**WordCount测试报告**

软工1603 任抒怀 U201617052

软工1603 李晨星 U201617044

软工1603 蔡东鸿 U201617038

软工1603 艾雨晨 U201616068

软工1603 刘俊傲 U201617047

软工1603 陈健豪 U201617040

# 1引言

## 1.1编写目的

编写此文档的目的是进一步定制软件开发的细节问题，希望能使本软件开发工作更具体。为了使用户、软件开发者及分析和测试人员对该软件的初始规定有一个共同的理解，它说明了本软件的各项功能需求、性能需求和数据需求，明确标识各项功能的具体含义，阐述实用背景及范围，提供客户解决问题或达到目标所需要的条件或权能，提供一个度量和遵循的基准。具体而言，编写软件需求说明的目的是为所开发的软件提出：

a) 软件设计总体要求，作为软件开发人员、软件测试人员相互了解的基础。

b) 功能、性能要求，数据结构和采集要求，重要的接口要求，作为软件设计人员进行概要设计的依据。

c) 软件确认测试的依据。

## 1.2背景

待开发的软件系统：WordCount。

本项目的任务提出者：华中科技大学软件学院武剑洁团队。

开发者：华中科技大学软件学院武剑洁团队。

用户：程序员、编辑、打字员、有字符统计需求的人等。

## 1.3定义

-c : 字符数

-w：单词总数

-l：总行数

-o：将结果输出到指定文件

-s：递归处理目录下符合条件的文件

-a：返回更复杂的数据(代码行 / 空行 / 注释行)

-e：停用词表，统计文件单词总数时，不统计该表中的单词

-x：该参数单独使用，如果命令行有该参数，则程序会显示图形界面，用户可以通过界面选取 单个文件，程序就会显示文件的字符数、单词数、行数等全部统计信息

## 1.4参考资料

《现代化软件工程讲义》

《WordCount原始程序需求》

# 2测试概要

## 2.1测试内容

|  |  |  |
| --- | --- | --- |
| 功能模块 | 需求名称 | 具体描述 |
| 基本功能 | 统计字符数 | 通过输入-c [文件名] 读取指定文件，并输出内容的字符数 |
| 统计单词数 | 通过输入-w [文件名] 读取指定文件，并输出内容的单词数 |
| 统计总行数 | 通过输入-l [文件名] 读取指定文件，并输出内容的总行数 |
| 输出统计信息 | 通过输入 –o [文件名] 将统计信息输出到指定文件中 |
| 共用输入文件 | -c,-w,-l可以共用一个输入文件。 |
| 统计要求说明 | 空格，水平制表符，换行符，均算字符。  由空格或逗号分割开的都视为单词，且不做单词的有效性校验，例如：thi#,that视为用逗号隔开的2个单词。  -c, -w, -l参数可以共用同一个输入文件，形如：wc.exe –w –c file.c 。  -o 必须与文件名同时使用，且输出文件必须紧跟在-o参数后面，不允许单独使用-o参数。 |
| 扩展功能 | 递归输入文件 | 通过输入-s 递归处理当前目录下所有文件。 |
| 统计代码行，空行，注释行 | 通过输入 –a [文件名] 读取指定文件，输出内容的代码行，空行，注释行。 |
| 设置停用词表 | 通过输入-e [文件名]，在读取指定文件时统计单词数时，忽略该文件中的单词，不进行统计。 |
| 通配符的使用 | 通过输入\*，代表选中符合条件的所有文件。 |
| 行类型判断 | 代码行：本行包括多于一个字符的代码。  空   行：本行全部是空格或格式控制字符，如果包括代码，则只有不超过一个可显示的字符，例如“{”。  注释行：本行不是代码行，并且本行包括注释。一个有趣的例子是有些程序员会在单字符后面加注释：  }//注释  在这种情况下，这一行属于注释行。 |
| 高级功能 | 图形界面 | 通过输入 –x，使用图形界面与用户进行清晰的交互。 |
| 显示顺序 | 若涉及多种数据的统计，按照字符-->单词-->行数-->代码行数/空行数/注释行的顺序，依次分行显示。 |

## 2.2测试方法

|  |  |  |  |
| --- | --- | --- | --- |
| 功能模块 | 功能细分 | 测试方法 | 负责人 |
| 基本功能 | 统计字符数 | 黑盒、白盒测试、单元测试 | 蔡东鸿 |
| 统计单词总数 | 黑盒、白盒测试、单元测试 | 蔡东鸿 |
| 统计总行数 | 黑盒、白盒测试、单元测试 | 陈健豪 |
| 输出结果到指定文件 | 黑盒、白盒测试 | 艾雨晨 |
| 扩展功能 | 递归处理文件 | 黑盒、白盒测试 | 李晨星 |
| 统计代码行/空行/注释行 | 黑盒、白盒测试、单元测试 | 任抒怀 |
| 根据停用词表统计单词总数 | 黑盒、白盒测试 | 刘俊傲 |
| 高级功能 | 通过图形界面完成基本功能 | 黑盒、白盒测试 | 任抒怀 |

# 3单元测试

## 3.1 单元测试脚本

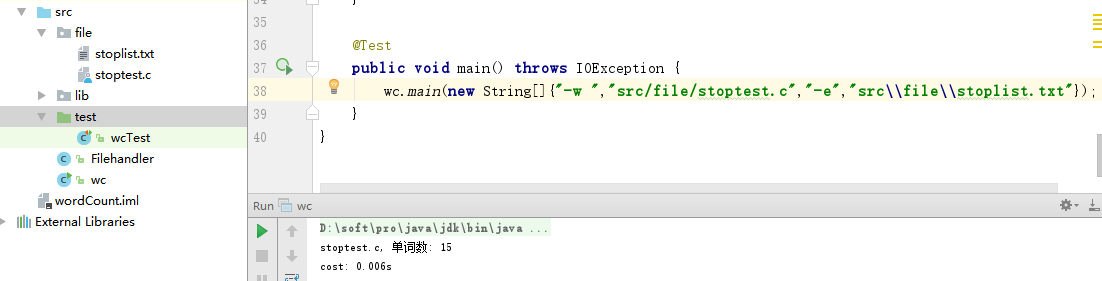
使用Junit4之后，只需在wc.java中点击Ctrl+Shift+T即可建立wcTest.java的测试文件

针对iteratorPath()函数与output()函数编写的Junit单元测试脚本如下：

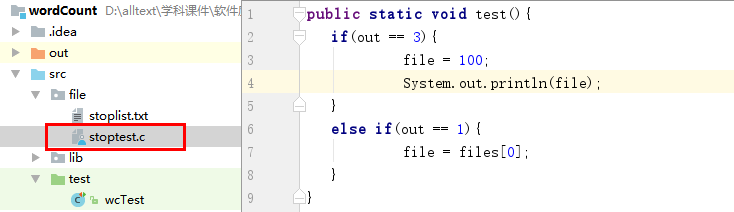


## 3.2 单元测试脚本执行情况

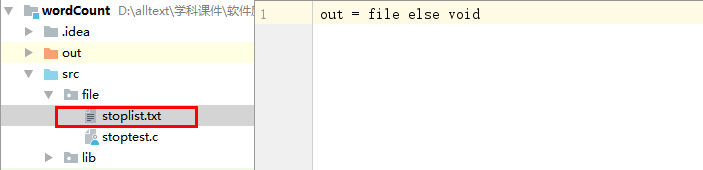
执行单元测试脚本wcTest.java得到结果：



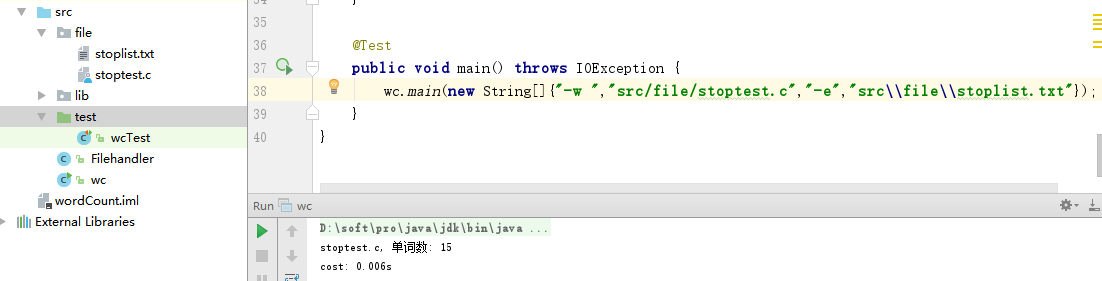
其中stoptest.c的内容为：



更改stoplist.txt的内容为：



结果：



# 4静态测试

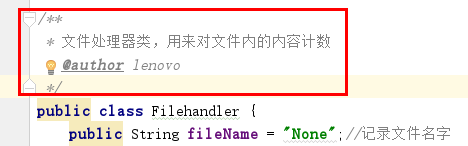
## 4.1 《阿里巴巴Java开发手册》阅读理解

《阿里巴巴Java开发手册》中指出

1. 类名使用UpperCamelCase风格，以下情形例外：DO / BO / DTO / VO / AO
2. 类、类属性、类方法的注释必须使用 Javadoc 规范，使用/\*内容/格式，不得使用//xxx 方式。

说明： 在 IDE 编辑窗口中， Javadoc 方式会提示相关注释，生成 Javadoc 可以正确输出相应注释； 在 IDE 中，工程调用方法时，不进入方法即可悬浮提示方法、参数、返回值的意义，提高阅读效率。

WordCount源代码在类名、类注释方面符合开发规范：



1. 【参考】Service/DAO层方法命名规约

1） 获取单个对象的方法用get作前缀。

2） 获取多个对象的方法用list作前缀。

3） 获取统计值的方法用count作前缀。

4） 插入的方法用save/insert作前缀。

5） 删除的方法用remove/delete作前缀。

6） 修改的方法用update作前缀。

WordCount程序非常简单且不是Java Web程序，并没有Service/DAO层，但这些命名规约还是对我们的代码命名产生很大启发。按照这种规约命名的方法，要实现什么功能一目了然，代码可读性大大提高。

1. 任何运算符左右必须加一个空格。

说明： 运算符包括赋值运算符=、逻辑运算符&&、加减乘除符号、三目运行符等。

1. 单行字符数限制不超过 120 个，超出需要换行，换行时遵循如下原则：

1） 第二行相对第一行缩进 4 个空格，从第三行开始，不再继续缩进，参考示例。

2） 运算符与下文一起换行。

3） 方法调用的点符号与下文一起换行。

4） 在多个参数超长，逗号后进行换行。

5） 在括号前不要换行，见反例

这些在IDEA中可以直接使用Ctrl+Shift+L进行代码格式化，以达到规范，非常方便。

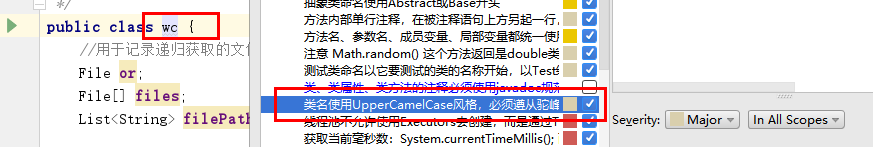
1. 【强制】在高并发场景中，避免使用”等于”判断作为中断或退出的条件。

说明：如果并发控制没有处理好，容易产生等值判断被“击穿”的情况，使用大于或小于的区间 判断条件来代替。

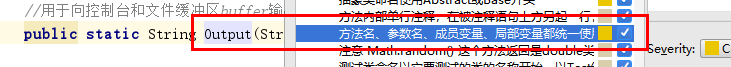
反例：判断剩余奖品数量等于 0 时，终止发放奖品，但因为并发处理错误导致奖品数量瞬间变 成了负数，这样的话，活动无法终止。

## 4.2 Findbugs静态代码检测结果

1. 类名wc首字母应大写



1. 方法名Output首字母应小写



## 4.3 动态测试与静态测试的对比

静态测试主要包括：

（1）代码检查：代码会审、代码走查、桌面检查；

（2）静态结构分析；

（3）代码质量度量。

        动态测试主要包括：

（1）黑盒测试：又称功能测试。这种方法把被测软件看成黑盒，在不考虑软件内部结构和特性的情况下测试软件的外部特性。

（2）白盒测试：又称结构测试。这种方法把被测软件看成白盒，根据程序的内部结构和逻辑设计来设计测试实例，对程序的路径和过程进行测试。

黑盒测试有可能是动态测试（运行程序，只看输入和输出），也有可能是静态测试（不运行程序，只是查看界面）

白盒测试有可能是动态测试（运行程序，并分析代码结构），也有可能是静态测试（不运行程序，只是静态查看代码）

动态测试有可能是黑盒测试（运行程序，只看输入和输出），也有可能是白盒测试（运行程序，并分析代码结构）

静态测试有可能是黑盒测试（不运行程序，只是查看界面），也有可能是白盒测试（不运行程序，只是静态查看代码）

# 5测试总结

## 5.1能力

WordCount的源程序较好的实现了基础功能与扩展功能。但对于高级功能，由于处理图形界面选择模式时，在判断分支内使用System.*exit*(0);主动结束程序，而之前未将outputBuffer打印出来，所以最终只能显示cost time，而不能显示统计信息。该程序的高级功能未能实现。

此外，该程序的隐式需求也未能实现。即对于用户错误的参数输入，不能给予错误信息反馈，只能打印Wrong!。

## 5.2缺陷和限制

见缺陷报告。

## 5.3建议

为了更好地满足质量方面对于软件的要求，在编码过程之前，就应该考虑用户的隐式需求和实际需求，并在设计上有所体现。在编码过程中，应当注意各个功能模块的独立性，应当注意对于错误的显示。在测试用例的设计方面，应当使用不同的测试用例的设计方法，尽可能达到高覆盖度，低冗余，方便定位缺陷等等要求。在单元测试的设计方面，应当依据单元测试框架设计单元测试，尽可能自动化执行

## 5.4评价

该项软件的开发仍存在缺陷，未达到预定目标，不能交付使用。

# 6团队贡献度

任抒怀：0.180

李晨星：0.172

艾雨晨：0.170

刘俊傲：0.167

蔡东鸿：0.156

陈健豪：0.155