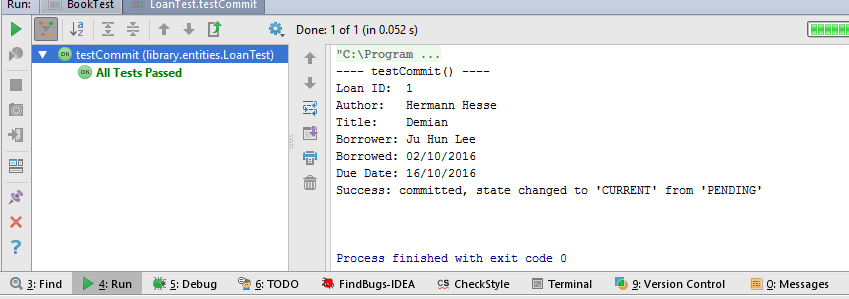
# JUNIT TEST FOR LOAN Test .JAVA

**Version History**

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

1.

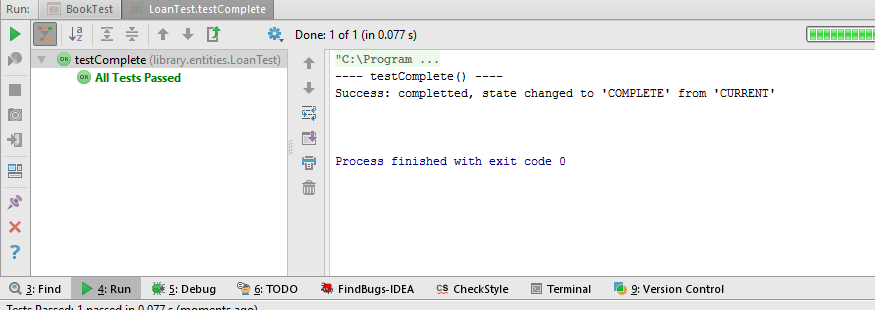
@Test  
 */\*\*  
 \* Test if state changes into 'Current' from 'Pending'  
 \* if (this.state != ELoanState.PENDING) { throw new RuntimeException(); }  
 \*/* **public void** testCommit() **throws** Exception {  
 System.***out***.println(**"---- testCommit() ----"**);  
 ELoanState expectedResult = ELoanState.***CURRENT***;  
 **try** {  
 **loan**.commit(1);  
 *assertEquals*(expectedResult, **loan**.getState());  
 System.***out***.println(**loan**.toString());  
 System.***out***.println(**"Success: committed, state changed to '"** + expectedResult + **"' from '"** + **state** + **"'"**);  
 } **catch**(RuntimeException e) {  
 *assertThat*(e, *instanceOf*(RuntimeException.**class**));  
 *fail*(**"Fail: RuntimeException caught"**);  
 }  
 System.***out***.println(**"\n"**);  
}  
 **RESULT ON THE SCREEN**

****

2.

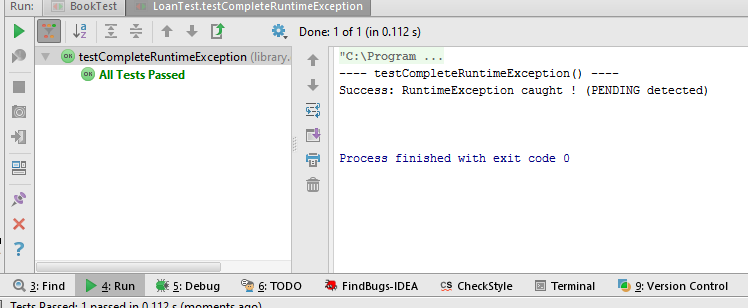
@Test  
*/\*\*  
 \* Test if state changes into 'COMPLETE' from 'CURRENT'  
 \* if (this.state != ELoanState.CURRENT && this.state != ELoanState.OVERDUE) { throw new RuntimeException(); }  
 \*/***public void** testComplete() **throws** Exception {  
 System.***out***.println(**"---- testComplete() ----"**);  
 ELoanState prevResult = ELoanState.***CURRENT***;  
 ELoanState expectedResult = ELoanState.***COMPLETE***;  
 **loanEntity**.setState(prevResult);  
 **try** {  
 **loanEntity**.complete();  
 System.***out***.println(**"Success: completted, state changed to '"** + expectedResult + **"' from '"** + prevResult + **"'"**);  
 } **catch** (RuntimeException e) {  
 *assertThat*(e, *instanceOf*(RuntimeException.**class**));  
 *fail*(**"Fail: RuntimeException caught"**);  
 }  
 System.***out***.println(**"\n"**);  
}

**RESULT ON THE SCREEN**

  
3.

@Test  
*/\*\*  
 \* Test if method catches RuntimeException when state is 'CURRENT' or 'OVERDUE'  
 \* if (this.state != ELoanState.CURRENT && this.state != ELoanState.OVERDUE) { throw new RuntimeException(); }  
 \*/***public void** testCompleteRuntimeException() **throws** Exception {  
 System.***out***.println(**"---- testCompleteRuntimeException() ----"**);  
 **loanEntity**.setState(ELoanState.***PENDING***);  
 **try** {  
 **loanEntity**.complete();  
 System.***out***.println(**"1"**);  
 } **catch** (RuntimeException e) {  
 *assertThat*(e, *instanceOf*(RuntimeException.**class**));  
 System.***out***.println(**"Success: RuntimeException caught ! ("** + **loanEntity**.getState() + **" detected)"**);  
 }  
 System.***out***.println(**"\n"**);  
}

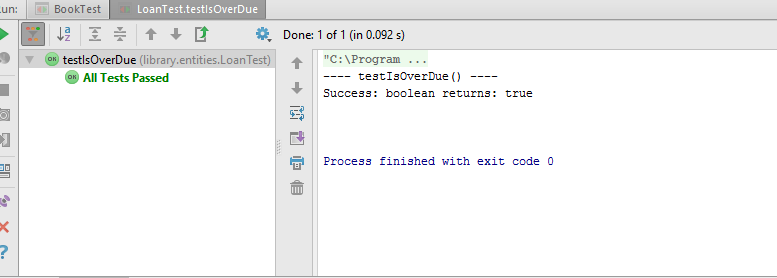
**RESULT ON THE SCREEN**



4.

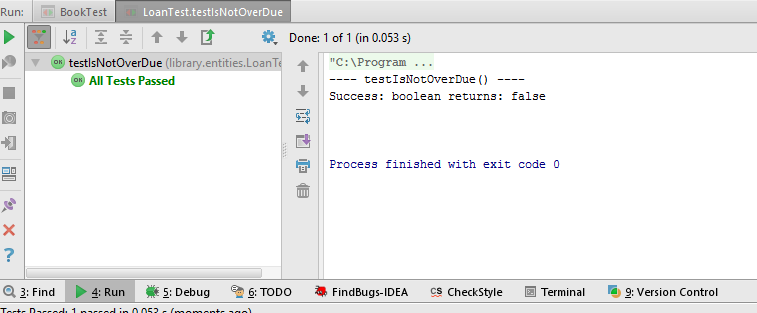
@Test  
*/\*\*  
 \* Test if the method returns 'true' boolean value  
 \* if (this.state == ELoanState.OVERDUE) { return true; }  
 \*/***public void** testIsOverDue() **throws** Exception {  
 System.***out***.println(**"---- testIsOverDue() ----"**);  
 ELoanState overdue = ELoanState.***OVERDUE***;  
 **boolean** actualResult = **false**;  
 **loanEntity**.setState(overdue);  
 **try** {  
 actualResult = **loanEntity**.isOverDue();  
 System.***out***.println(**"Success: boolean returns: "** + actualResult);  
 } **catch**(RuntimeException e) {  
 *fail*(**"Fail: return value is "** + actualResult);  
 }  
 System.***out***.println(**"\n"**);  
}

**RESULT ON THE SCREEN**



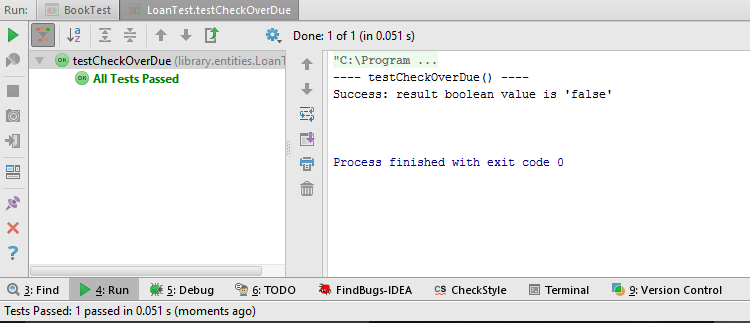
5.  
  
@Test  
*/\*\*  
 \* Test if the method returns 'false' boolean value  
 \* if (this.state == ELoanState.OVERDUE) { return true; }  
 \*/***public void** testIsNotOverDue() **throws** Exception {  
 System.***out***.println(**"---- testIsNotOverDue() ----"**);  
 ELoanState notOverDue = ELoanState.***COMPLETE***;  
 **boolean** actualResult = **false**;  
 **loanEntity**.setState(notOverDue);  
 **try** {  
 actualResult = **loanEntity**.isOverDue();  
 System.***out***.println(**"Success: boolean returns: "** + actualResult);  
 } **catch**(RuntimeException e) {  
 *assertThat*(e, *instanceOf*(RuntimeException.**class**));  
 *fail*(**"Fail: return value is: "** + actualResult);  
 }  
 System.***out***.println(**"\n"**);  
}

**RESULT ON THE SCREEN**



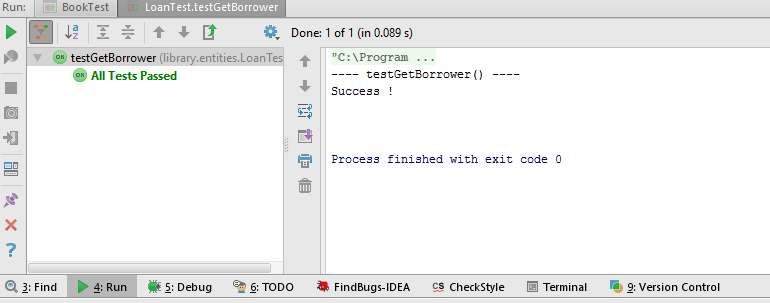
6.

@Test  
*/\*\*  
 \* Test if the method returns 'false' boolean value when state is 'CURRENT' or 'OVERDUE'  
 \* if (this.state != ELoanState.CURRENT && this.state != ELoanState.OVERDUE) { throw new RuntimeException(); }  
 \*/***public void** testCheckOverDue() **throws** Exception {  
 System.***out***.println(**"---- testCheckOverDue() ----"**);  
 **boolean** actualResult = **false**;  
 **try** {  
 **loanEntity**.setState(ELoanState.***CURRENT***);  
 actualResult =**loanEntity**.checkOverDue(**new** Date());  
 System.***out***.println(**"Success: result boolean value is '"** + actualResult + **"'"**);  
 } **catch**(Exception e) {  
 *fail*(**"Fail: return value is: "** + actualResult);  
 }  
 System.***out***.println(**"\n"**);  
}



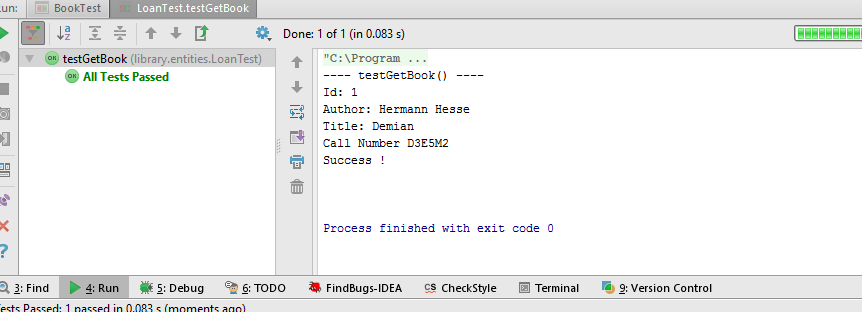
7.  
  
@Test  
**public void** testGetBorrower() **throws** Exception {  
 System.***out***.println(**"---- testGetBorrower() ----"**);  
  
 **try** {  
 System.***out***.println(**"Success !"**);  
 } **catch**(Exception e) {  
 *fail*(**"Fail: "** + e.toString());  
 }  
 System.***out***.println(**"\n"**);  
}

**RESULT ON THE SCREEN**



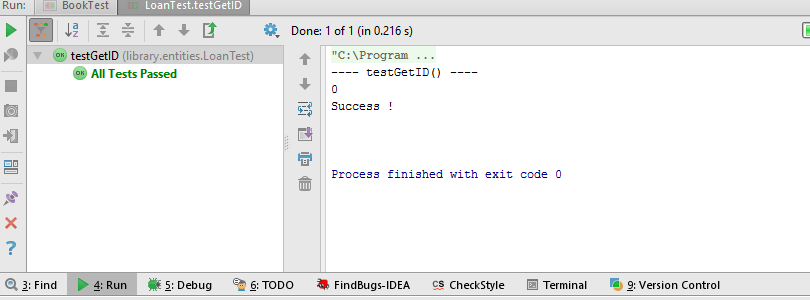
8.  
@Test  
**public void** testGetBook() **throws** Exception {  
 System.***out***.println(**"---- testGetBook() ----"**);  
 **try** {  
 **loan**.getBook();  
 System.***out***.println(**loan**.getBook());  
 System.***out***.println(**"Success !"**);  
 } **catch**(Exception e) {  
 *fail*(**"Fail: "** + e.toString());  
 }  
 System.***out***.println(**"\n"**);  
}

**RESULT ON THE SCREEN**

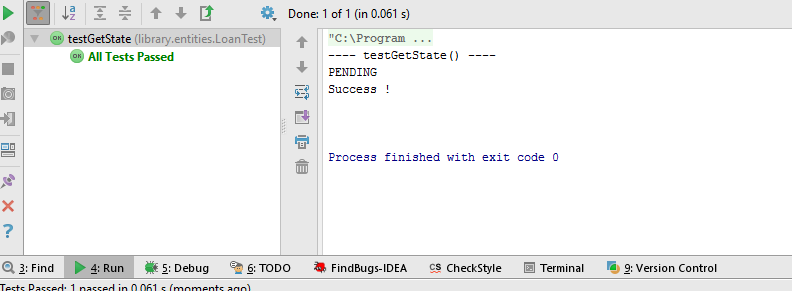


9.  
@Test  
**public void** testGetID() **throws** Exception {  
 System.***out***.println(**"---- testGetID() ----"**);  
  
 **try** {  
 **loan**.getID();  
 System.***out***.println(**loan**.getID());  
 System.***out***.println(**"Success !"**);  
 } **catch**(Exception e) {  
 *fail*(**"Fail: "** + e.toString());  
 }  
 System.***out***.println(**"\n"**);  
}

**RESULT ON THE SCREEN**



10.  
  
@Test  
**public void** testGetState() **throws** Exception {  
 System.***out***.println(**"---- testGetState() ----"**);  
  
 **try** {  
 **loan**.getState();  
 System.***out***.println(**loan**.getState());  
 System.***out***.println(**"Success !"**);  
 } **catch**(Exception e) {  
 *fail*(**"Fail: "** + e.toString());  
 }  
 System.***out***.println(**"\n"**);  
}



11.  
@Test  
**public void** testToString() **throws** Exception {  
 System.***out***.println(**"---- testToString() ----"**);  
 System.***out***.println(**loanEntity**.toString());  
 System.***out***.println(**"\n"**);  
}

**RESULT ON THE SCREEN**

