**单元测试用例设计（db.unc）**

测试包： org.symagic.common.db.func

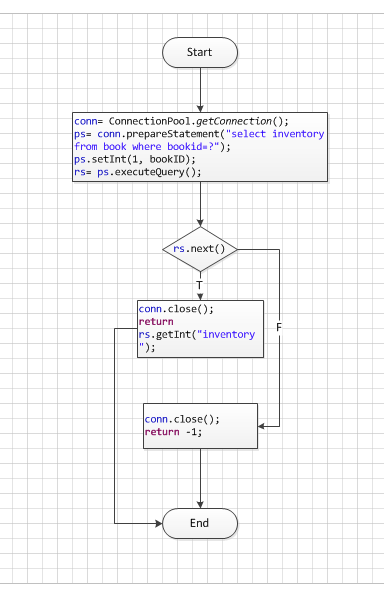
1. 测试类: DaoBook

测试方法： 判定/条件法

被测方法：

* 1. getInventory(int bookID)

流程图：

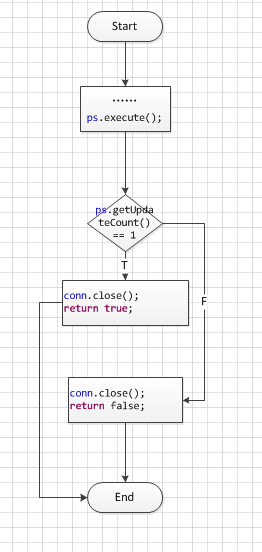


测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 输入 | 覆盖判定 | 覆盖条件 | 预期结果 |
| bookID |
| 1 | 1 | rs.next()=true | rs.next()=true | 30 |
| 2 | 1234 | rs.next()=false | rs.next()=false | -1 |

* 1. addBook(BeanBook book)

流程图：

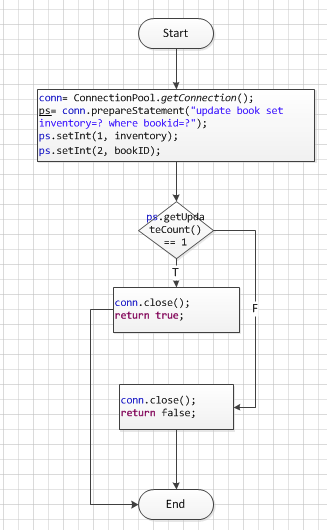


测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 输入 | 覆盖判定 | 覆盖条件 | 预期结果 |
| book |
| 3 | 实际输入A | ps.getUpdateCount() == 1 | ps.getUpdateCount() == 1 | true |
| 4 | BeanBook参数Offline("error"); | ps.getUpdateCount() != 1 | ps.getUpdateCount() != 1 | false |

* 1. setInventory(int bookID, int inventory)

流程图：

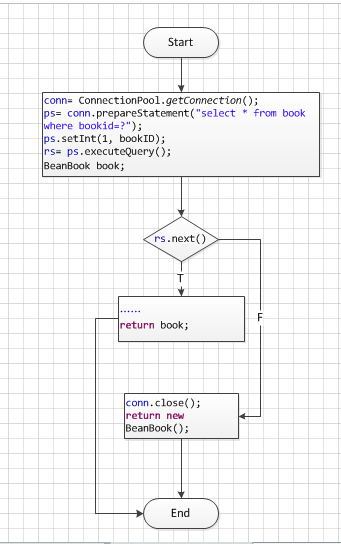


测试用例：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用例ID | 输入 | | 覆盖判定 | 覆盖条件 | 预期结果 |
| bookID | inventory |
| 5 | 1 | 30 | ps.getUpdateCount() == 1 | ps.getUpdateCount() == 1 | true |
| 6 | 15275 | 25 | ps.getUpdateCount() != 1 | ps.getUpdateCount() != 1 | false |

* 1. getDeatil(int bookID)

流程图：

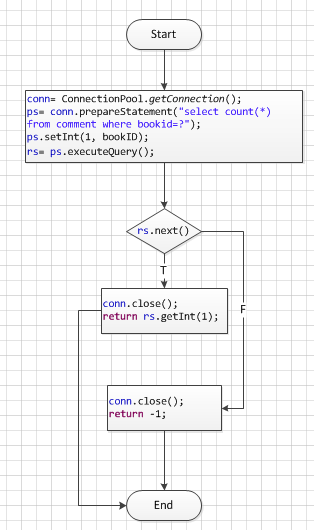


测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 输入 | 覆盖判定 | 覆盖条件 | 预期结果 |
| bookID |
| 7 | 1 | rs.next()=true | rs.next()=true | 一个有值的BeanBook对象 |
| 8 | 1234 | rs.next()=false | rs.next()=false | 一个空的BeanBook对象 |

* 1. getCommnetNumber(int bookID)

流程图：

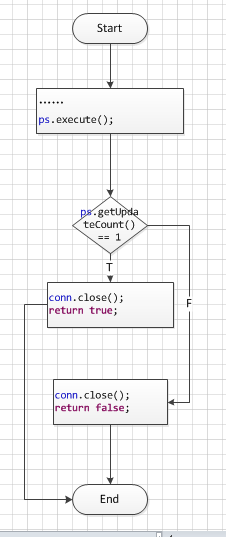


测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 输入 | 覆盖判定 | 覆盖条件 | 预期结果 |
| bookID |
| 9 | 1 | rs.next()=true | rs.next()=true | 1 |
| 10 | 1234 | rs.next()=false | rs.next()=false | -1 |

* 1. publishComment(BeanComment comment)

流程图：



测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 输入 | 覆盖判定 | 覆盖条件 | 预期结果 |
| BeanComment comment |
| 11 | 有内容的BeanComment对象 | ps.getUpdateCount() == 1 | ps.getUpdateCount() == 1 | true |
| 12 | BeanComment对象，user=“12345@qq.com” | ps.getUpdateCount() != 1 | ps.getUpdateCount() != 1 | false |

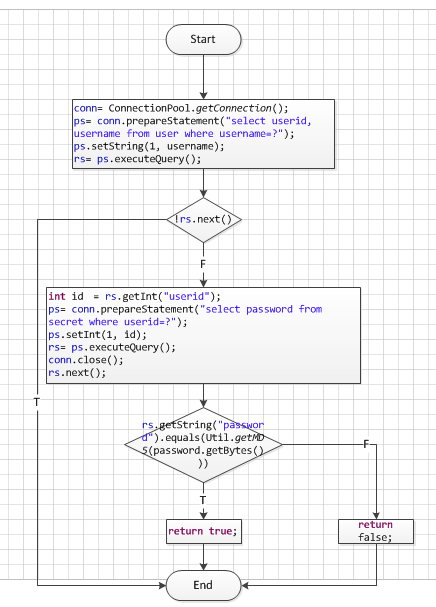
1. 测试类: DaoUser

测试方法： 判定/条件法

被测方法：

* 1. validateUser(String username, String password)

流程图：

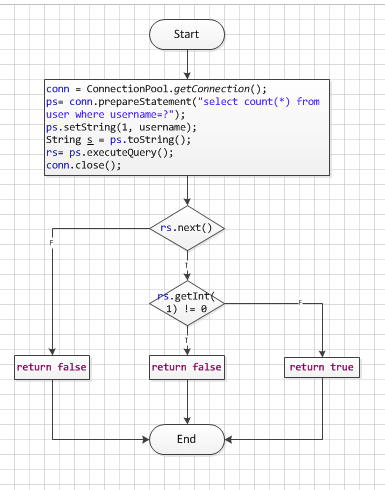


测试用例：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用例ID | 输入 | | 覆盖判定 | 覆盖条件 | 预期结果 |
| username | password |
| 13 | 641567179@qq.com | 123456 | T1,T2 | T1,T2 | true |
| 14 | 123456@qq.com | 112233 | F1-- | F1-- | false |
| 15 | 641567179@qq.com | 000000 | T1F2 | T1F2 | false |

* 1. validateUserName(String username)

流程图：



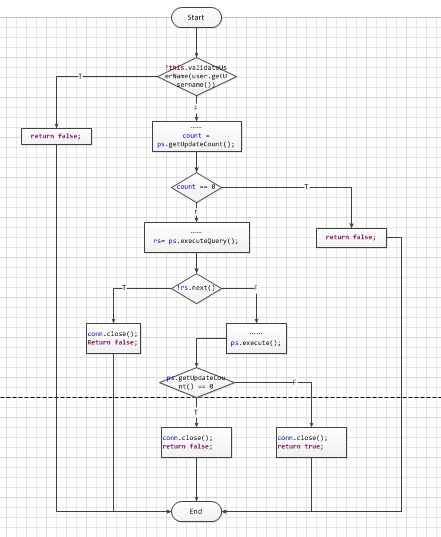
测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 输入 | 覆盖判定 | 覆盖条件 | 预期结果 |
| username |
| 16 | 641567179@qq.com | T1，T2 | T1,T2 | false |
| 17 | 12345@qq.com | T1，F2 | T1，F2 | true |

第一个条件的false执行不到

* 1. addUser(BeanUser user)

流程图：



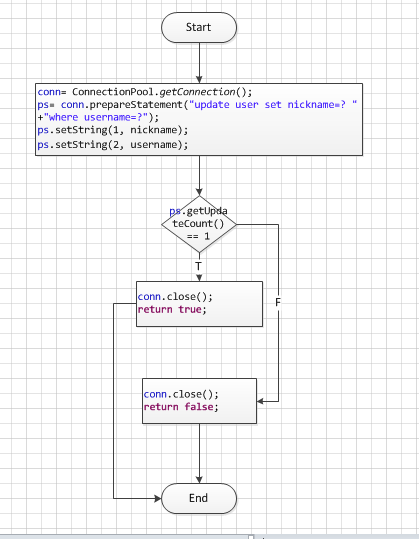
测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 输入 | 覆盖判定 | 覆盖条件 | 预期结果 |
| user |
| 18 | 一个beanuser对象名字为641567179@qq.com | T1， | T1 | false |
| 19 | 一个空对象 | F1，T1 | F1，T1 | false |

第三个，第四个判定现在无法测试

* 1. updateNickname(String username, String nickname)

流程图：

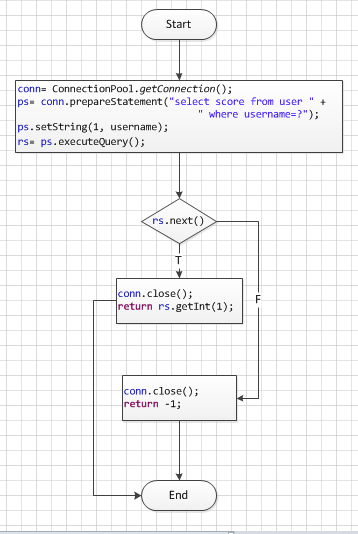


测试用例：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用例ID | 输入 | | 覆盖判定 | 覆盖条件 | 预期结果 |
| username | nickname |
| 20 | 641567179@  qq.com | yusen | ps.getUpdateCount() == 1 | ps.getUpdateCount() == 1 | true |
| 21 | 12345@  qq.com | ssade | ps.getUpdateCount() != 1 | ps.getUpdateCount() != 1 | false |

* 1. getScore(String username)

流程图：

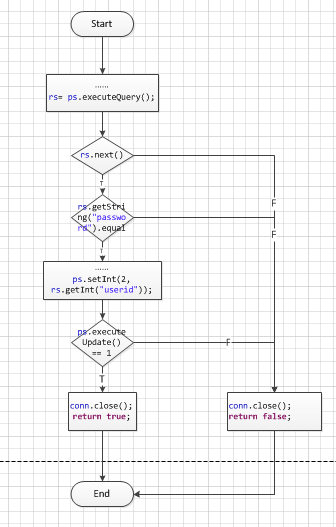


测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 输入 | 覆盖判定 | 覆盖条件 | 预期结果 |
| username |
| 22 | 641567179@qq.com | T1 | T1 | 一个整数 |
| 23 | 12345@qq.com | F1 | F1 | -1 |

* 1. updatePassword(String username, String password, String oldPassword)

流程图：



测试用例：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 用例ID | 输入 | | | 覆盖判定 | 覆盖条件 | 预期结果 |
| username | password | oldPassword |
| 24 | 12345qq.com | 135454 | 54768768 | F1 | F1 | false |
| 25 | 641567179@qq.com | 123456 | 23456789 | T1,F2 | T1,F2 | false |
| 26 | 641567179@qq.com | 为空 | 123456 | T1,T2,F3 | T1,T2,F3 | false |
| 27 | 641567179@qq.com | 1234567 | 123456 | T1,T2,T3 | T1,T2,T3 | true |

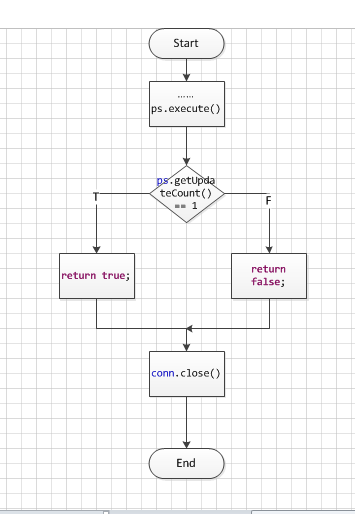
1. 测试类: DaoAddress

测试方法： 判定/条件法

被测方法：

* 1. addAddress(BeanAddress addr)

流程图：

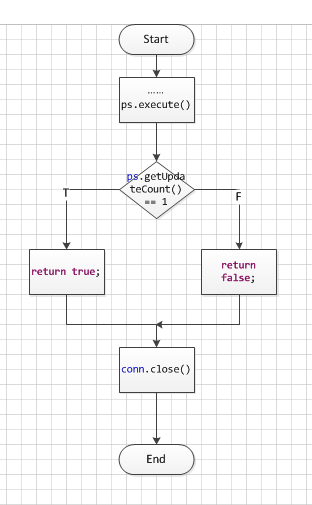


测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 输入 | 覆盖判定 | 覆盖条件 | 预期结果 |
| BeanAddress |
| 28 | 一个有详细内容的Adress对象 | T1 | T1 | true |
| 29 | Adress对象，userID为空 | F1 | F1 | false |

* 1. deleteAddress(int addrID)

流程图：

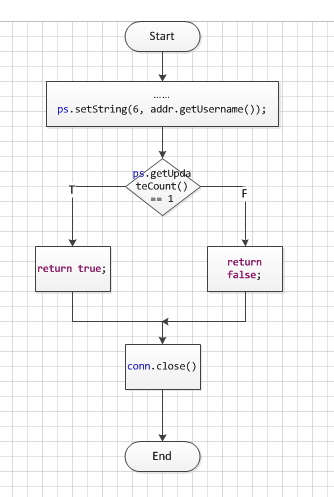


测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 输入 | 覆盖判定 | 覆盖条件 | 预期结果 |
| addrID |
| 30 | 1 | T1 | T1 | true |
| 31 | 0 | F1 | F1 | false |

* 1. modifyAddress(BeanAddress addr)

流程图:

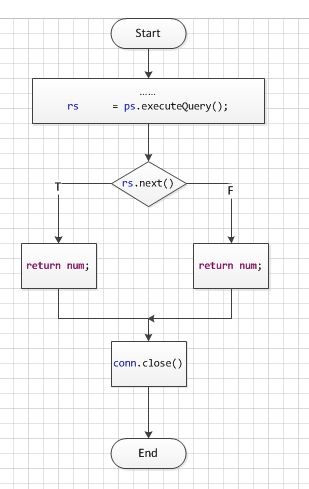


测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 输入 | 覆盖判定 | 覆盖条件 | 预期结果 |
| BeanAddress |
| 32 | 一个Address对象,usename在地址表中存在（23） | T1 | T1 | true |
| 33 | 一个Address对象,usename在地址表中不存在（1024） | F1 | F1 | false |

* 1. getAddressNumber(String username)

流程图：

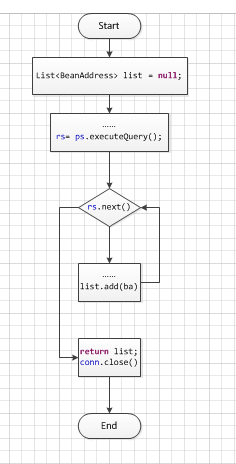


测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 输入 | 覆盖判定 | 覆盖条件 | 预期结果 |
| username |
| 34 | “641567179@qq.com” | T1 | T1 | 1 |
| 35 | “12345qq.com” | F1 | F1 | 0 |

* 1. listAddress(String username)

流程图：

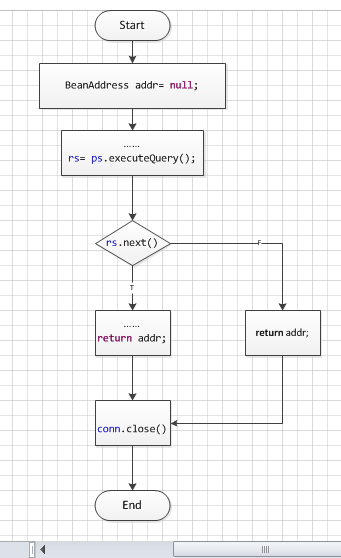


测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 输入 | 覆盖判定 | 覆盖条件 | 预期结果 |
| username |
| 36 | “641567179@qq.com” | T1 | T1 | Not null |
| 37 | “1234q22.com” | F1 | F1 | Null |

* 1. getAddressDetail(String username, int addrid)

流程图：



测试用例：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用例ID | 输入 | | 覆盖判定 | 覆盖条件 | 预期结果 |
| username | addrid |
| 38 | “641567179@qq.com” | 1 | T1 | T1 | Not null |
| 39 | “641567179@qq.com” | 1234 | F1 | F1 | null |
| 40 | “12345@qq.com” | 1 | F1 | F1 | null |

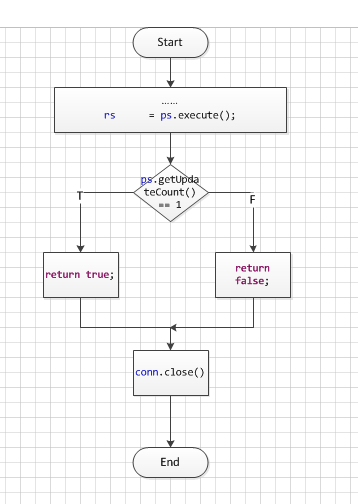
1. 测试类: DaoCart

测试方法： 判定/条件法

被测方法:

* 1. addBook(String username, int bookID, int bookNumber)

流程图：

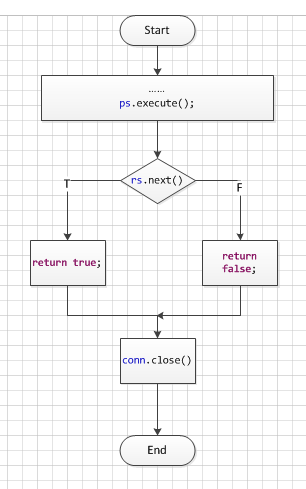


测试用例：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 用例ID | 输入 | | | | 覆盖判定 | 覆盖条件 | 预期结果 |
| username | bookID | bookNumber | |
| 41 | “641567179@qq.com” | 1 | 12 | | T1 | T1 | Not null |
| 42 | “1234qq.com” | 1 | | 12 | F1 | F1 | Null |
| 43 | “641567179@qq.com” | 2355 | | 12 | F1 | F1 | Null |
| 44 | “641567179@qq.com” | 1 | | 0 | F1 | F1 | Null |

* 1. deleteBook(String username, int bookID)

流程图：

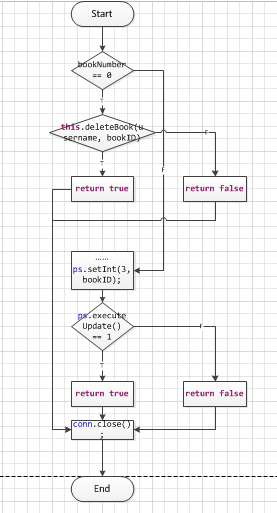


测试用例：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用例ID | 输入 | | 覆盖判定 | 覆盖条件 | 预期结果 |
| username | bookID |
| 45 | “641567179@qq.com” | 1 | T1 | T1 | True |
| 46 | “641567179@qq.com” | 1234 | F1 | F1 | False |
| 47 | “12345@qq.com” | 1 | F1 | F1 | False |

* 1. modifyBook(String username, int bookID, int bookNumber)

流程图：

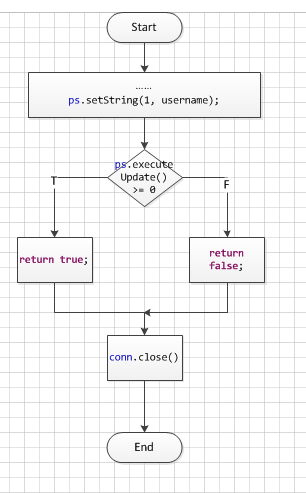


测试用例：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 用例ID | 输入 | | | 覆盖判定 | 覆盖条件 | 预期结果 |
| username | bookID | bookNumber |
| 48 | “641567179@qq.com” | 2 | 0 | T1,T2 | T1,T2 | true |
| 49 | “641567179@qq.com” | 3 | 0 | T1,F2 | T1,F2 | false |
| 50 | “641567179@qq.com” | 1 | 3 | F1,T3 | F1,T3 | true |
| 51 | “1234@qq.com” | 2 | 23 | F1,F3 | F1,F3 | false |

* 1. clean(String username)

流程图：

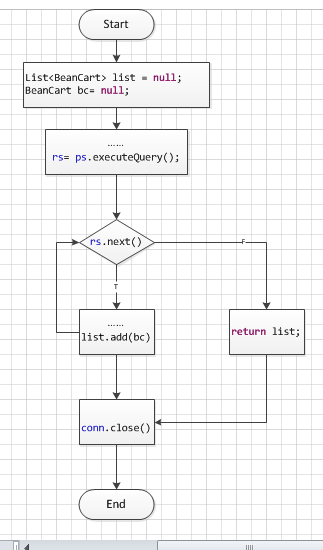


测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 输入 | 覆盖判定 | 覆盖条件 | 预期结果 |
| username |
| 52 | “641567179@qq.com” | T1 | T1 | True |
| 53 | “1234qq.com” | F1 | F1 | False |

* 1. getBooks(String username)

流程图：



测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 输入 | 覆盖判定 | 覆盖条件 | 预期结果 |
| username |
| 54 | “641567179@qq.com” | T1 | T1 | Not null |
| 55 | “1234qq.com” | F1 | F1 | Null |

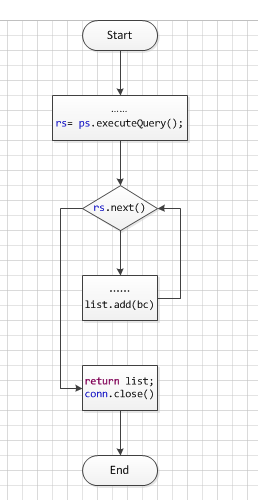
1. 测试类: DaoCatalog

测试方法： 判定/条件法

被测方法:

* 1. getCatalog()

流程图：

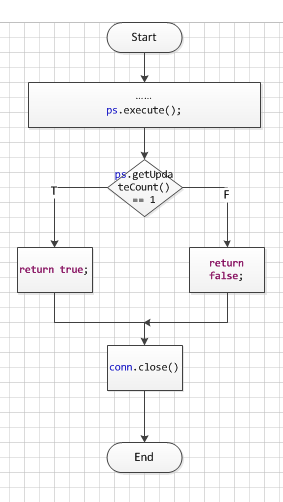


测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 输入 | 覆盖判定 | 覆盖条件 | 预期结果 |
| 有无目录 |
| 56 | 无 | F1 | F1 | Null |
| 57 | 有 | T1 | T1 | Not Null |

* 1. addCatalog(BeanCatalog catalog)

流程图：

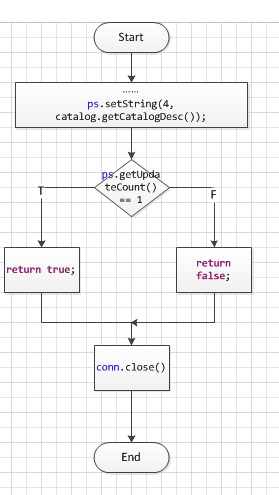


测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 输入 | 覆盖判定 | 覆盖条件 | 预期结果 |
| catalog |
| 58 | Beancatalog对象（非空） | T1 | T1 | True |
| 59 | Beancatalog对象（name为空） | F1 | F1 | False |

* 1. modifyCatalog(BeanCatalog catalog)

流程图：



测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 输入 | 覆盖判定 | 覆盖条件 | 预期结果 |
| catalog |
| 60 | Beancatalog对象（非空） | T1 | T1 | True |
| 61 | Beancatalog对象（name为空） | F1 | F1 | False |

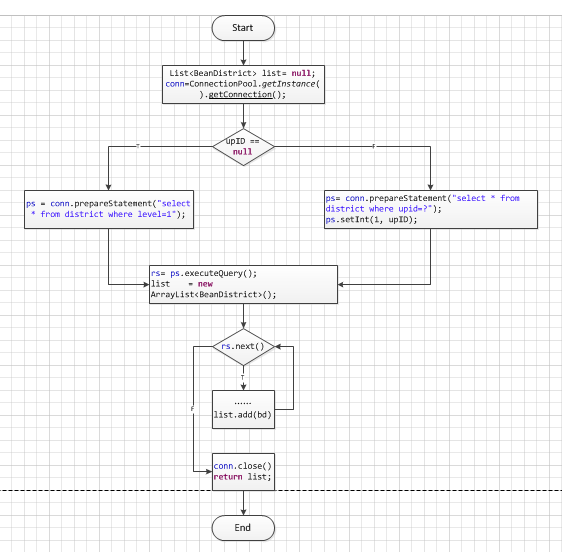
1. 测试类: DaoDistrict

测试方法： 判定/条件法

被测方法:

* 1. getDistrict(Integer upID)

流程图：



测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 输入 | 覆盖判定 | 覆盖条件 | 预期结果 |
| upID |
| 62 | Null,地址为空 | T1，F2 | T1，F2 | null |
| 63 | 1，地址为空 | F1,F2 | F1,F2 | null |
| 64 | Null，地址不为空 | T1，T2 | T1，T2 | Not null |
| 65 | 1，地址不为空 | F1，T2 | F1，T2 | Not null |

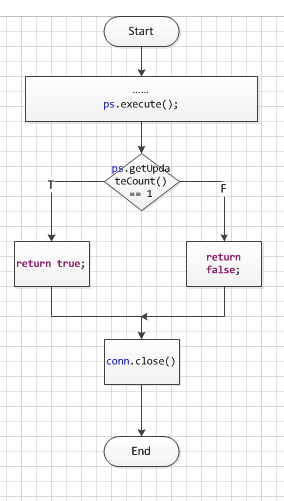
1. 测试类: DaoLevel

测试方法： 判定/条件法

被测方法:

* 1. add(BeanLevel level)

流程图：

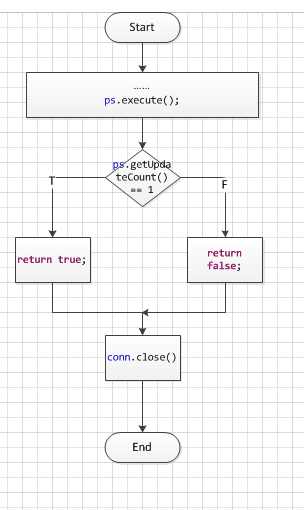


测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 输入 | 覆盖判定 | 覆盖条件 | 预期结果 |
| level |
| 66 | Beanlevel对象，非空 | T1 | T1 | True |
| 67 | Beanlevel对象，name为空 | F1 | F1 | False |

* 1. update(BeanLevel level)

流程图：

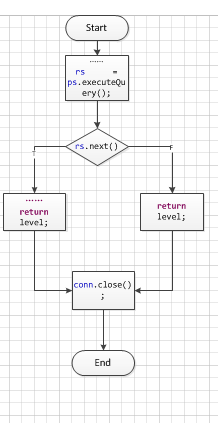


测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 输入 | 覆盖判定 | 覆盖条件 | 预期结果 |
| level |
| 68 | Beanlevel对象，非空 | T1 | T1 | True |
| 69 | Beanlevel对象，name为空 | F1 | F1 | False |

* 1. judgeLevel(**int** score)

流程图：



测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 输入 | 覆盖判定 | 覆盖条件 | 预期结果 |
| Score |
| 70 | 100 | T1 | T1 | Beanlevel对象，level为1 |
| 71 | -100 | F1 | F1 | Null |

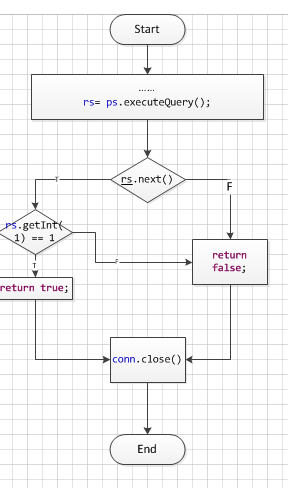
1. 测试类: DaoAdmin

测试方法： 判定/条件法

被测方法：

* 1. validateAdmin(String adminName, String pwd)

流程图：



测试用例：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用例ID | 输入 | | 覆盖判定 | 覆盖条件 | 预期结果 |
| adminName | pwd |
| 72 | admin | admin | T1，T2 | T1，T2 | True |
| 73 | Root | Root | F1 | F1 | False |
| 74 | Admin（表中有多项） | admin | T1，F2 | T1，F2 | False |

用例74无法测试（数据库限定）

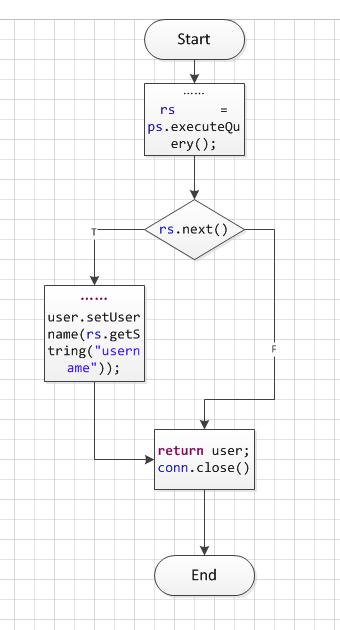
1. 测试类: DaoUser （对DaoUser新增方法的测试）

测试方法： 判定/条件法

被测方法：

* 1. getUser(String username)

流程图：



测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 输入 | 覆盖判定 | 覆盖条件 | 预期结果 |
| username |
| 75 | “641567179@qq.com” | T1 | T1 | Not null |
| 76 | “12345@qq.com” | F1 | F1 | Null |

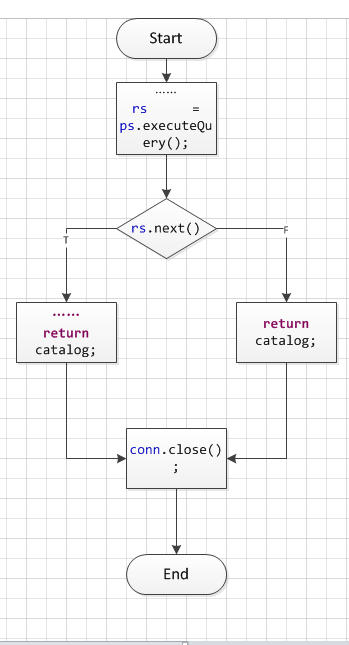
1. 测试类: DaoCatalog(对DaoCatalog新增方法进行测试)

测试方法： 判定/条件法

被测方法:

* 1. getCatalogByID(**int** catalogID)

流程图：



测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 输入 | 覆盖判定 | 覆盖条件 | 预期结果 |
| catalogID |
| 77 | 1 | T1 | T1 | Not null |
| 78 | 123456 | F1 | F1 | Null |

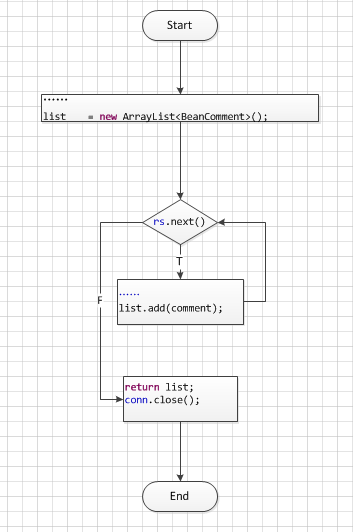
1. 测试类: DaoComment （其中getCommnetNumber(**int** bookID)和publishComment(BeanComment comment)在原来的DaoBook中测试过）

测试方法： 判定/条件法

被测方法:

* 1. getAllComment(**int** page, **int** lines)

流程图：

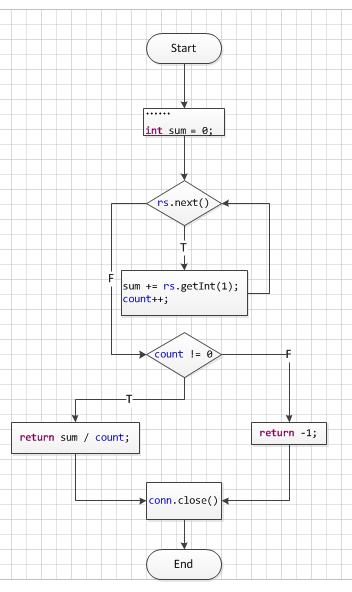


测试用例：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用例ID | 输入 | | 覆盖判定 | 覆盖条件 | 预期结果 |
| page | lines |
| 79 | 1 | 3 | T1， | T1 | Not null |
| 80 | 1 | 0 | F1 | F1 | Null |

* 1. getAverageRating(**int** bookID)

流程图：



测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 输入 | 覆盖判定 | 覆盖条件 | 预期结果 |
| bookID |
| 81 | 1000 | F1，F2， | F1，F2， | -1 |
| 82 | 11 | T1，T2 | T1，T2 | 3 |

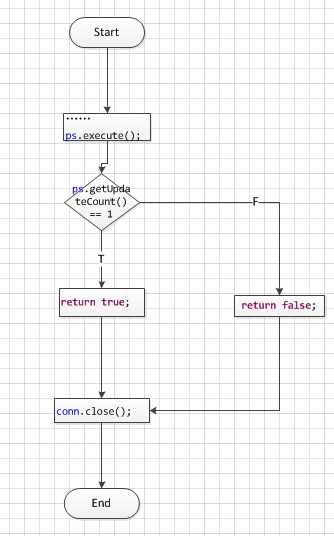
1. 测试类: DaoFavorityFolder

测试方法： 判定/条件法

被测方法:

* 1. add(String username, int bookID)

流程图：

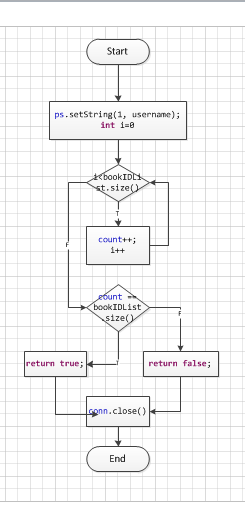


测试用例：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用例ID | 输入 | | 覆盖判定 | 覆盖条件 | 预期结果 |
| username | bookID |
| 83 | “641567179@qq.com” | 3 | T1， | T1 | true |
| 84 | “1234@qq.com” | 3 | F1 | F1 | false |
| 85 | “641567179@qq.com” | 0 | F1 | F1 | false |

* 1. delete(String username, List<Integer>bookIDList)

流程图：

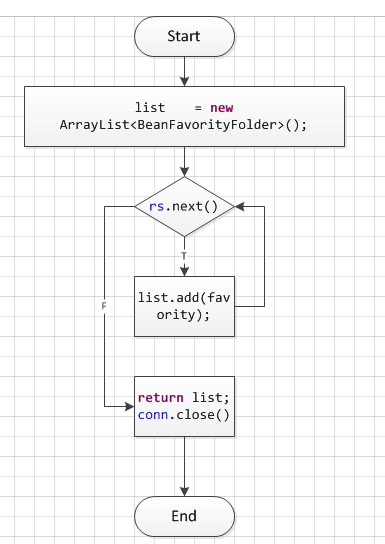


测试用例：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用例ID | 输入 | | 覆盖判定 | 覆盖条件 | 预期结果 |
| username | bookIDList |
| 86 | “641567179@qq.com” | List为空 | F1，F2 | F1，F2 | False |
| 87 | “641567179@qq.com” | List不为空，bookID有效 | T1，T2 | T1，T2 | True |
| 88 | “641567179@qq.com” | List不为空，bookID无效 | T1，F2 | T1，F2 | False |
| 89 | “12345@qq.com” | List不为空，bookID有效 | T1，F2 | T1，F2 | False |

* 1. get(String username, int page, int lines)

流程图：



测试用例：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 用例ID | 输入 | | | 覆盖判定 | 覆盖条件 | 预期结果 |
| username | page | lines |
| 90 | “641567179@qq.com” | 1 | 3 | T1 | T1 | Not null |
| 91 | “641567179@qq.com” | 1 | 0 | F1 | F1 | null |
| 92 | “12345@qq.com” | 1 | 3 | F1 | F1 | null |

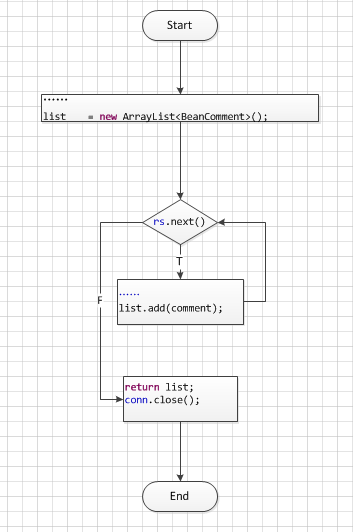
1. 测试类: DaoComment（对getComment(int, int, int)的补充测试）

测试方法： 判定/条件法

被测方法：

1. getComment(int, int, int)

流程图：



测试用例：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 用例ID | 输入 | | | 覆盖判定 | 覆盖条件 | 预期结果 |
| bookID | page | lines |
| 93 | 11 | 1 | 3 | T1 | T1 | Not null |
| 94 | 11 | 1 | 0 | F1 | F1 | Null |
| 95 | 30 | 1 | 3 | F1 | F1 | Null |

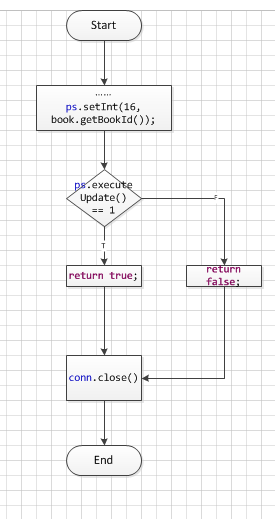
1. 测试类: DaoBook（对DaoBook的补充测试）

测试方法： 判定/条件法

被测方法：

* 1. modifyBook(BeanBook book)

流程图：

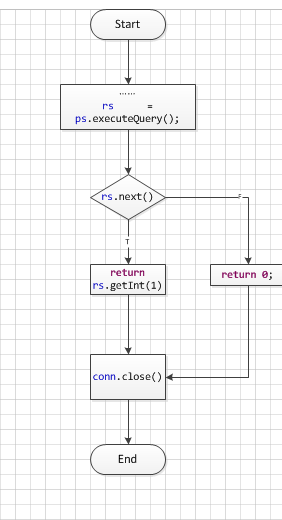


测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 输入 | 覆盖判定 | 覆盖条件 | 预期结果 |
| BeanBook |
| 96 | 一个非空的BeanBook | T1 | T1 | True |
| 97 | BeanBook（空） | F1 | F1 | False |

* 1. getProductNum()

流程图：

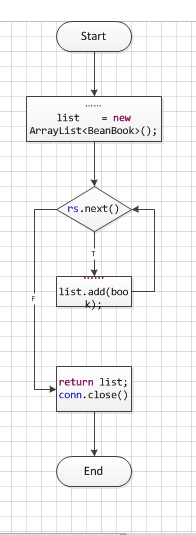


测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 输入 | 覆盖判定 | 覆盖条件 | 预期结果 |
| 无 |
| 98 | Book有值 | T1 | T1 | 30 |
| 99 | Book无值 | F1 | F1 | 0 |

* 1. getLatestBook()

流程图：

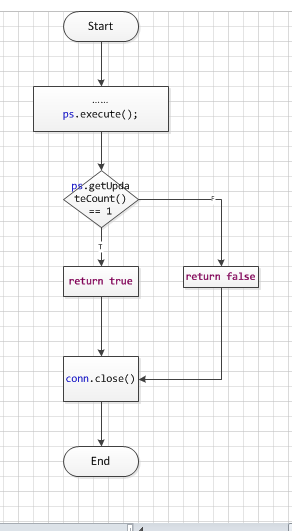


测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 输入 | 覆盖判定 | 覆盖条件 | 预期结果 |
| 无 |
| 100 | Book有值 | T1 | T1 | Not null |
| 101 | Book无值 | F1 | F1 | Null |
| 102 | Book有值只足1个 | T1 | T1 | Not null |

* 1. deleteBook(int)

流程图：



测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 输入 | 覆盖判定 | 覆盖条件 | 预期结果 |
| bookid |
| 103 | 29 | T1 | T1 | True |
| 104 | 50 | F1 | F1 | False |

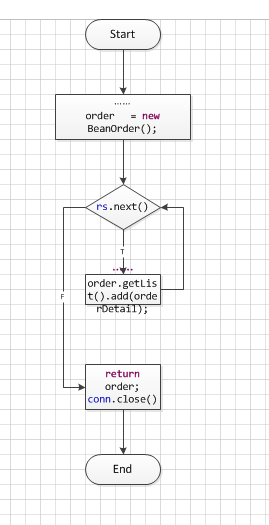
1. 测试类：DaoOrder

测试方法：判定/条件

被测方法：

* 1. getOrderDetail(int)

流程图：

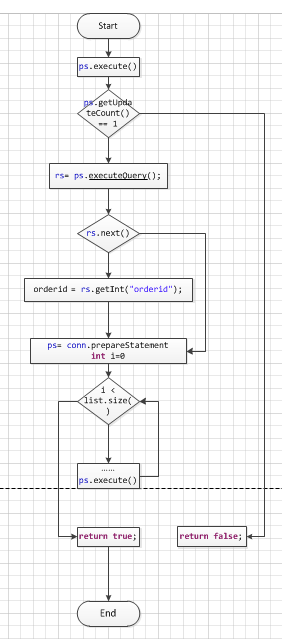


测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 输入 | 覆盖判定 | 覆盖条件 | 预期结果 |
| orderID |
| 103 | 1 | T1 | T1 | BeanOrder中有book |
| 104 | 100 | F1 | F1 | BeanOrder中没有book |

* 1. addOrder(BeanOrder)

流程图：

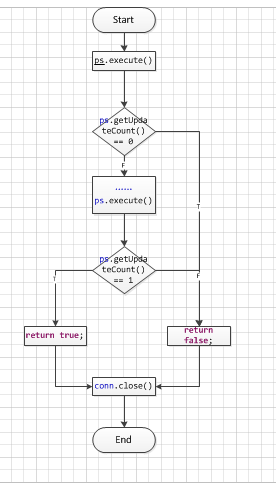


测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 输入 | 覆盖判定 | 覆盖条件 | 预期结果 |
| BeanOrder |
| 105 | BeanOrder对象，username为空 | F1 | F1 | False |
| 106 | BeanOrder对象，但list为空 | T1 | T1 | True |
| 107 | BeanOrder对象，都不为空 | T1 | T1 | True |

* 1. deleteOrder(int)

流程图：



测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 输入 | 覆盖判定 | 覆盖条件 | 预期结果 |
| orderID |
| 109 | 25 | F1 | F1 | False |
| 110 | 6 | T1 | T1 | True |

* 1. getTotalSaleAmount()

流程图：



测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 输入 | 覆盖判定 | 覆盖条件 | 预期结果 |
| 无 |
| 111 | 无数据 | F1 | F1 | -1 |
| 112 | 有数据 | T1 | T1 | 26 |

* 1. getTotalSalesRevenue()

流程图：



测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 输入 | 覆盖判定 | 覆盖条件 | 预期结果 |
| 无 |
| 113 | 无数据 | F1 | F1 | -1 |
| 114 | 有数据 | T1 | T1 | 234 |

* 1. getTotalOrderAmount()

流程图：



测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 输入 | 覆盖判定 | 覆盖条件 | 预期结果 |
| 无 |
| 115 | 无数据 | F1 | F1 | -1 |
| 116 | 有数据 | T1 | T1 | 10 |

* 1. getUnauditedOrderAmount()

流程图:



测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 输入 | 覆盖判定 | 覆盖条件 | 预期结果 |
| 无 |
| 117 | 无数据 | F1 | F1 | -1 |
| 118 | 有数据 | T1 | T1 | 0 |

* 1. getFinishOrderAmount()

流程图：



测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 输入 | 覆盖判定 | 覆盖条件 | 预期结果 |
| 无 |
| 119 | 无数据 | F1 | F1 | -1 |
| 120 | 有数据 | T1 | T1 | 2 |

* 1. getLatestOrders()

流程图：



测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 输入 | 覆盖判定 | 覆盖条件 | 预期结果 |
| 无 |
| 121 | 无数据 | F1 | F1 | Null |
| 122 | 有数据 | T1 | T1 | Not null |

* 1. getItemOrders(int, int, int)

流程图：



测试用例：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 用例ID | 输入 | | | 覆盖判定 | 覆盖条件 | 预期结果 |
| bookID | page | lines |
| 123 | 29 | 1 | 3 | F1 | F1 | Null |
| 124 | 1 | 1 | 3 | T1 | T1 | Not null |

* 1. search(OrderRequire req, String username)

流程图：



测试用例：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用例ID | 输入 | | 覆盖条件 | 覆盖判定 | 预期结果 |
| OrderRequire | username |
| 125 | 有值的对象 | “641567179@qq.com” | T1,T2,T3,T4 | T1,T2,T3,T4 | Not null |
| 126 | 对象默认值值，startTime=2012/9/9 | null | F1,F2,F3,F4 | F1,F2,F3,F4 | null |
| 127 | 有值的对象 | null | F1,T2,T3,T4 | F1,T2,T3,T4 | Not null |

* 1. getRowNumber(OrderRequire req, String username)

流程图：



测试用例：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用例ID | 输入 | | 覆盖条件 | 覆盖判定 | 预期结果 |
| OrderRequire | username |
| 128 | 有值的对象 | “641567179@qq.com” | T1,T2,T3,T4 | T1,T2,T3,T4 | 2 |
| 129 | 对象默认值值，startTime=2012/9/9 | null | F1,F2,F3,F4 | F1,F2,F3,F4 | -1 |
| 130 | 有值的对象 | null | F1,T2,T3,T4 | F1,T2,T3,T4 | 4 |

1. 测试类 DaoBook(附件测试)

测试方法：判定/条件

被测方法：

* 1. search(int sign, BookRequire req)

流程图：



测试用例：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用例ID | 输入 | | 覆盖条件 | 覆盖判定 | 预期结果 |
| sign | BookRequire |
| 131 | 0 | 对象有值 | T1,T10 | T1,T10 | Not Null |
| 132 | 0 | 对象无值 | T1，F10 | T1，F10 | Null |
| 133 | 1 | 对象有值,beforeyear=true，year有值 | F1,T2..T9,T10 | F1,T2..T9,T10 | Not Null |
| 134 | 1 | 对象有值,beforeyear=true，year无值 | F1,F2.T3.T9,T10 | F1,T2..T9,T10 | Not Null |
| 134 | 1 | 对象无值 | F1,F2…F9,F10 | F1,F2…F9,F10 | Null |

* 1. getSearchRowNumber(int sign, BookRequire req)

流程图：



测试用例：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用例ID | 输入 | | 覆盖条件 | 覆盖判定 | 预期结果 |
| sign | BookRequire |
| 135 | 0 | 对象有值 | T1,T10 | T1,T10 | 1 |
| 136 | 0 | 对象无值 | T1，F10 | T1，F10 | 3 |
| 137 | 1 | 对象有值, | F1,T2..T9,T10 | F1,T2..T9,T10 | 1 |
| 138 | 1 | 对象无值 | F1,F2…F9,F10 | F1,F2…F9,F10 | 3 |
| 139 | 1 | Book不存在 | F1,F2…F9,F10 | F1,F2…F9,F10 | 0 |

* 1. getBookStatistics(BookStatisticsRequire req)

流程图：



测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 输入 | 覆盖判定 | 覆盖条件 | 预期结果 |
| BookStatisticsRequire |
| 140 | catalogid =null, lowlimit = 0 | T1,T2,F3 | T1,T2,F3 | Not null |
| 141 | catalogid =null, lowlimit =1000 | T1,T2,F3 | T1,T2,F3 | null |
| 142 | catalogid !=null, lowlimit = 0 | F1,F2,T3 | F1,F2,T3 | Not null |
| 143 | catalogid !=null, lowlimit =1000 | F1,F2,T3 | F1,F2,T3 | null |
| 144 | catalogid !=null, lowlimit =0  startTime=2011-01-01 | F1,F2,T3 | F1,F2,T3 | Not null |
| 145 | catalogid !=null, lowlimit = 0  startTime=2013-01-01 | T1,T2,F3 | T1,T2,F3 | Not null |
| 146 | catalogid !=null, 但不存在lowlimit =1000 | F1,F2,T3 | F1,F2,T3 | Null |

1. 测试类 DaoUser

测试方法：判定/条件

被测方法：

* 1. getUserNum

流程图：



测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 输入 | 覆盖判定 | 覆盖条件 | 预期结果 |
| 无 |
| 147 | 数据库有数据 | T1 | T1 | 10 |

即使数据库无数据，得到的应该是0，不可能是-1，一般情况不会出现-1，有关-1的测试取消。

* 1. search(UserRequire req)

流程图：



测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 输入 | 覆盖判定 | 覆盖条件 | 预期结果 |
| UserRequire |
| 148 | 对象有值 | T1,T2,T3,T4 | T1,T2,T3,T4 | Not null |
| 149 | 对象有值，usename不存在 | T1,T2,T3,F4 | T1,T2,T3,F4 | null |
| 150 | 对象为默认值 | F1,F2,F3,T4 | F1,F2,F3,T4 | Not null |

* 1. getSearchNum(UserRequire req)

流程图：



测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 输入 | 覆盖判定 | 覆盖条件 | 预期结果 |
| UserRequire |
| 151 | 对象有值 | T1,T2,T3,T4 | T1,T2,T3,T4 | 1 |
| 152 | 对象有值，usename不存在 | T1,T2,T3,T4 | T1,T2,T3,T4 | 0 |
| 153 | 对象为默认值 | F1,F2,F3,T4 | F1,F2,F3,T4 | 10 |

即使数据库无数据，得到的应该是0，不可能是-1，一般情况不会出现-1，有关-1的测试取消。

* 1. updateScore(int score, String username)

流程图：



测试用例：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用例ID | 输入 | | 覆盖判定 | 覆盖条件 | 预期结果 |
| score | username |
| 154 | 10000 | 544921965@qq.com | T1 | T1 | True |
| 155 | 10000 | 123456@qq.com | F1 | F1 | False |

* 1. updateQA(String username, String question, String answer)

流程图：



测试用例：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 用例ID | 输入 | | | | 覆盖判定 | 覆盖条件 | 预期结果 |
| username | question | | answer |
| 154 | 544921965@qq.com | “我是谁” | 刘光宗 | | T1 | T1 | True |
| 155 | 123456@qq.com | “我是谁” | 刘光宗 | | F1 | F1 | False |