

数据结构项目五文档

同济大学 软件学院 15级2班 1552651 王依睿

目录

- [使用说明](#)
 - [操作手册](#)
 - [注意事项](#)
- [概述](#)
 - [项目功能要求](#)
 - [程序设计目的](#)
 - [算法思路](#)
 - [数据结构](#)
 - [文件目录](#)
- [源代码](#)

使用说明

操作手册

- 运行程序后，请求用户输入文件名。

```
关键字检索系统
请输入文件名:  src.txt
```

- 请求用户输入一段英文，以“Ctrl+Z”结束。

```
请输入一段英文:
Data data "Data "data Data" data" DATA Data. I love data structure.
^Z
```

- 输出“本段文本已保存在文本文件中”字样，表示此段文本已保存在当前目录下用户输入的文件名中。

本段文本已保存在文本文件“src.txt”中。

- 请求用户输入要检索的关键字。

请输入要检索的关键字: data

- 输出“显示源文件”字样，输出源文件。

显示源文件“src.txt”:

Data data "Data "data Data" data" DATA Data. I love data structure.

- 输出检索到的关键字个数和关键字。

在源文件中共检索到: 9个关键字"data"

- 程序结束。



```
C:\WINDOWS\system32\cmd.exe
关键字检索系统
请输入文件名: src.txt

请输入一段英文:
Data data "Data "data Data" data" DATA Data. I love data structure.
^Z
^Z
本段文本已保存在文本文件“src.txt”中。

请输入要检索的关键字: data
显示源文件“src.txt”:

Data data "Data "data Data" data" DATA Data. I love data structure.

在源文件中共检索到: 9个关键字"data"
请按任意键继续. . .
```

注意事项

- 输入文本以“Ctrl+Z”结束。

- 输入的文本保存在当前目录下用户输入的文件名中。
 - 仅允许输入英文。
 - 文本中单词以空格区分。
 - 文本中单词在计数时对大小写不敏感。
 - 文本中单词在保存到文件中时对大小写敏感。
 - 文本中单词如与标点符号相邻，不影响其被计入。
-

概述

建立一个文本文件，文件名由用户用键盘输入，输入一个不含空格的关键字，统计输出关键字在文本中的出现次数。

项目功能要求

本项目的设计要求可以分成两个部分实现：首先建立一个文本文件，文件名由用户用键盘输入；然后输入一个不含空格的关键字，统计输出该单词在文本中的出现次数。

程序设计目的

实现一个可以将文本写入文件、将文本从文件中读取并记录文章中各单词出现次数的简单程序。

算法思路

- 利用C++内置IO库中头文件fstream中ofstream实现向文件写入数据，ifstream实现从文件中读取数据。
- 利用C++STL中关联容器map实现对文本中各单词的计数。

数据结构

- C++STL中关联容器map。

文件目录

- 5_1552651_wangyirui.cpp（主文件）
 - 5_1552651_wangyirui.exe（可执行文件）
 - 5_1552651_wangyirui.pdf（项目文档）
-

源代码

```
#include<iostream>
#include<fstream>
#include<string>
#include<algorithm>
#include<map>

using namespace std;

int main() {

    map<string, size_t> article;
    string filename;

    cout << "关键字检索系统\n"
         << "请输入文件名:  " << flush;

    cin >> filename;
    ofstream outfile(filename);
    cout << "\n请输入一段英文:  " << endl;

    string word;
    char letter;
    getchar();
    //将文本逐字符读入文件中
    while (scanf("%c", &letter) != EOF)
        outfile << letter;

    outfile.close();
    cout << "本段文本已保存在文本文件\"src.txt\"中。\\n\\n" << endl;

    ifstream infile(filename);
    string keyword;
    cout << "请输入要检索的关键字:  " << flush;
    //读入用户输入的关键字
    cin >> keyword;
    cout << "显示源文件\" " << filename << "\":\\n" << endl;

    //将文件中的文本逐字符串读入，逐字符串进行处理
    while (infile >> word) {
        string realword;
        //将单词中的标点符号去除
        for (auto it = word.begin(); it != word.end(); it++) {
            if ((it == word.begin() || it == word.end() - 1) && ispunct(*it))
```

```

        continue;
        realword += *it;
    }
    //将单词转为全小写字母
    transform(realword.begin(), realword.end(), realword.begin(), ::tolower);
    //若该单词不存在map中, 则将该将单词放入map中, 否则map的值自加一
    ++article[realword];
    cout << word << " ";
}

//将用户输入的关键字转为全小写字母
transform(keyword.begin(), keyword.end(), keyword.begin(), ::tolower);

//在map中搜索用户输入的关键字
auto it = article.find(keyword);
cout << "\n\n在源文件中检索到:  " << it->second << "个关键字\"" << keyword << "\"
" << endl;

return 0;
}

```