Table of Contents

Bug 1 & 2

Description

`Function : getDaysDifference()(Class-Calendar.java, Package-entities)`

This function is intend to calculate difference between two dates. This is used in our program to get date difference to calculate fine. This function cause both described bugs.

Pre-conditions

This function required to pass a date, this date will be differentiated with current date.

Post-Conditions

This function should return date day difference between argument passed date and current date.

Test Result

FAIL (This function returns half value of actual difference and if value is odd then it returns lower base, that's why we got $0 fine when we overdue by one day)

Screenshots

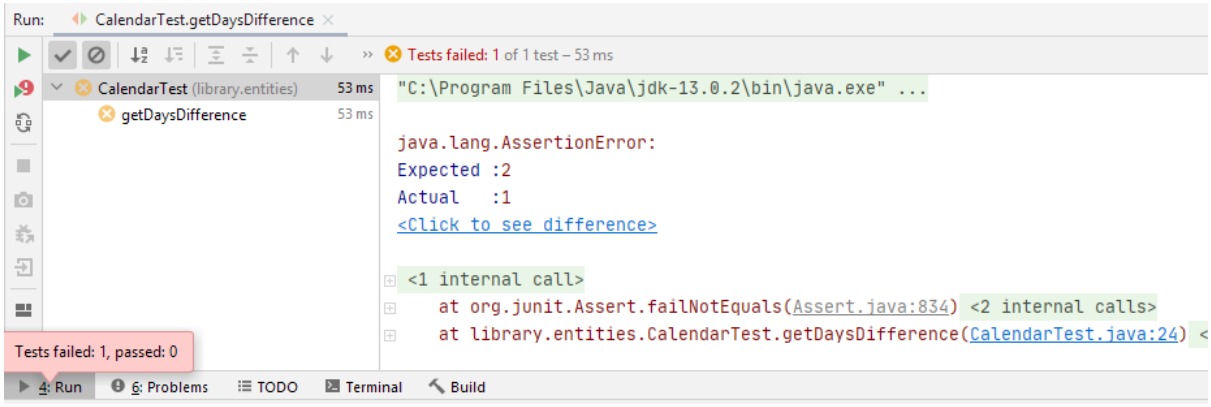
Source code

```
62      @Override
63      public synchronized long getDaysDifference(Date targetDate) {
64          long diffMilliseconds = getDate().getTime() - targetDate.getTime();
65          long diffDays = diffMilliseconds / MILLIS_PER_DAY;
66          return diffDays;
67      }
68
```

Test code

```
11      // this function intend to give difference between two dates
12      @Test
13      public void getDaysDifference() {
14          Calendar calendar = Calendar.getInstance();
15          Date date = library.entities.Calendar.getInstance().getDate();
16
17          calendar.setTime((Date)date.clone());
18          // setting due date, two days before current date
19          calendar.add(Calendar.DATE, amount: -2);
20
21          long actualResult = library.entities.Calendar.getInstance().getDaysDifference(calendar.getTime());
22          long expectedResult = 2L;
23
24          assertEquals(expectedResult,actualResult);
25      }
```

Test result


Run:
CalendarTest.getDaysDifference
Tests failed: 1 of 1 test – 53 ms
CalendarTest (library.entities) 53 ms
getDaysDifference 53 ms
"C:\Program Files\Java\jdk-13.0.2\bin\java.exe" ...
java.lang.AssertionError:
Expected :2
Actual :1
<Click to see difference>
<1 internal call>
at org.junit.Assert.failNotEquals(Assert.java:834) <2 internal calls>
at library.entities.CalendarTest.getDaysDifference(CalendarTest.java:24)
Tests failed: 1, passed: 0
4: Run
6: Problems
TODO
Terminal
Build

Bug 3

Description

- This bug not found when we do return book operation but it found when we do pay fines operation in main menu.
- This bug is incur fine second time for overdue fine. This bug is not listed in task.
- There is nothing to make automate test for that.
- We just need to remove any one of them.

First time

```
36  public void bookScanned(int bookId) {
37      if (!controlState.equals(ControlStateConstants.READY)) {
38          throw new RuntimeException("ReturnBookControl: cannot call bookScanned except in READY state");
39      }
40      IBook currentBook = library.getBookById(bookId);
41      if (currentBook == null) {
42          returnBookUI.display( object: "Invalid Book Id");
43          return;
44      }
45      if (!currentBook.isOnLoan()) {
46          returnBookUI.display( object: "Book has not been borrowed");
47          return;
48      }
49      currentLoan = library.getCurrentLoanByBookId(bookId);
50      double overDueFine = 0.0;
51      if (currentLoan.isOverDue()) {
52          overDueFine = library.calculateOverDueFine(currentLoan);
53          currentLoan.getPatron().incurFine(overDueFine);
54      }
55      returnBookUI.display( object: "Inspecting");
56      returnBookUI.display(currentBook);
57      returnBookUI.display(currentLoan);
58
59      if (currentLoan.isOverDue()) {
60          returnBookUI.display(String.format("\nOverdue fine : $%.2f", overDueFine));
61      }
62      returnBookUI.setState(IReturnBookUI.UIStateConstants.INSPECTING);
63      controlState = ControlStateConstants.INSPECTING;
```


Second time

```
210     @Override
211     public void dischargeLoan(ILoan loan, boolean isDamaged) {
212         IPatron patron = loan.getPatron();
213         IBook book = loan.getBook();
214
215         double overDueFine = calculateOverDueFine(loan);
216         patron.incurFine(overDueFine);
217
218         Integer bookId = book.getId();
219         if (isDamaged) {
220             patron.incurFine(DAMAGE_FEE);
221             damagedBooks.put(bookId, book);
222         }
223         loan.discharge(isDamaged);
224
225         currentLoans.remove(bookId);
226         setPatronBorrowingRestrictions(patron);
227     }
```