测试用例列表：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试用例ID | 测试用例描述 | 输入数据 | 预期输出 | 实际输出 | Pass/Fail | 备注 |
| RepositoryDataTest-001 | RepositoryData中方法的测试 |  |  |  |  |  |
| UserDataTest-001 | UserData中方法的测试 |  |  |  |  |  |
| RepositoryTest-001 | Repository中方法的测试 |  |  |  |  |  |
| StatisticsTest-001 | Statistics中方法的测试 |  |  |  |  |  |
| UserTest-001 | User中方法的测试 |  |  |  |  |  |

相关的Junit源码文件：

RepositoryDataTest类：

/\*测试CheckRepository方法\*/

**public** **void** testCheckRepository() **throws** IOException{

*assertTrue*(po1.getName().equals(*testRepository*.checkRepository("mojombo","grit").getName()));

*assertTrue*(po2.getName().equals(*testRepository*.checkRepository("mojombo","god").getName()));

*assertTrue*(po3.getName().equals(*testRepository*.checkRepository("macournoyer","thin").getName()));

}

/\*测试getRepositoriesNames方法\*/

**public** **void** testGetRepositoriesNames() **throws** IOException{

ArrayList<String> a=*testRepository*.getRepositoriesNames();

*assertEquals*(po1.getName(), a.get(0));

*assertEquals*(po2.getName(), a.get(1));

*assertEquals*(po3.getName(), a.get(4));

}

/\*测试languagesOfRepository方法\*/

**public** **void** testLanguagesOfRepository() **throws** IOException{

*assertTrue*(po1.getLanguage().equals(*testRepository*.checkRepository("mojombo","grit").getLanguage()));

*assertTrue*(po2.getLanguage().equals(*testRepository*.checkRepository("mojombo","god").getLanguage()));

*assertTrue*(po3.getLanguage().equals(*testRepository*.checkRepository("macournoyer","thin").getLanguage()));

}

/\*测试search方法\*/

**public** **void** testSearch() **throws** IOException{

*assertEquals*(po1.getName(), *testRepository*.Search("grit").get(0).getName());

*assertEquals*(po2.getDescription(),*testRepository*.Search("god").get(0).getDescription());

*assertEquals*(po3.getForks(), *testRepository*.Search("thin").get(0).getForks());

}

RepoSortTest类:

/\*测试sort方法\*/

**public** **void** testSort() **throws** IOException{

System.***out***.println(po1.getCreated());

*assertEquals*(po1.getCreated(),*testRepository*.sort(Repository\_Sort.***fork***).get(0).getCreated());

}

UserDataTest类:

**public** **void** testCheckUser(){

*assertTrue*(po1.getLogin().equals("mojombo"));

*assertTrue*(po2.getLogin().equals("technoweenie"));

*assertTrue*(po3.getLogin().equals("macournoyer"));

}

**public** **void** testSearch(){

*assertTrue*(search.get(0).getEmail().equals("tom@mojombo.com"));

*assertTrue*(search.get(1).getEmail().equals("contractfrombelow@gmail.com"));

}

RepositoryTest类:

/\*测试getRepositoriesNames方法\*/

**public** **void** testgetRepositoriesNames() {

Iterator<String> itr=*testRepository*.getRepositoriesNames();

ArrayList<String> str=**new** ArrayList<String>();

**while**(itr.hasNext()){

String s=itr.next();

str.add(s);

}

*assertEquals*(repositoryVOs.get(0).getName(), str.get(0));

*assertEquals*(repositoryVOs.get(1).getName(), str.get(1));

*assertEquals*(repositoryVOs.get(2).getName(), str.get(2));

}

/\*测试checkRepository方法\*/

**public** **void** testcheckRepository() **throws** IOException{

RepositoryVO vo=*testRepository*.checkRepository("mojombo","grit");

*assertEquals*(repositoryVOs.get(0).getName(), vo.getName());

}

/\*测试languagesOfRepository方法\*/

**public** **void** testlanguagesOfRepository() **throws** IOException{

*assertEquals*("Ruby", *testRepository*.languagesOfRepository("mojombo","grit").getStr().get(0));

}

/\*测试Search方法\*/

**public** **void** testSearch() **throws** IOException{

Iterator<RepositoryVO> testSearch=*testRepository*.Search("grit");

ArrayList<RepositoryVO> vos=**new** ArrayList<RepositoryVO>();

**while**(testSearch.hasNext()){

RepositoryVO vo=testSearch.next();

vos.add(vo);

}

*assertTrue*(vos.get(0).getName().equals(repositoryVOs.get(0).getName()));

}

StatisticsTest类:

/\*测试getStar方法\*/

**public** **void** testgetStar() **throws** IOException{

*assertTrue*(*testsStatistics*.getStar().getStr().get(0)!=**null**);

}

/\*测试getForks方法\*/

**public** **void** testgetForks() **throws** IOException{

*assertTrue*(*testsStatistics*.getForks().getStr().get(0)!=**null**);

}

/\*测试getRepoCreated方法\*/

**public** **void** testgetRepoCreated() **throws** IOException{

*assertTrue*(*testsStatistics*.getRepoCreated().getStr().get(0)!=**null**);

}

/\*测试getLanguage方法\*/

**public** **void** testgetLanguage() **throws** IOException{

*assertTrue*(*testsStatistics*.getLanguage().getStr().get(0)!=**null**);

}

/\*测试getUserCreated方法\*/

**public** **void** testgetUserCreated() **throws** IOException{

*assertTrue*(*testsStatistics*.getUserCreated().getStr().get(0)!=**null**);

}

/\*测试getUserType方法\*/

**public** **void** testgetUserType() **throws** IOException{

*assertTrue*(*testsStatistics*.getUserType().getStr().get(0)!=**null**);

}

/\*测试getUserHas方法\*/

**public** **void** testgetUserHas(){

*assertTrue*(*testsStatistics*.getUserHas().getInt().get(0)!=**null**);

}

/\*测试getUserRelated方法\*/

**public** **void** testgetUserRelated(){

*assertTrue*(*testsStatistics*.getUserRelated().getInt().get(0)!=**null**);

}

/\*测试getCompany方法\*/

**public** **void** testgetCompany() **throws** IOException{

*assertTrue*(*testsStatistics*.getCompany().getStr().get(0)!=**null**);

}

/\*测试getCollaborator方法\*/

**public** **void** testgetCollaborator() **throws** IOException{

*assertTrue*(*testsStatistics*.getCollaborator().getStr().get(0)!=**null**);

}

/\*测试getContributor方法\*/

**public** **void** testgetContributor() **throws** IOException{

*assertTrue*(*testsStatistics*.getContributor().getStr().get(0)!=**null**);

}

UserTest类:

/\*测试checkUser方法\*/

**public** **void** testcheckUser() **throws** IOException{

*assertTrue*(*testUser*.checkUser("mojombo").getName().equals("Tom Preston-Werner"));

}

/\*测试UserInfo方法\*/

**public** **void** testUserInfo() **throws** IOException{

*assertTrue*(*testUser*.UserInfo("mojombo", UserInfo.***id***).equals("1"));

}

/\*测试sortUser方法\*/

**public** **void** testsortUser() **throws** IOException{

Iterator<UserVO> test=*testUser*.sortUser(User\_Sort.***Followers***);

ArrayList<UserVO> vos=**new** ArrayList<UserVO>();

**while**(test.hasNext()){

UserVO s=test.next();

vos.add(s);

}

*assertTrue*(vos.get(0).getName().equals("Tom Preston-Werner"));

}

/\*测试screenTime方法\*/

**public** **void** testscreenTime() **throws** IOException{

Iterator<UserVO> test=*testUser*.screenTime("2010");

ArrayList<UserVO> vos=**new** ArrayList<UserVO>();

**while**(test.hasNext()){

UserVO s=test.next();

vos.add(s);

}

*assertTrue*(vos.get(0).getName().equals("Tekkub's WoW addons"));

}