LU04.L01 - Library

test_book.py

```
import pytest as pytest

from book import Book


def test_init_correct():
    """
    create a book with title and isbn
    """
    book = Book('My book', '978-8-456-01789-2')
    assert type(book) is Book
    assert book.title == 'My book'
    assert book.isbn == '978-8-456-01789-2'

@pytest.mark.xfail
def test_init_wrong():
    """
    create a book with one value or none
    """
    book = Book('wrong')

def test_output(book1):
    """
    check the output when printing a book
    """
    assert book1.__str__() == 'Book(title=\'First book\',
isbn=\'978-1-111-11111-1\', location=None)'

def test_location_1(book1):
    """
    is location empty when creating a book with title and isbn
    """
    assert type(book1) is Book
    assert book1.location is None

def test_location_2():
    """
    is location empty when creating a book with title, isbn and
location
    """
    book = Book('The location', '978-1-111-11111-1', 'shelf a1')
    assert type(book) is Book
    assert book.location is None

@pytest.fixture
```

```
def book1():
    return Book('First book', '978-1-111-11111-1')
```

test_library.py

```
import pytest

from customer import Customer
from library import Library


def test_init():
    library = Library()
    assert type(library.customers) is list
    assert library.customers == []
    assert type(library.booklist) is list
    assert library.booklist == []

def test_add_customers(library_empty, customer_a, customer_b):
    assert len(library_empty.customers) == 2
    assert customer_a in library_empty.customers
    assert customer_b in library_empty.customers

@pytest.fixture
def library_empty():
    return Library()

@pytest.fixture
def library_customers(library_empty):
    return Library()

@pytest.fixture
def customer_a(library_empty):
    return Customer('Hanna', None, library_empty)

@pytest.fixture
def customer_b(library_empty):
    return Customer('Edy', None, library_empty)
```

M450-LU04