Test Case

Class under test: CopyController

# Generate copy number:

* Input: book number (String).
* Output: copy number (String).

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | 1 | 2 | 3 |
| Input | " SC0001" | "DR0001" | "MA0001" |
| Output | “SC000108” | "DR000102" | “MA000107” |
| Passed | Pass | Pass | Pass |

@Test  
public void generateCopyNumberTestCase1() throws Exception {  
 String bookNumber = "SC0001";  
 String number = new DecimalFormat("00")

.format((getHighestCopyNumber(bookNumber) + 1) % 10);  
 String copyNumber = bookNumber + number;  
 *assertEquals*(copyNumber, copyController.generateCopyNumber(bookNumber));  
}  
  
@Test  
public void generateCopyNumberTestCase2() throws Exception {  
 String bookNumber = "DR0001";  
 String number = new DecimalFormat("00")

.format((getHighestCopyNumber(bookNumber) + 1) % 10);  
 String copyNumber = bookNumber + number;  
 *assertEquals*(copyNumber, copyController.generateCopyNumber(bookNumber));  
}  
  
@Test  
public void generateCopyNumberTestCase3() throws Exception {  
 String bookNumber = "MA0001";  
 String number = new DecimalFormat("00")

.format((getHighestCopyNumber(bookNumber) + 1) % 10);  
 String copyNumber = bookNumber + number;  
 *assertEquals*(copyNumber, copyController.generateCopyNumber(bookNumber));  
}  
  
@Test  
public void checkAvailableCopyTestCase1() throws Exception {  
 *assertEquals*(false, copyController.checkAvailableCopy("SC0001"));  
}  
  
@Test  
public void checkAvailableCopyTestCase2() throws Exception {  
 *assertEquals*(true, copyController.checkAvailableCopy("MA0001"));  
}  
  
private int getHighestCopyNumber(String bookNumber) {  
 List<Copy> copyList = DataHandler.*requestCopyList*();  
 int max = 0;  
 for (Copy copy : copyList) {  
 if (copy.getBooknumber().equals(bookNumber)) {  
 int tmp = Integer.*parseInt*(copy.getCopynumber().substring(6));  
 if (tmp > max) {  
 max = tmp;  
 }  
 }  
 }  
 return max;  
}