LU20.L01 - Bibliotheksverwaltung

rental.py

```
from dataclasses import dataclass
from datetime import datetime, timedelta


@dataclass
class Rental:
    """
    the rental data for a book
    """
    rental_date: datetime
    return_date: datetime
    num_rental_days: int

    @property
    def cost(self):
        days_overdue = (self.to_date - self.from_date).days -
self.included_days

        if days_overdue > 0:
            return round(4.5 + days_overdue*3.35,2)
        else:
            return round(4.5,2)
```

library.py

```
from rental import Rental
from datetime import datetime


def init_books():
    """
    initialize the dictionary with some books
    :return: books dictionary
    """
    books = {
        'LOTR 1': [],
        'LOTR 2': [],
        'LOTR 3': [],
    }
    return books


def read_rental():
```

```
    """
    asks the user to input the data for a book rental
    :return:
    """
    rental_date = read_date('Enter rental date (dd.mm.yyyy): ')
    num_rental_days = read_int('Enter number of rental days: ', 1, 99)
    return_date = read_date('Enter return date (dd.mm.yyyy): ')
    return Rental(rental_date, return_date, num_rental_days)


def add_rental(books):
    """
    adds the rentals to the book dictionary
    :param books:
    :return:
    """
    while True:
        book_name = input('Enter the book name: ')
        if book_name in books:
            rental = read_rental()
            books[book_name].append(rental)
            choice = input('Do you want to add another rental? (y/n):
').strip().lower()
            if choice != 'y':
                break
        else:
            print('Book not found in the library.')


def show_balance(books):
    """
    shows the balance of all the rentals
    :param books:
    :return: None
    """
    for book, rentals in books.items():
        print(f'Statement for {book}')
        total_cost = 0

        for rental in rentals:
            print(f'  - {rental.rental_date.strftime("%d.%m.%Y")}: CHF
{rental.cost}')
            total_cost += rental.cost

        print(f'Total: CHF {round(total_cost, 2)}')


def read_int(prompt, minimum=None, maximum=None):
    """
```

```
    asks the user to enter a whole number between minimum and maximum
    :param prompt:
    :param minimum:
    :param maximum:
    :return: the number
    """
    while True:
        try:
            num = int(input(prompt))
        except ValueError:
            print('Please, enter a whole number!')
            continue
        else:
            if minimum is not None and num < minimum:
                print('Please, enter a number greater than or equal
to', minimum)
                continue
            if maximum is not None and num > maximum:
                print('Please, enter a number less than or equal to',
maximum)
                continue
            return num


def read_date(prompt):
    """
    asks the user to input a datetime
    :param prompt:
    :return: the datetime
    """
    while True:
        try:
            date_str = input(prompt)
            rental_date = datetime.strptime(date_str, '%d.%m.%Y')
            return rental_date
        except ValueError:
            print('Please enter a valid date.')


def main():
    """
    manage the book rentals at a library
    :return: None
    """
    books = init_books()
    while True:
        print('\nLibrary Management System')
        print('1. Add Rental')
        print('2. Show Balances')
        print('3. Exit')
        choice = read_int('Enter your choice (1/2/3): ', 1, 3)
```

```
        if choice == 1:
            add_rental(books)
        elif choice == 2:
            show_balance(books)
        elif choice == 3:
            break


if __name__ == '__main__':
    main()
```

⇒ *GitHub Repo für externe Besucher*

GitHub Repository https://github.com/templates-python/m319-lu20-a01-library

*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/loesungen/bibliothek

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