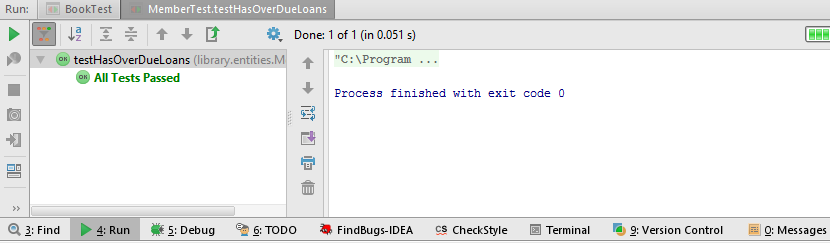
# JUNIT TEST FOR MEMBER Test .JAVA

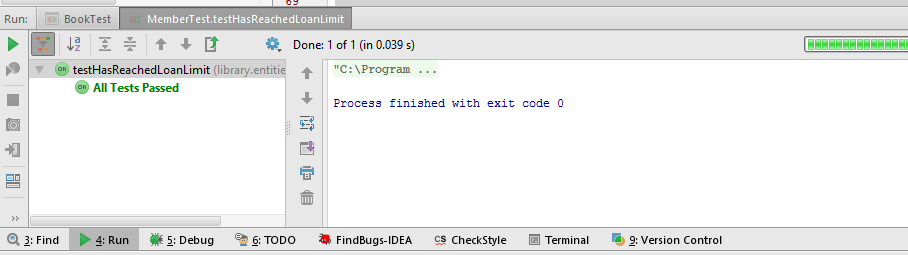
**Version History**

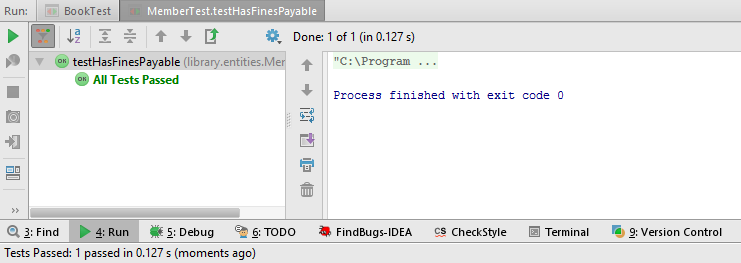
|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Description** | **Author** |
| 1.0 | 1/10/2016 | Initial version | KAMAL RAJ KANDEL |

@Test  
*/\*\*  
 \* Test if the method returns false  
 \* MockData has no overdue  
 \*/***public void** testHasOverDueLoans() **throws** Exception {  
 **boolean** result = **member**.hasOverDueLoans();  
 **boolean** expectedResult = **false**;  
 *assertEquals*(expectedResult, result);  
}  
 **RESULT ON THE SCREEN**

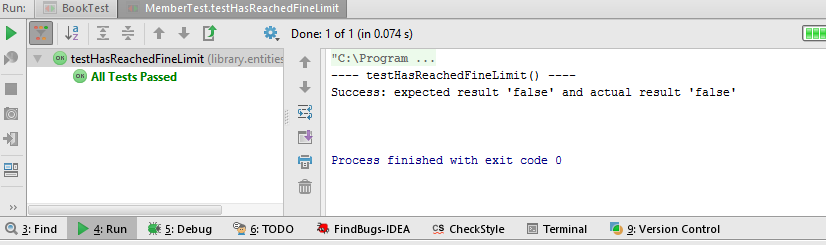


@Test  
*/\*\*  
 \* Test if the method returns false  
 \* member in mockdata has no reachedLoanLimit  
 \*/***public void** testHasReachedLoanLimit() **throws** Exception {  
 **boolean** result = **member**.hasReachedFineLimit();  
 **boolean** expectedResult = **false**;  
 *assertEquals*(expectedResult, result);  
}

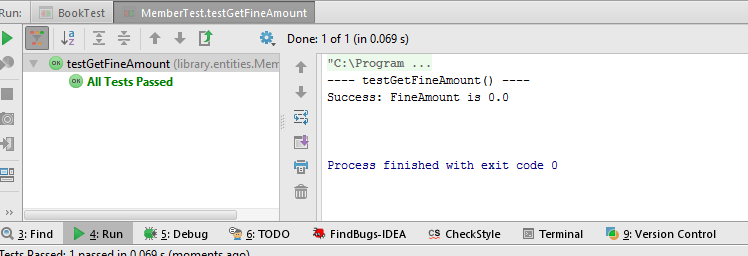
  
  
@Test  
*/\*\*  
 \* Test if the method returns false  
 \* member in mockdata has no finesPayable  
 \*/***public void** testHasFinesPayable() **throws** Exception {  
 **boolean** result = **member**.hasFinesPayable();  
 **boolean** expectedResult = **false**;  
 *assertEquals*(expectedResult, result);  
}



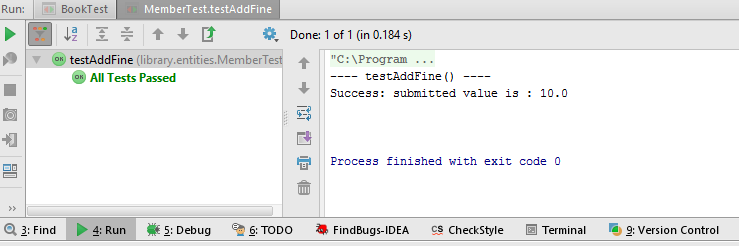
@Test  
*/\*\*  
 \* Test if the method returns false  
 \* member in mockdata has no reachedFineLimit  
 \*/***public void** testHasReachedFineLimit() **throws** Exception {  
 System.***out***.println(**"---- testHasReachedFineLimit() ----"**);  
 **boolean** expectedResult = **false**;  
 **try** {  
 **boolean** result = **member**.hasReachedFineLimit();  
 **if**(expectedResult == result) {  
 System.***out***.println(**"Success: expected result '"** + expectedResult + **"' and actual result '"** + result + **"'"**);  
 }  
 } **catch**(Exception e) {  
 *fail*(**"Fail : "** + e.toString());  
 }  
  
  
  
 System.***out***.println(**"\n"**);  
}



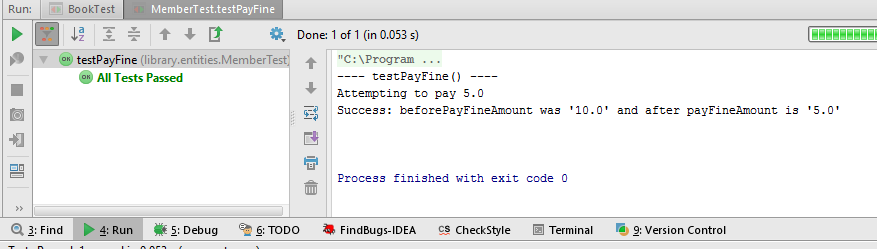
@Test  
**public void** testGetFineAmount() **throws** Exception {  
 System.***out***.println(**"---- testGetFineAmount() ----"**);  
  
 **float** expectedFineAmount = 0.0f;  
 **float** actualFineAmount;  
 **try** {  
 actualFineAmount = **member**.getFineAmount();  
 **if**(actualFineAmount == expectedFineAmount) {  
 System.***out***.println(**"Success: FineAmount is "** +**member**.getFineAmount());  
 }  
 } **catch**(Exception e) {  
 *fail*(**"Fail: "** + e);  
 }  
 System.***out***.println(**"\n"**);  
}



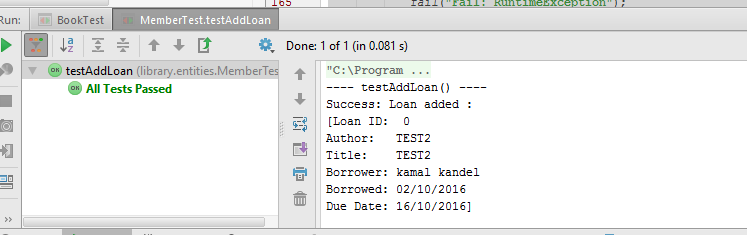
@Test  
**public void** testAddFine() **throws** Exception {  
 System.***out***.println(**"---- testAddFine() ----"**);  
  
 **member**.addFine(10);  
 **float** expectedFineAmount = 10;  
 **try** {  
 **member**.getFineAmount();  
 **if**(expectedFineAmount == **member**.getFineAmount()){  
 System.***out***.println(**"Success: submitted value is : "** + expectedFineAmount);  
 }  
 **else**{  
 System.***out***.println(**"Fail: submitted fineAmount does not match to member.getFineAmount()"**);  
 }  
 } **catch**(Exception e) {  
 *fail*(**"Fail: "** + e + **"\n submitted value is : "** + **member**.getFineAmount());  
 }  
 System.***out***.println(**"\n"**);  
}



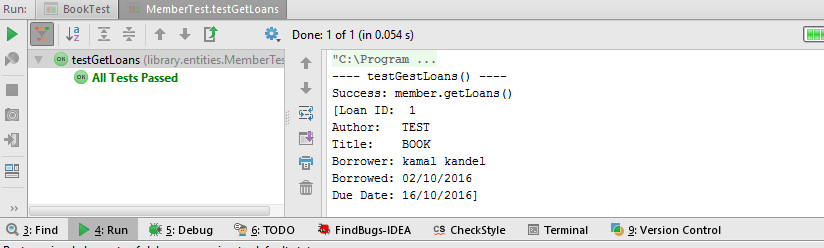
@Test  
**public void** testPayFine() **throws** Exception {  
 System.***out***.println(**"---- testPayFine() ----"**);  
 **float** beforePayFine = 10;  
 **member**.addFine(beforePayFine);  
 **try** {  
 System.***out***.println(**"Attempting to pay 5.0"**);  
 **member**.payFine(5);  
 System.***out***.println(**"Success: beforePayFineAmount was '"** + beforePayFine + **"' and after payFineAmount is '"** + **member**.getFineAmount() +**"'"**);  
 } **catch**(Exception e) {  
 *fail*(**"Fail: "** + e);  
 }  
 System.***out***.println(**"\n"**);  
}



@Test  
**public void** testAddLoan() **throws** Exception {  
 System.***out***.println(**"---- testAddLoan() ----"**);  
 **book** = **new** Book(**"TEST2"**, **"TEST2"**, **"TEST2"**, 20);  
 **member** = **new** Member(**"kamal"**, **"kandel"**, **"045123698"**, **"kamal\_kandel62@hotmail.com"**, 20);  
 **try** {  
 **this**.**loan** = **this**.**loanDAO**.createLoan(**member**, **book**);  
 **member**.addLoan(**loan**);  
 System.***out***.println(**"Success: Loan added : \n"** + **member**.getLoans());  
 } **catch**(RuntimeException e) {  
 *fail*(**"Fail: RuntimeException"**);  
 }  
 System.***out***.println(**"\n"**);  
}



@Test  
**public void** testGetLoans() **throws** Exception {  
 System.***out***.println(**"---- testGestLoans() ----"**);  
 **try** {  
 **member**.getLoans();  
 System.***out***.println(**"Success: member.getLoans() \n"** + **member**.getLoans());  
 } **catch**(RuntimeException e) {  
 *fail*(**"Fail: RuntimeException"**);  
 }  
 System.***out***.println(**"\n"**);  
}

  
@Test

**public void** testRemoveLoan() **throws** Exception {  
 System.***out***.println(**"---- testRemoveLoan() ----"**);  
 **try** {  
 System.***out***.println(**"Before Loan removed: \n"** + **member**.getLoans());  
 **member**.removeLoan(**loan**);  
 System.***out***.println(**"Success: Loan removed: \n"** + **member**.getLoans());  
 } **catch**(RuntimeException e) {  
 *fail*(**"Fail: RuntimeException"**);  
 *assertThat*(e,*instanceOf*(RuntimeException.**class**));  
 }  
 System.***out***.println(**"\n"**);  
}

