Sprint 1 (Java) (Adolfo)

Empezar tarea

Fecha de entrega Viernes a las 23:59 **Puntos** 100
Entregando una URL de página web o una carga de archivo
Disponible 26 de mar en 0:00 - 6 de abr en 23:59

Objective

To understand the principles of unit testing, practice writing unit tests, and enhance the Library Management System with additional features in Java.

Part 1: Understanding the Existing Code

1. **Review the Code**:
   - Go through the Book, Patron, Library, and LibraryTest classes. Please, understand how each class works and how they interact with each other.
   - Identify methods in the Library class that modify the state of the system and which of those are critical to test.
2. **Identify Test Cases**:
   - Adding a book that already exists.
   - Checking out a book that is already checked out.
   - Returning a book that was never checked out.
   - Write down scenarios that they think are critical to test. For example:

1. You should write a **matrix table** that look like below, every test you put there should have a valid unit test case in our Test Class.
2. Source Code https://github.com/adolfoarroyotec/UnitTesting/blob/main/README.md
3. Summarize Chapter 5. Mocks and test fragility from **Unit Testing Principles, Practices, and Patterns, you can find the reference below in references** and propose in a single Unit **mock** test for calling the public testable API https://catfact.ninja/fact

   Above mocked Unit Test code can be included in the document where you write your test matrix or in your code.

Feature / Class	Test Case ID	Unit Test Case Name	Test Scenario	Input	Expected Outcome	Remarks

Library Class	TC-LIB-001	TestAddBookWithError	Add Book - Duplicate Book	Add Book with title "Moby Dick" (already exists)	System should reject the addition with an error message	Test for handling duplicates
Book Class	TC-BOOK-001	TestBookCreated	Create Book - Valid Book	Book with title "Brave New World"	Book object should be created successfully	Valid test case
Patron Class	TC-PATRON-001	TestRegisterPatron	Register Patron - Valid Patron	Patron with name "John Doe"	Patron should be registered successfully	Valid test case

Part 2: Writing Unit Tests

4. **Create New Tests**:
    - Instruct students to create unit tests for the following scenarios:
        - **Test for Adding Duplicate Books**: Ensure that the system does not allow adding a book that already exists in the library.
        - **Test for Nonexistent Book Checkout**: Attempt to check out a book that does not exist in the library and assert that the operation fails.
        - **Test for Fine Calculation**: Write tests that check if the fine is calculated correctly for various overdue scenarios.
        - **Test for Listing Books and Patrons**: Verify that the list of available books and patrons reflects the current state of the library correctly.
5. **Test Coverage** should be > 90 in every class (**LibraryTest** is included with 2 tests for example purposes)

Stucture of the program

```
LibraryManagementSystem/
│
├── src/
│   ├── Book.java
│   ├── Patron.java
│   ├── Library.java
│   └── LibraryTest.java
│
└── README.md
```