

EX1 实验报告

一、小组成员及得分分配

小组成员：陈轶凡、陈香秀、李昶、徐昌越

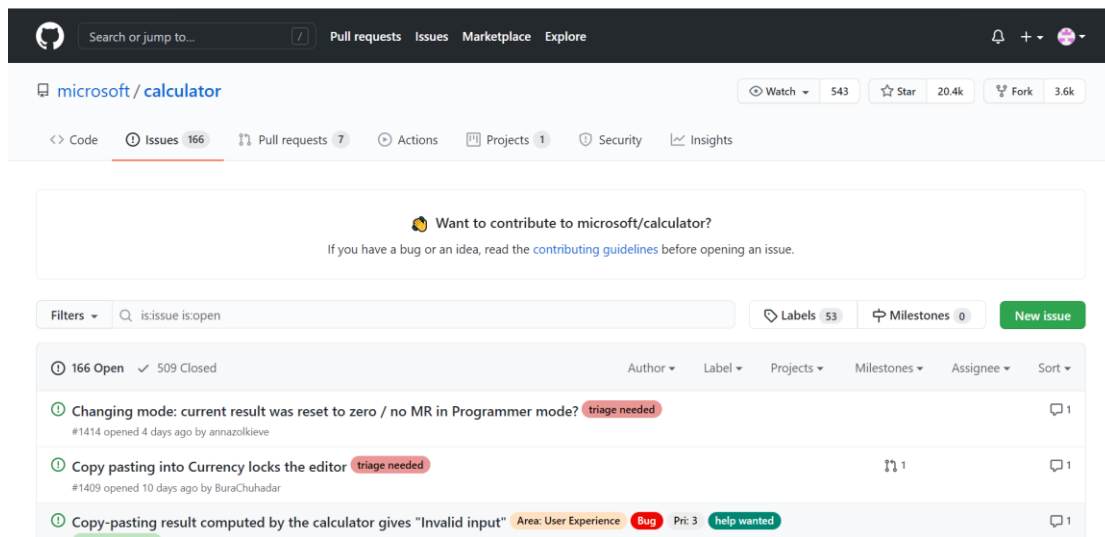
得分分配：陈轶凡 25% 陈香秀 25% 李昶 25% 徐昌越 25%

二、实验目的

通过本次实验体验软件需求分析的整个流程，熟悉软件需求获取与分析分类的各种方法，同时加深对所选择软件的了解，认识到软件需求对于整个软件开发流程的重要性与必要性。

三、实验方法（含数据获取）

(1) 本次实验我们小组所选取的开源项目是 Microsoft Calculator，即微软计算器。信息源是 Github 上对应项目的 issues 模块，网址为 <https://github.com/microsoft/calculator/issues>，具体实验思路参考思路一与思路二。



(2) 我们在 Github 上对应项目的 issues 版块爬取所有关于目标软件 Microsoft Calculator 的 500+ 条问答，将其保存至文本中形成问答文本。 https://github.com/youneta/nju2020-software_requirements/blob/main/Source%20code%20and%20Result/main.py

```

1  from lxml import etree
2  import requests
3
4  # 获取一个项目的issues列表
5  def get_issues_list(repo_name):
6      issues_list = []
7      url = 'https://github.com' + repo_name + '/issues'
8      # print(url)
9      response = requests.get(url)
10     # 获取源码
11     page_source = response.text
12     tree = etree.HTML(page_source)
13     # 获取issues数量
14     number = tree.xpath('//*[@id="js-repo-pjax-container"]/div[1]/nav/u1/li[2]/a/span[2]')
15     if len(number) == 0:
16         number = '0'
17     else:
18         number = number[0].text
19     if number.isdigit():
20         number = int(number)
21     else:
22         number = 1000
23     print(number)
24     # 计算分页数量，每页25个issues
25     page = 0
26     if number % 25 == 0:
27         page = int(number / 25)
28     else:
29         page = int(number / 25) + 1

```

(3) 我们将爬取的问答保存为问答文档，此外，我们还将英文版问答文本翻译简化为中文版本。[https://github.com/youneta/nju2020-software_requirements/blob/main/Source%20code%20and%20Result/result\(translated%20and%20simplified\).txt](https://github.com/youneta/nju2020-software_requirements/blob/main/Source%20code%20and%20Result/result(translated%20and%20simplified).txt)

```

1  166
2
3  描述错误
4
5  从任何来源复制粘贴到货币中，都会锁定编辑器，无法清除编辑器。
6  重现的步骤
7
8  打开记事本
9  文本 "20.83"，并复制文本。
10  打开应用，用Ctrl+V粘贴文字。
11  你看你删除或CE按钮并清除任何东西。
12  预期行为
13
14  我希望小编还能继续编辑。复制粘贴后什么都发生。
15  截图
16
17
18  设备和应用信息
19
20  OS Build: 10.0.19041.0
21  架构: X64
22  应用程序版本: 10.2009.4.0
23  地区: en-US
24  已安装的开发版本: 假
25
26
27  其他背景
28
29  不适用

```

(4) 而后对问答文本进行词云分析，得到若干关键词，词云也有中英版本，中文版还是有很多无关词，因此我们又对文档进行了简化筛选。

https://github.com/youneta/nju2020-software_requirements/blob/main/Source%20code%20and%20Result/makecloud.py

```

1 # from wordcloud import WordCloud #导入词云模块
2 # words = open('result.txt',encoding='utf-8').read()#打开歌词文件，获取到歌词
3 # wordcloud = WordCloud(width=1000, #图片的宽度
4 #                         height=860, #高度
5 #                         margin=2, #边距
6 #                         background_color='black',#指定背景颜色
7 #                         font_path='C:\Windows\Fonts\Sitka Banner\msyh.ttc'#指定字体文件，要有这个字体文件，自己随便想用什么字体，就下载一个，然后指定路径就ok了
8 #                         )
9 # wordcloud.generate(words) #分词
10 # wordcloud.to_file('cloud.jpg')#保存图片
11
12 # coding:utf-8
13
14 import jieba
15 import matplotlib.pyplot as plt
16 #from scipy.misc import imread
17 from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator
18 import numpy as np
19 from PIL import Image
20 text_from_file=open('result(translated and simplified).txt','r').read()
21 word_spilt_jieba = jieba.cut(text_from_file, cut_all = False)
22 word_space = ' '.join(word_spilt_jieba)
23 my_wordcloud = WordCloud(
24     background_color='white', #设置背景颜色
25     #mask=img, #背景图片
26     max_words = 200, #设置最大显示的词数
27     stopwords = STOPWORDS, #设置停用词
28     #设置字体格式，字体格式 .ttf文件需自己网上下载，最好将名字改为英文，中文名路径加载会出现问题。
29     font_path = 'simkai.ttf',
30     max_font_size = 100, #设置字体最大值

```

[https://github.com/youneta/nju2020-software_requirements/blob/main/Source%20code%20and%20Result/cloud\(Chinese_ver\).jpg](https://github.com/youneta/nju2020-software_requirements/blob/main/Source%20code%20and%20Result/cloud(Chinese_ver).jpg)



(5) 最后我们根据关键词对所有需求进行了分类 (聚类), 得到最终成果 (需求分析与需求分类), 见第四部分。

四、实验结果及效果分析

1. 实验结果

需求分析

https://github.com/youneta/nju2020-software_requirements/blob/main/%E8%BD%AF%E4%BB%B6%E9%9C%80%E6%B1%82%E5%88%86%E6%9E%90%E5%88%86%E7%B1%BB.docx

显示需求，提醒用户：

1. 显示常用公式的名称，通过选择一个公式，如交叉乘法，速度方程，或质量方程等，用户可以在底部的输入框输入值得到自动计算的结果。

在开发模式的计算器中添加开发横幅以明显区别

2. 在用户无法对“剪切”“复制”“粘贴”“全选”进行操作时，禁用这些按钮。
3. 在现有的科学计算器模式中增加支持在计算中使用重要数字，或增加一个单独的模式来进行重要数字的计算。其中重要数字（也叫有意义的数字）是科学和数学计算的重要组成部分，涉及到数字的准确性和精确性。
4. 用纯色和半透明色而不是点说明不等式。
5. 调整文字的发光率最低为 4.5:1。

减少用户重复性或可避免的额外劳动：

6. 保存计算器的状态，使得关闭计算器后重新打开时，恢复之前的历史和计算。
7. 复制数值时去除其中的空格。
8. 添加支持作为游戏栏小部件启动。
9. 添加在图形模式下定义新的函数时，引用之前定义的函数的功能。
10. 添加编辑从内存中插入的数值的功能。

支持更多数据类型与数据处理功能：

添加对自定义函数和自定义变量的支持。

11. 添加对复数的支持，允许绘制复数和位点的表格。

添加允许使用符号“cosec”表示余弦三角函数。

12. 添加支持“f(y)”类型的函数。
13. 允许图形模式绘制极性曲线，其形式为 $r=f(\theta)$ 。
14. 添加对基本财务公式的支持，允许用户在给定其他三个值的情况下，计算本金/利率/期限/总额中的一个。
15. 添加计算矩阵和矢量的功能，可以执行单矩阵运算（转置、反转、排序、确定、幂等...），执行两个矩阵运算（加、减、乘、除），可以使用 Vectors 和点/交叉积，可以在矩阵单元格中点击修改数值，用键盘有效输入矩阵值，使用方向键浏览矩阵单元，可以从历史中恢复工作状态。

添加绘制点、线段和多边形的功能。

添加排列组合的功能，可以使用 nCr 和 nPr 进行计算。

添加计算精确测量中的不确定性的功能，允许用户输入计算所需的不确定性，既可以是相对不确定性，也可以是绝对不确定性

16. 添加绘制参数方程的功能，允许以 $x=f(t)$, $y=g(t)$ 的形式绘制方程。
17. 允许在 3d 中绘制图形，允许绘制 $f(x)=g(y)=h(z)$ 形式的线段，以及绘制 $z=f(x,y)$ 等两变量函数。
18. 支持更多的数字系统作为输入，例如印度-阿拉伯数字系统

用户需要通过其他软件或方式实现的简易功能：

19. 添加度数、弧度、梯度和坡度之间互相转化的功能。
20. 添加将数学表达式解析为字符串的功能。
21. 添加 GitHub repo 的链接，提高 GitHub 项目在计算器内的可发现性。

给予用户更多的自主选择性：

22. 添加设置页面。
23. 保存图片时支持选择文件格式。
24. 在计算器中增加对自定义颜色和照片主题的支持。用户可以选择自定义颜色主题，可以选择一张图片作为主题。
25. 添加语言/地域设置，应用程序可以在系统默认的语言环境下启动，用户可以更改语言环境，允许在选定的语言环境下复制、输入和显示数据。

需求分类

https://github.com/youneta/nju2020-software_requirements/blob/main/%E5%BE%90%E6%98%8C%E8%B6%8A_%E8%AE%A1%E7%AE%97%E5%99%A8%E9%9C%80%E6%B1%82.docx

功能需求

26. 显示常用公式的名称，通过选择一个公式，如交叉乘法，速度方程，或质量方程等，用户可以在底部的输入框输入值得到自动计算的结果。
27. 保存计算器的状态，使得关闭计算器后重新打开时，恢复之前的历史和计算。
28. 添加绘制点、线段和多边形的功能。
29. 添加排列组合的功能，可以使用 nCr 和 nPr 进行计算。
30. 添加对自定义函数和自定义变量的支持。
31. 添加对复数的支持，允许绘制复数和位点的表格。
32. 复制数值时去除其中的空格。
33. 保存图片时支持选择文件格式。
34. 在开发模式的计算器中添加开发横幅以明显区别。
35. 添加允许使用符号 "cosec" 表示余弦三角函数。
36. 添加支持 "f(y)" 类型的函数。
37. 添加支持作为游戏栏小部件启动。
38. 在用户无法对 "剪切" "复制" "粘贴" "全选" 进行操作时，禁用这些按钮。
39. 添加计算精确测量中的不确定性的功能，允许用户输入计算所需的不确定性，既可以是相对不确定性，也可以是绝对不确定性。
40. 添加在图形模式下定义新的函数时，引用之前定义的函数的功能。
41. 添加度数、弧度、梯度和坡度之间互相转化的功能。
42. 添加绘制参数方程的功能，允许以 $x=f(t)$, $y=g(t)$ 的形式绘制方程。
43. 允许在 3d 中绘制图形，允许绘制 $f(x)=g(y)=h(z)$ 形式的线段，以及绘制 $z=f(x,y)$ 等两变量函数。
44. 允许图形模式绘制极性曲线，其形式为 $r=f(\theta)$ 。
45. 添加对基本财务公式的支持，允许用户在给定其他三个值的情况下，计算本金/利率/期限/总额中的一个。
46. 在现有的科学计算器模式中增加支持在计算中使用重要数字，或增加一个单独的模式来进行重要数字的计算。其中重要数字（也叫有意义的数字）是科学和数学计算的重要组成部分，涉及到数字的准确性和精确性。
47. 添加设置页面。

- 48. 添加编辑从内存中插入的数值的功能。
- 49. 添加将数学表达式解析为字符串的功能。
- 50. 添加计算矩阵和矢量的功能，可以执行单矩阵运算（转置、反转、排序、确定、幂等...），执行两个矩阵运算（加、减、乘、除），可以使用 Vectors 和点/交叉积，可以在矩阵单元格中点击修改数值，用键盘有效输入矩阵值，使用方向键浏览矩阵单元，可以从历史中恢复工作状态。

非功能需求

- 51. 用纯色和半透明色而不是点说明不等式。
- 52. 调整文字的发光率最低为 4.5:1。
- 53. 在计算器中增加对自定义颜色和照片主题的支持。用户可以选择自定义颜色主题，可以选择一张图片作为主题。
- 54. 添加 GitHub repo 的链接，提高 GitHub 项目在计算器内的可发现性。
- 55. 使转换器完全由数据驱动，并与在线数据源或服务集成，这样转换器可以提供更全面的、每天更新的转换表。
- 56. 添加语言/地域设置，应用程序可以在系统默认的语言环境下启动，用户可以更改语言环境，允许在选定的语言环境下复制、输入和显示数据。
- 57. 支持更多的数字系统作为输入，例如印度-阿拉伯数字系统。

2. 效果分析

我们所用的方法相对于传统人工遍历所有需求后进行分类分析，更加快捷高效，极短时间内便可以显示出软件的主要需求，并辅助我们对软件的需求进行分类，使用效果相对较好。但是使用过程中也出现很多问题，如爬取的问答中许多与需求无关的高频词，做词云时这些词会挤占真正高频词的空间，因此我们需要进行一些筛选。此外由于单个词汇较短，所以我们无法完全根据词汇得出对应需求，还是需要查找问答文本再进行具体分析分类。另外，通过该方法分析出的更多是目标软件所欠缺的功能需求。总的来说，这个方法还是利大于弊。

五、结论

Microsoft Calculator 相对比较稳定，问题主要集中于屏幕显示、数据输入处理以及用户使用不够自主便捷这三点。