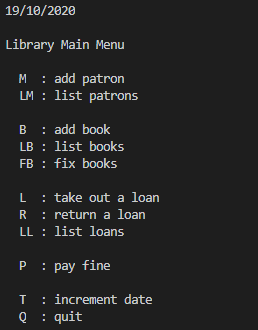
Bug 1: Incorrect Calculation of Fines

**Description:**

When a book becomes overdue by one day, no fine is imposed

**Menu:**

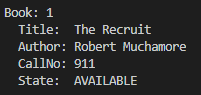


**Inputs:**

**Purpose:** Book Creation

|  |  |
| --- | --- |
| **Prompt** | **Input** |
| Choice : | b |
| Enter author: | Robert Muchamore |
| Enter Title: | The Recruit |
| Enter call number: | 911 |

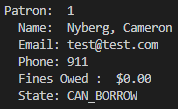
**Outcome:** Book Created Successfully;



**Purpose:** Patron Creation

|  |  |
| --- | --- |
| **Prompt** | **Input** |
| Choice : | m |
| Enter last name: | Nyberg |
| Enter first name: | Cameron |
| Enter email: | test@test.com |
| Enter phone number: | 911 |

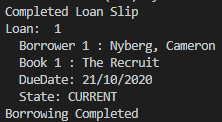
**Outcome:** Patron Created Successfully;



**Purpose:** Borrow Book

|  |  |
| --- | --- |
| **Prompt** | **Input** |
| Choice : | l |
| Swipe patron card (press <enter> to cancel): | 1 |
| Scan Book (<enter> completes): | 1 |
| Scan Book (<enter> completes): | ENTER |
| Commit loans? (Y/N): | y |

**Outcome:** Book Borrowed Successfully;

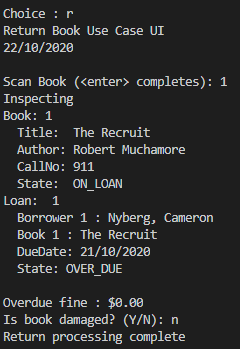


**Purpose:** Bug Replication

**Note:** The due date listed previous is ‘21/10/2020’, with the current date listed as ‘19/10/2020’. An ‘increment date’ (of three days) is therefore required, in order to meet the use case of a book overdue by one day.

|  |  |  |
| --- | --- | --- |
| **Prompt** | **Input** | **Outcome** |
| Choice : | t | Increment days selected |
| Enter number of days: | 3 | Date incremented by 3 |
| Choice : | r | Return a loan selected |
| Scan Book (<enter> completes): | 1 | Book 1 selected, with an OVER\_DATE associated with the attached Loan |

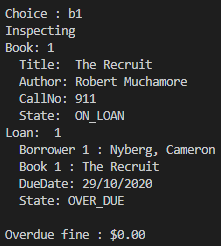
**Outcome:** Overdue book returned, identified as overdue and no fine is associated.



**“Simplification” 80 lines of over-engineering**

The bug is easily replicated with then newly added B1 option, which will check if a Loan exists and if not, attempt to create one (creating a Book & Patron, as necessary). From here, we hard coded (disgusting) increment thrice, to force an OVER\_DUE state (this is bad, if the Loan was pre-existing but the Calendar class is unreliable with the getDaysDifference method).

Sample Output:



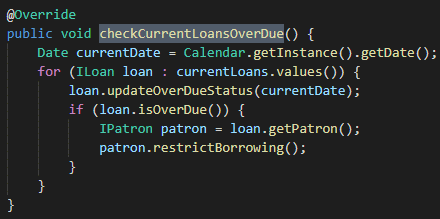
**Tracing**

Logically speaking, we know that the fines are incurred, once a calculation of the loan being overdue is complete; first thing to check is where this happens.

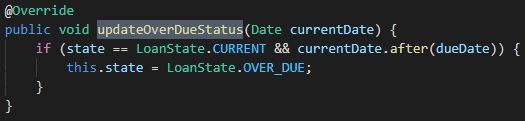
However, first let’s check the actual breaking point (where our fines start to begin and stop to break).

|  |  |
| --- | --- |
| **Days Overdue** | **Fine Incurred** |
| 0 | 0 |
| 1 | 0 |
| 2 | $1.00 |
| 3 | $1.00 |

In our ‘Main’ method, if we search for any check methods that occur, we are presented with 1 (library.checkCurrentLoansOverDue();). Following to the function definition..



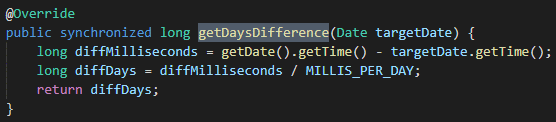
The only really line of interest here is loan.updateOverDueStatus(currentDate); - we search further to this function definition, only to find that this method is simply responsible for changing the attribute on the loan (dictating whether or not the loan is overdue).



Just below the definition for checkCurrentLoansOverDue, there is a function calculateOverDueFine.

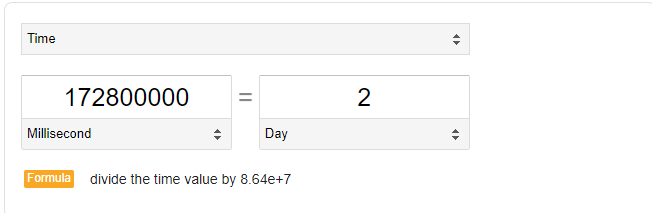
During the debugging, of my Simplification, I discovered that getDaysDifference() was incredibly unreliable (and have noted as such, above).

Checking the function definition



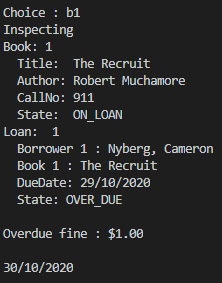
We can note that this function relies heavily on MILLIS\_PER\_DAY, which has a value of 172800000.

Upon entering the value 172800000 into Google, we can convert this to 2 days



This means we should expect an exponential issue (x^2).

Halving this value for testing.. sample output:



Committing the test fix to git.