2019-2020学年 第 一 学期

**实验报告**

课程名称 Python程序设计实验

系（部） 信息工程系

专业班级 电子信息科学与技术18-2

学生姓名 段明宇

学生学号 201823010205

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 实验名称 | | 异常的查找和程序的测试 | | | | |
| 实验目的 | | 熟悉python出现的内置异常并且基本了解测试 | | | | |
| 实验内容 | | 搜索查找11.1.2的异常对11-1例题进行现 | | | | |
| 实验仪器设备 | | Jupyer Notebook、spyder、电脑 | | | | |
| 姓 名 | | 段明宇 | | 组 别 |  |
| 同组实验者 | |  | | 实验日期 | 2019.11.14 |
| 指导教师 | | 魏光村 | |  |  |
| 批阅意见  及 成 绩 | | |  | | | |
| **实**  **验**  **步**  **骤** | 1 实验原理  2 实验过程记录  3 实验结果  4实验过程中存在的问题及解决方案  5 实验总结 | | | | | |

1 实验原理

程序出错是件非常难避 免的事情。再厉害的程序员也无法提前预 见代码运行时可能会遇到的所有情况，几乎每个程序员都被用户说过“你编的那个软件不好用啊”，而程序员经过反复检查以后发现问题的原因是用户操作不规范或者输人了错误类型的数据,于是一边修改代码加强类型检查一边抱怨用户不按套路出牌。 其实呢,作者个人认为这样的问题的根源还是在程序员而不在用户，程序员编写代码时有义务也有必要考虑这些特殊情况，因为大多时候恰恰是少数特殊情况影响了整个系统的美感和开发人员的成就感(二八定律)。虽然大部分软件在发布前一般都经过了严格的测试，然而充分的测试也很难枚举所有可能出现的情况，这时异常处理结构则是避免特殊情况下软件崩溃的利器。

异常是指程序运行时引发的错误,引发错误的原因有很多;如除零、下标越界、文件不存在、网络异常等。如果这些错误得不到正确的处理将会导致程序崩溃并终止运行，合理地使用异常处理结构可以使得程序更加健壮具有更高的容错性，不会因为用户不小心的错误输人而造成程序崩溃，也可以便用异常处理结构为用户提供更地友好的提示。另外，有效的软件测试方法能够在软件发布之前发现尽可能多的bug.而软件发布之后再出现错误时是否能够调试程序并快速定位和解决存在的问题则是程序员综合水平和能力的重要体现。

2 实验过程记录

现在网上搜素各个异常的意思，之后在加以理解，异常内容意思如下：

<https://github.com/KeyNG666/python-study.git>在此网站打开[11.1.2解释异常.docx](https://github.com/KeyNG666/python-study/blob/master/11.1.2%E8%A7%A3%E9%87%8A%E5%BC%82%E5%B8%B8.docx)即为异常解释。

在jupyter中编写11-1例题，演示如何利用unittest库对Stack类中出栈、入栈改变大小等方法来进行测试，并将其写如文件。过程如下：

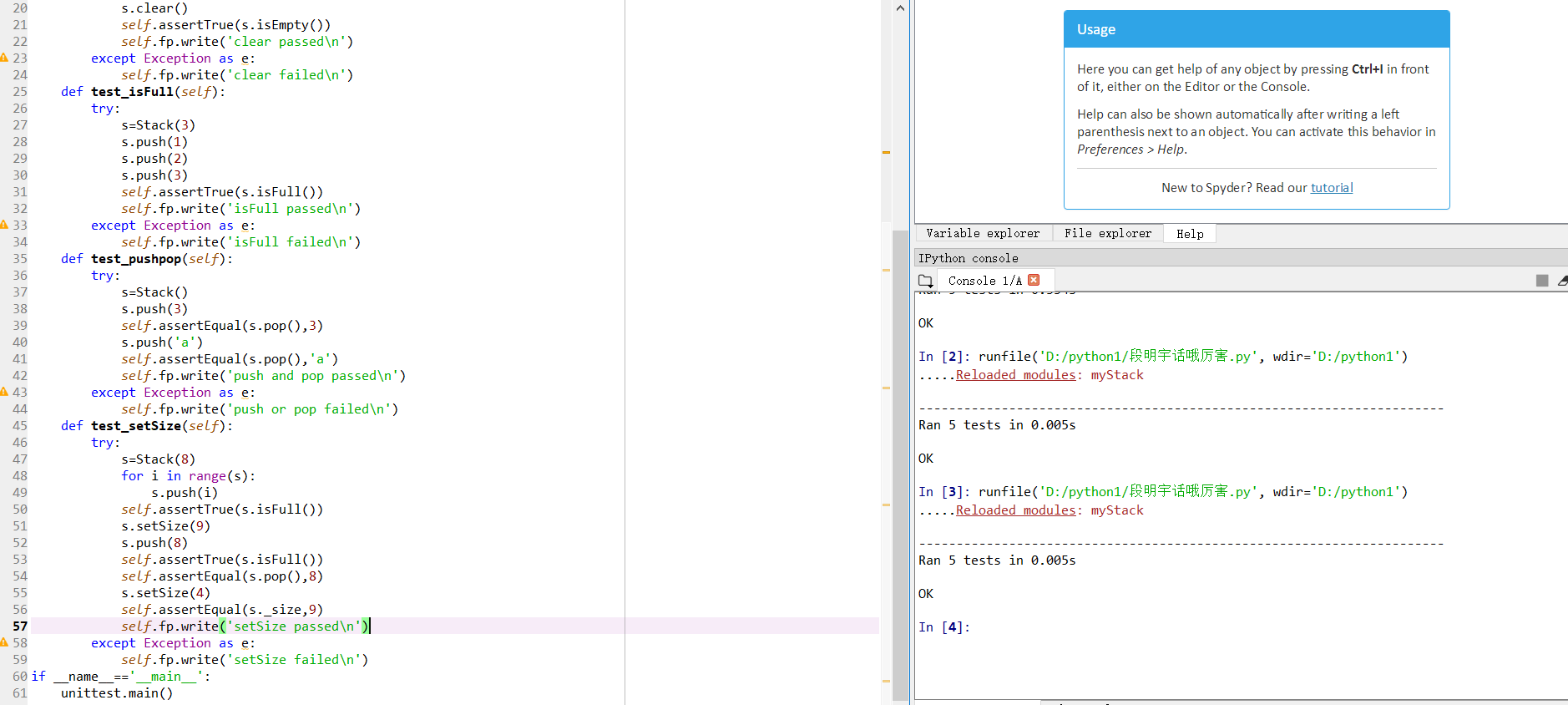


而在Spyder中编写运行出以下结果：

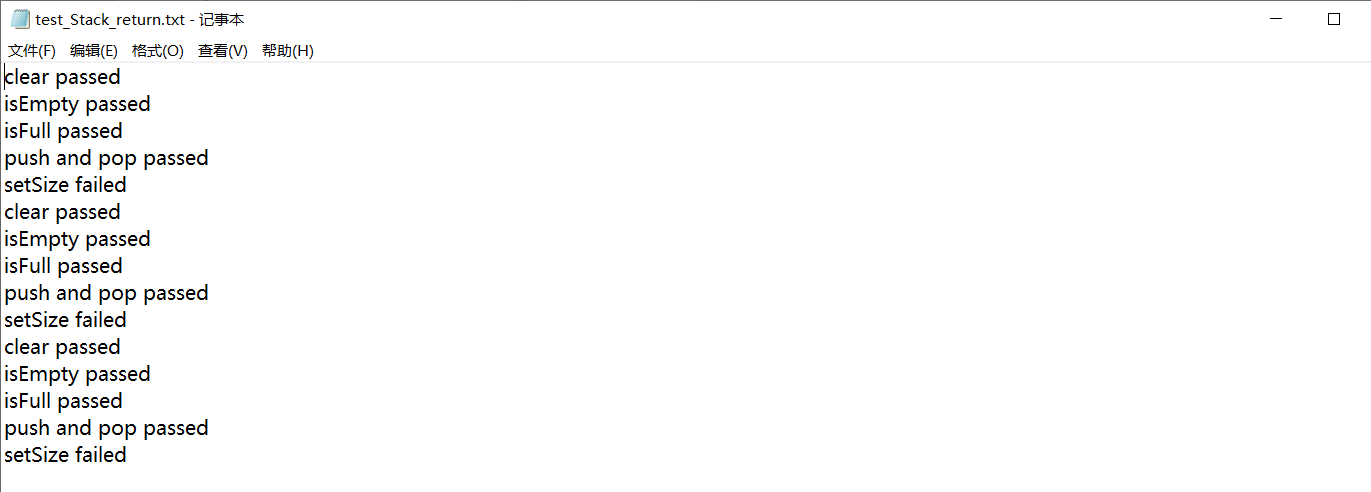


3 实验结果

实验全部结果如下：



写入txt的内容如下：



4实验过程中存在的问题及解决方案

5 实验总结

程序出错是一件非常难以避免的事情；异常一般是指程序运行时发生的错误；合理使用异常处理结构会使程序更加健壮；软件测试对于软件质量非常重要，单元测试是软件测试的重要技术之一。