软件复用课程设计

Client-Server 应用程序

测试文档

小组： Team10

成员： 谢志杰 1352975

计鹏玥 1352914

王思尧 1352896

谢明玥 1352937

徐锦程 1353012

文档变更记录

|  |  |  |
| --- | --- | --- |
| 更改人 | 日期 | 更改内容 |
| 王思尧 | 2016年4月20日 | 创建文档 |
| 计鹏玥，谢明玥 | 2016年4月22日 | 完善文档 |
| 谢明玥 | 2016年5月3日 | SaveToFile测试 |

[文档变更记录 1](#_Toc32277)

[1．测试规划 1](#_Toc15944)

[1.1测试人员 1](#_Toc5046)

[1.2测试计划 1](#_Toc22283)

[1.3测试环境和工具 1](#_Toc28959)

[2．测试执行 2](#_Toc10713)

[2.1CM 2](#_Toc4406)

[2.2 PM 3](#_Toc29486)

[2.3 License 4](#_Toc24334)

[2.4 File 6](#_Toc6215)

[3．测试结论 6](#_Toc8058)

1．测试规划

1.1测试人员

|  |  |
| --- | --- |
| 学号 | 姓名 |
| 1352975 | 谢志杰 |
| 1352914 | 计鹏玥 |
| 1352896 | 王思尧 |
| 1352937 | 谢明玥 |
| 1353012 | 徐锦程 |

1.2测试计划

|  |  |  |
| --- | --- | --- |
| 测试时间 | 测试人员 | 测试内容 |
| 2016年3月26日 | 谢志杰 | Server端的测试 |
| 2016年3月29日 | 计鹏玥、谢明玥 | Client端的测试 |
| 2016年3月29日 | 谢志杰 | MessageCount构件的测试 |
| 2016年4月13日 | 徐锦程 | CM |
| 2016年4月13日 | 谢志杰 | PM、License、MessageUtils |
| 2016年4月20日 | 王思尧 | 对各个构件的测试进行整理 |
| 2016年4月30日 | 徐锦程 | 对SaveToFile构件的测试 |

1.3测试环境和工具

GitHub - 版本管理工具

Java - 测试语言

JUnit - 单元测试框架

RabbitMQ - 中间件

2．测试执行

2.1CM

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No. | 功能点 | 输入参数 | 预期结果 | 测试结果 | 正确性 |
| 1 | GetConfigurationInfo | getSERVER\_IP() | 127.0.0.1 | 127.0.0.1 | 正确 |
| getSERVER\_PORT() | 2095 | 2095 |
| GetMAX\_MESSAGE\_PER\_SECOND() | 5 | 5 |
| getMAX\_MESSAGE\_FOR\_TOTAL() | 10 | 10 |
| getDBUSER() | root | root |
| getDBPW() | root | root |
| 2 | WriteJSONFile |  |  |  | 正确 |
| 3 | ReadJSONFile | testoutput.json | {"key1":"value1",  "key2":"value2",  "key3":"value3"} | {"key1":"value1",  "key2":"value2",  "key3":"value3"} | 正确 |
| 4 | GetStringByKey | server\_ip | 127.0.0.1 | 127.0.0.1 | 正确 |
| 5 | GetIntByKey | server\_port | 2095 | 2095 | 正确 |
| 6 | getSERVER\_IP |  | 127.0.0.1 | 127.0.0.1 | 正确 |
| 7 | getSERVER\_PORT |  | 2095 | 2095 | 正确 |
| 8 | getMAX\_MESSAGE  \_PER\_SECOND |  | 5 | 5 | 正确 |
| 9 | getMAX\_MESSAGE  \_FOR\_TOTAL |  | 10 | 10 | 正确 |
| 10 | getDBUSER |  | root | root | 正确 |
| 11 | getDBPW |  | root | root | 正确 |

**测试描述：**

1. 用于测试用户是否可以使用该构件实现对配置文件的读写。

**测试结论：**

1. 测试执行充分。

2. 测试已经完成。

3. 测试已经通过。

4. 可以进入下一阶段任务目标。

2.2 PM

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No. | 功能点 | 输入参数 | 预期结果 | 测试结果 | 正确性 |
| 1 | addCountType | type1  type2  type3 | 1 | 1 | 正确 |
| 2 | addCount | type1  type2  type3 | 1 | 1 | 正确 |
| 3 | getLock |  | java.lang.Object@497470ed | java.lang.Object@497470ed | 正确 |
| 4 | run |  | 20160419\_162432: testType 1 | 20160419\_162432: testType 1 | 正确 |
| 5 | IOLog |  | 20160419\_162826: test | 20160419\_162826: test | 正确 |
| 6 | addCountType  & addCount & commence |  | 20160419\_163718: type1: 0 20160419\_163718: type2: 0 20160419\_163718: type3: 0 20160419\_163818: type1: 0 20160419\_163818: type2: 0 20160419\_163818: type3: 0 20160419\_163918: type1: 1 20160419\_163918: type2: 1 20160419\_163918: type3: 1 20160419\_163938: type1: 0 20160419\_163938: type2: 0 20160419\_163938: type3: 0 20160419\_164038: type1: 0 20160419\_164038: type2: 0 20160419\_164038: type3: 0 | 20160419\_163718: type1: 0 20160419\_163718: type2: 0 20160419\_163718: type3: 0 20160419\_163818: type1: 0 20160419\_163818: type2: 0 20160419\_163818: type3: 0 20160419\_163918: type1: 1 20160419\_163918: type2: 1 20160419\_163918: type3: 1 20160419\_163938: type1: 0 20160419\_163938: type2: 0 20160419\_163938: type3: 0 20160419\_164038: type1: 0 20160419\_164038: type2: 0 20160419\_164038: type3: 0 | 正确 |

**测试描述：**

1. 用于测试用户是否可以根据自己的需要添加所需的计数字段，计数字段一旦添加即不可更改，在写入文件时全部字段会写入日志。
2. 用于测试用户通过实例化一个类，并传入写文件地址、选择是否在文件尾写入作为参数，实例化后调用类中IOWrite的方法，传入写入的字符串作为参数的行为，是否能够完成写操作的功能。
3. 用于测试用户是否可以使用该构件实现对于登录次数的计数功能。

**测试结论：**

1. 测试执行充分。

2. 测试已经完成。

3. 测试已经通过。

4. 可以进入下一阶段任务目标。

2.3 License

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No. | 功能点 | 输入参数 | 预期结果 | 测试结果 | 正确性 |
| 1 | checkMsgInSecond  checkTotalMsg  increaseMsg | =0s | checkMsgInSecond = true  checkTotalMsg = true | checkMsgInSecond = true  checkTotalMsg = true | 正确 |
| <3s | checkMsgInSecond = false  checkTotalMsg = true | checkMsgInSecond = false  checkTotalMsg = true |
| <8s | checkMsgInSecond = false  checkTotalMsg = false | checkMsgInSecond = false  checkTotalMsg = false |
| 2 | reset | =0s | checkMsgInSecond = true  checkTotalMsg = true | checkMsgInSecond = true  checkTotalMsg = true | 正确 |
|  |  | <3s | checkMsgInSecond = false  checkTotalMsg = true | checkMsgInSecond = false  checkTotalMsg = true |  |
| <8s | checkMsgInSecond = false  checkTotalMsg = false | checkMsgInSecond = false  checkTotalMsg = false |
| reset() | checkMsgInSecond = true  checkTotalMsg = true | checkMsgInSecond = true  checkTotalMsg = true |
| 3 | checkMsgInSecond | =0s | checkMsgInSecond = true | checkMsgInSecond = true | 正确 |
| <3s | checkMsgInSecond = false | checkMsgInSecond = false |
| <8s | checkMsgInSecond = false | checkMsgInSecond = false |
| 4 | checkTotalMsg | =0s | checkTotalMsg = true | checkTotalMsg = true | 正确 |
| <3s | checkTotalMsg = true | checkTotalMsg = true |
| <8s | checkTotalMsg = false | checkTotalMsg = false |
| 5 | setPerSec | =0s | checkMsgInSecond = true | checkMsgInSecond = true | 正确 |
| <3s | checkMsgInSecond = false | checkMsgInSecond = false |
| setPerSec(1)  =0s | checkMsgInSecond = false | checkMsgInSecond = false |
| <2s | checkMsgInSecond = false | checkMsgInSecond = false |
| 6 | setMAX | =0s | checkTotalMsg = true | checkTotalMsg = true | 正确 |
| <3s | checkTotalMsg = true | checkTotalMsg = true |
| setMAX(1)  =0s | checkTotalMsg = false | checkTotalMsg = false |
| <2s | checkTotalMsg = false | checkTotalMsg = false |

**测试描述：**

1. 测试用户是否可以使用该构件来设置每分钟消息收发最大数目、登录后收发的消息总数、在计数器规定范围内收发的消息数目等。

2. 测试用户是否可以使用该构件实现对于消息的计数功能，计数包括每分钟消息的计数以及消息总数的计数。

**测试结论：**

1. 测试执行充分。

2. 测试已经完成。

3. 测试已经通过。

4. 可以进入下一阶段任务目标。

2.4 File

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No. | 功能点 | 输入参数 | 预期结果 | 测试结果 | 正确性 |
| 1 | SaveToFile | null | 在运行目录下成功新建“history.txt” | 在运行目录下成功新建“history.txt” | 正确 |
| "D:\\" | 在D盘下成功新建“history.txt” | 在D盘下成功新建“history.txt” |
| 2 | write | "write successfully" | “history.txt”中有"write successfully"和"write successfully again" | “history.txt”中有"write successfully"和"write successfully again" | 正确 |
| "write successfully again" | D盘下的“history.txt”中有"write successfully"和"write successfully again" | D盘下的“history.txt”中有"write successfully"和"write successfully again" | 正确 |

**测试描述：**

1. 测试用户是否可以使用该构件获取文件写入流。

2. 测试用户是否可以使用该构件将收发的消息写入文件

**测试结论：**

1. 测试执行充分。

2. 测试已经完成。

3. 测试已经通过。

4. 可以进入下一阶段任务目标。

3．测试结论

1. 从分析中总结此次设计和执行做的好的地方：

此次设计和执行做的好的地方在于很多功能的测试都使用了多个用例作为测试的输入，可以对部分功能进行较为完整的测试。以及对每个功能都进行了测试，所以保证了所有的功能都得到了测试，降低了缺陷率。

1. 从分析中总结此次设计和执行需要努力的地方：

此次设计和执行需要努力的地方在于构件的开发需要不断的完善，对于可复用构件的开发和使用都需要继续完善。

1. 对此项目的一个质量评价：

总体来说预期的功能点都得到了实现，且在最终的测试过程中都可以正确运行。我们接下来会考虑优化性能，并提高产品的可用性与易用性，也在此基础上提高各个构件的复用程度，为以后其他项目的编写提供便捷渠道。