Test Case

Use case: Add new copy

# Check book number:

* Input: book number (String).
* Output: result (boolean).

## Black box test:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| String length | = 6 | = 6 | = 6 | = 6 | != 6 | != 6 | != 6 | != 6 |
| First 2 chars | Are English letters | Are English letters | Not English letters | Not English letters | Are English letters | Are English letters | Not English letters | Not English letters |
| Next 4 chars | Are Digits | Not Digits | Are Digits | Not Digits | Are Digits | Not Digits | Are Digits | Not Digits |
| Output | true | false | false | false | false | false | false | false |

@Test  
public void checkBookNumberTestCase1() throws Exception {  
 *assertEquals*(true, addCopyController.checkBookNumber("XX1234"));  
}  
  
@Test  
public void checkBookNumberTestCase2() throws Exception {  
 *assertEquals*(false, addCopyController.checkBookNumber("XX1a34"));  
}  
  
@Test  
public void checkBookNumberTestCase3() throws Exception {  
 *assertEquals*(false, addCopyController.checkBookNumber("X@1234"));  
}  
  
@Test  
public void checkBookNumberTestCase4() throws Exception {  
 *assertEquals*(false, addCopyController.checkBookNumber("X@1k34"));  
}  
  
@Test  
public void checkBookNumberTestCase5() throws Exception {  
 *assertEquals*(false, addCopyController.checkBookNumber("XX1111abc"));  
}  
  
@Test  
public void checkBookNumberTestCase6() throws Exception {  
 *assertEquals*(false, addCopyController.checkBookNumber("XX1a34def"));  
}  
  
@Test  
public void checkBookNumberTestCase7() throws Exception {  
 *assertEquals*(false, addCopyController.checkBookNumber("X@1234ghi"));  
}  
  
@Test  
public void checkBookNumberTestCase8() throws Exception {  
 *assertEquals*(false, addCopyController.checkBookNumber("X@1k34klm"));  
}

## White box test:

* C0 100% Coverage test:

@Test  
public void checkBookNumberTestCase1() throws Exception {  
 *assertEquals*(true, addCopyController.checkBookNumber("XX1234"));  
}

* C1 100% Coverage test:

@Test  
public void checkBookNumberTestCase1() throws Exception {  
 *assertEquals*(true, addCopyController.checkBookNumber("XX1234"));  
}  
  
@Test  
public void checkBookNumberTestCase2() throws Exception {  
 *assertEquals*(false, addCopyController.checkBookNumber("XX1a34"));  
}  
  
@Test  
public void checkBookNumberTestCase3() throws Exception {  
 *assertEquals*(false, addCopyController.checkBookNumber("X@1234"));  
}  
  
@Test  
public void checkBookNumberTestCase4() throws Exception {  
 *assertEquals*(false, addCopyController.checkBookNumber("X@1k34"));  
}  
  
@Test  
public void checkBookNumberTestCase5() throws Exception {  
 *assertEquals*(false, addCopyController.checkBookNumber("XX1111abc"));  
}  
  
@Test  
public void checkBookNumberTestCase6() throws Exception {  
 *assertEquals*(false, addCopyController.checkBookNumber("XX1a34def"));  
}  
  
@Test  
public void checkBookNumberTestCase7() throws Exception {  
 *assertEquals*(false, addCopyController.checkBookNumber("X@1234ghi"));  
}  
  
@Test  
public void checkBookNumberTestCase8() throws Exception {  
 *assertEquals*(false, addCopyController.checkBookNumber("X@1k34klm"));  
}

# Check type:

* Input: type (String).
* Output: result (boolean).

## Black box test:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case | 1 | 2 | 3 | 4 |
| String length | = 1 | = 1 | != 1 | != 1 |
| First char | ‘B’ || ‘R’ | Not ‘B’ & Not ‘R’ | ‘B’ || ‘R’ | Not ‘B’ & Not ‘R’ |
| Output | true | false | false | false |

@Test  
public void checkTypeTestCase1() throws Exception {  
 *assertEquals*(true, addBookController.checkType("B"));  
}  
  
@Test  
public void checkTypeTestCase2() throws Exception {  
 *assertEquals*(false, addBookController.checkType("A"));  
}  
  
@Test  
public void checkTypeTestCase3() throws Exception {  
 *assertEquals*(false, addBookController.checkType("B12ab"));  
}  
  
@Test  
public void checkTypeTestCase4() throws Exception {  
 *assertEquals*(false, addBookController.checkType(""));  
}

## White box test:

* C0 100% Coverage test:

@Test  
public void checkTypeTestCase1() throws Exception {  
 *assertEquals*(true, addBookController.checkType("B"));  
}

* C1 100% Coverage test:

@Test  
public void checkTypeTestCase1() throws Exception {  
 *assertEquals*(true, addBookController.checkType("B"));  
}  
  
@Test  
public void checkTypeTestCase2() throws Exception {  
 *assertEquals*(false, addBookController.checkType("A"));  
}  
  
@Test  
public void checkTypeTestCase3() throws Exception {  
 *assertEquals*(false, addBookController.checkType("B12ab"));  
}  
  
@Test  
public void checkTypeTestCase4() throws Exception {  
 *assertEquals*(false, addBookController.checkType(""));  
}

# Check price:

* Input: price (String).
* Output: result (boolean).

## Black box test:

|  |  |  |
| --- | --- | --- |
| Test Case | 1 | 2 |
| All characters | Are Digits | Not Digits |
| Output | true | false |

@Test  
public void checkPriceTestCase1() throws Exception {  
 *assertEquals*(true, addBookController.checkPrice("700000"));  
}  
  
@Test  
public void checkPriceTestCase2() throws Exception {  
 *assertEquals*(false, addBookController.checkPrice("700abc00"));  
}

## White box test:

* C0 100% Coverage test:

@Test  
public void checkPriceTestCase1() throws Exception {  
 *assertEquals*(true, addBookController.checkPrice("700000"));  
}

* C1 100% Coverage test:

@Test  
public void checkPriceTestCase1() throws Exception {  
 *assertEquals*(true, addBookController.checkPrice("700000"));  
}  
  
@Test  
public void checkPriceTestCase2() throws Exception {  
 *assertEquals*(false, addBookController.checkPrice("700abc00"));  
}