ISBN : 3-4321-334 / Titel: Harry Potter, die neue Welt / Ablage : C2
ISBN : 3-2123-554 / Titel: die schönsten Zugreisen / Ablage : B4
ISBN : 3-3345-678-X / Titel: Das Omen / Ablage : A9
---
```

9. Pushen Sie den Teilauftrag 4 auf GitHub

Teil 5: Moritz wird gemahnt

- 1. Ergänzen Sie den Code der Methode remind_customer in der Datei librarian.py.
- 2. Testen Sie den Code mit der Methode test_get_a_book_from_customer in der Datei test_librarian.py.
- 3. Pushen Sie die Lösung auf GitHub.
- 4. Ergänzen Sie den Code in der main-Methode.

 Hinweis: \\ Halten Sie sich an das Sequenzdiagramm.



5. Führen Sie das Programm aus. Es muss fehlerfrei laufen.

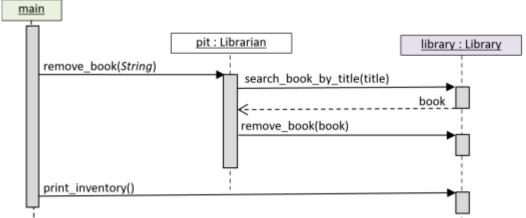
Die Ausgabe soll in etwa wie folgt aussehen:



6. Pushen Sie den Teilauftrag 5 auf GitHub.

Teil 6 : Der Bibliothekar entfernt ein Buch aus der Bibliothek. Entfernt wird das Buch "Harry Potter, die neue Welt"

- 1. Ergänzen Sie den Code in der Methode remove book in der Datei librarian.py.
- Testen Sie den Code mit der Methode test_remove_book in der Datei test_librarian.py.
- 3. Pushen Sie die Lösung auf GitHub.
- 4. Ergänzen Sie den Code in der main-Methode. Hinweise:
 - · Halten Sie sich an das Sequenzdiagramm.



5. Führen Sie das Programm aus. Es muss fehlerfrei laufen. Die Ausgabe soll in etwa wie folgt aussehen:

https://wiki.bzz.ch/ Printed on 2025/11/18 16:08

```
entferne Buch 'Harry Potter, die neue Welt'
Inventar:
ISBN : 6-444-856 / Titel: im West nichts neues / Ablage : D6
ISBN : 3-2123-554 / Titel: die schönsten Zugreisen / Ablage : B1
ISBN : 3-3345-678-X / Titel: Das Omen / Ablage : B6
```

6. Pushen Sie den Teilauftrag 6 auf GitHub

Teil 7: Ursula will ein Buch mit einem Titel, den es nicht gibt

1. Ergänzen Sie den Code in der main-Methode. (alles andere sollte schon implementiert sein.) Die Ausgabe des Programms muss in etwa wie folgt aussehen:

Das angefragte Buch ist nicht vorhanden

Dauer

3 - 5 Stunden

Abgabe

Pushen Sie - wie in der Anleitung erwähnt - jede lauffähige Version mit den jeweiligen Tests.



© René Probst

From:

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

Permanent link:

https://wiki.bzz.ch/modul/m320/learningunits/lu04/aufgaben/lu3-aufgabe_5

