**项目说明文档**

**数据结构课程设计**

**——关键字检索系统**

作 者 姓 名： 戴仁杰

学 号： 1951650

指 导 教 师： 张颖

学院、 专业： 软件学院 软件工程

同济大学

Tongji University

目 录

[1 分析 - 3 -](#_Toc91082819)

[1.1 背景分析 - 3 -](#_Toc91082820)

[1.2 功能分析 - 3 -](#_Toc91082821)

[2 设计 - 3 -](#_Toc91082822)

[2.1 数据结构设计 - 3 -](#_Toc91082823)

[2.2 类结构设计 - 3 -](#_Toc91082824)

[2.3 成员与操作设计 - 3 -](#_Toc91082825)

[2.3.1 KeywordSearchSystem类 - 3 -](#_Toc91082826)

[2.4 系统设计 - 4 -](#_Toc91082827)

[3 实现 - 5 -](#_Toc91082828)

[3.1 新建文件并写入文本功能的实现 - 5 -](#_Toc91082829)

[3.1.1 新建文件并写入文本功能流程图 - 5 -](#_Toc91082830)

[3.1.2 新建文件并写入文本功能核心代码 - 5 -](#_Toc91082831)

[3.1.3 新建文件并写入文本功能截屏展示 - 6 -](#_Toc91082832)

[3.2 关键字匹配和统计次数功能的实现 - 6 -](#_Toc91082833)

[3.2.1 关键字匹配和统计次数功能流程图 - 6 -](#_Toc91082834)

[3.2.2 关键字匹配和统计次数功能核心代码 - 7 -](#_Toc91082835)

[3.2.3 关键字匹配和统计次数功能截屏示例 - 8 -](#_Toc91082836)

[3.3 总体功能的实现 - 8 -](#_Toc91082837)

[3.3.1 总体功能流程图 - 8 -](#_Toc91082838)

[3.3.2 总体功能核心代码 - 8 -](#_Toc91082839)

[3.3.3 总体功能截屏示例 - 8 -](#_Toc91082840)

[4 测试 - 9 -](#_Toc91082841)

[4.1 正常功能测试 - 9 -](#_Toc91082842)

[4.2边界测试 - 9 -](#_Toc91082843)

[4.2.1 关键词不存在的情况 - 9 -](#_Toc91082844)

[4.2.2 查询与标点符号相邻的单词 - 9 -](#_Toc91082845)

[4.2.3 查询段落开头或结尾的单词 - 10 -](#_Toc91082846)

[4.3 出错测试 - 10 -](#_Toc91082847)

[4.3.1 输入的文件名不以.txt结尾 - 10 -](#_Toc91082848)

[4.3.2 输入的文件名长度不合法 - 10 -](#_Toc91082849)

1 分析

1.1 背景分析

文本文件在我们日常生活中应用十分广泛，对于文件中关键字的检索也是常用的功能，探究如何快速高效的检索出所需关键字具有十分重要的意义。本项目要求建立一个文本文件，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。

1.2 功能分析

对于文件关键字的查询，主要分为两个模块。首先要建立起一个文本文件文件，然后向文件中写入数据。此时操作要用到文件流ofstream进行操作。其次是读取文件并存入一个string字符串，然后对字符串进行解析，此时需要用ifstream文件流。对于读出的字符串，为了加快关键词查询的过程，可以使用KMP算法进行匹配，然后统计其出现的次数。

综上，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。

2 设计

2.1 数据结构设计

如上功能分析所述，读写文件要用到文件流ofstream和ifstream，所以要实例化两个流对象进行文件读写操作。对于读出的文件内容，用一个string类型字符串来进行接收。此外，为了加快关键词匹配，用到KMP算法，其中要用到一个数组记录回退的位置。

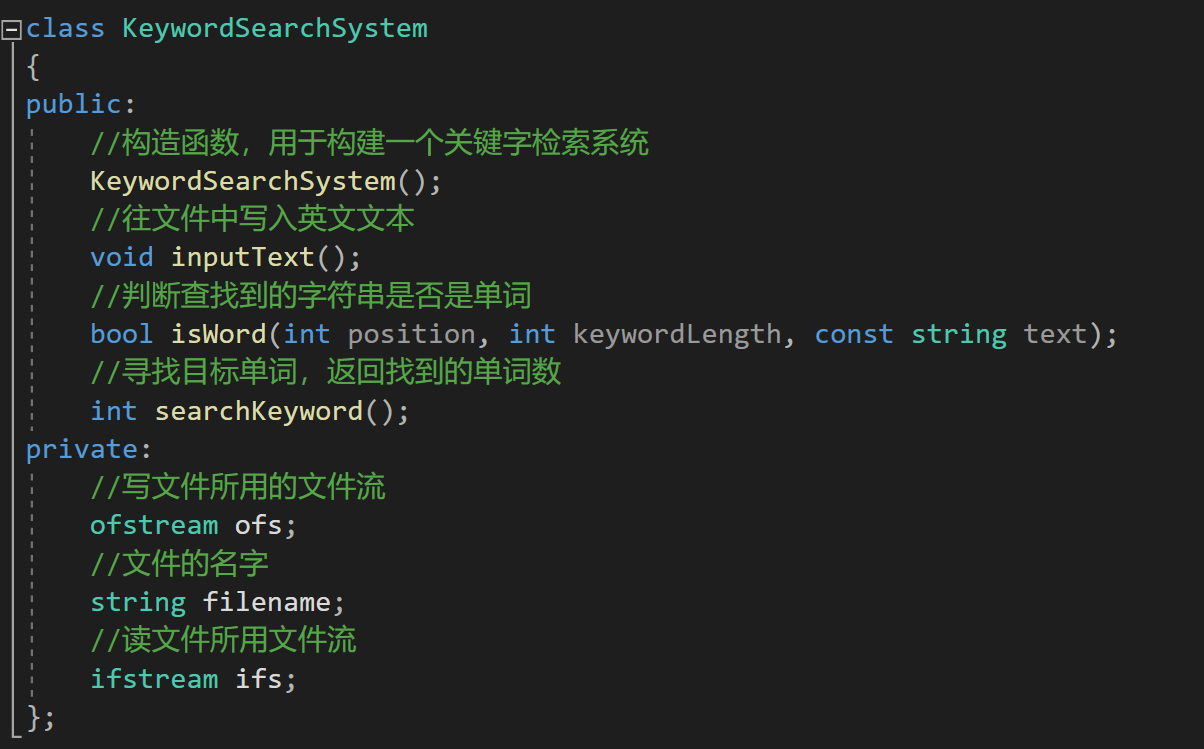
2.2 类结构设计

为了方便管理，将KMP算法解决关键词匹配的程序封装成了一个KeywordSearchSystem类，类内包含了与文件读取和关键词匹配、计数有关的函数方法，在main函数中进行调用即可实现N皇后问题的求解。

2.3 成员与操作设计

2.3.1 KeywordSearchSystem类

KeywordSearchSystem类为关键词检索系统类。关键词检索系统类包含了构造函数，作用为构建关键字检索系统，在磁盘新建一个文本文件并打开。此外，关键词检索系统还包含了写入文本方法和判断是否为单词的方法，为后续关键词的查找做准备。对于查找关键词并计数，KeywordSearchSystem类封装了一个利用KMP算法快速找到指定关键词的函数(searchKeyword()函数)。类内成员变量为两个用于文件操作的文件流对象和一个存储文件名的string字符串。



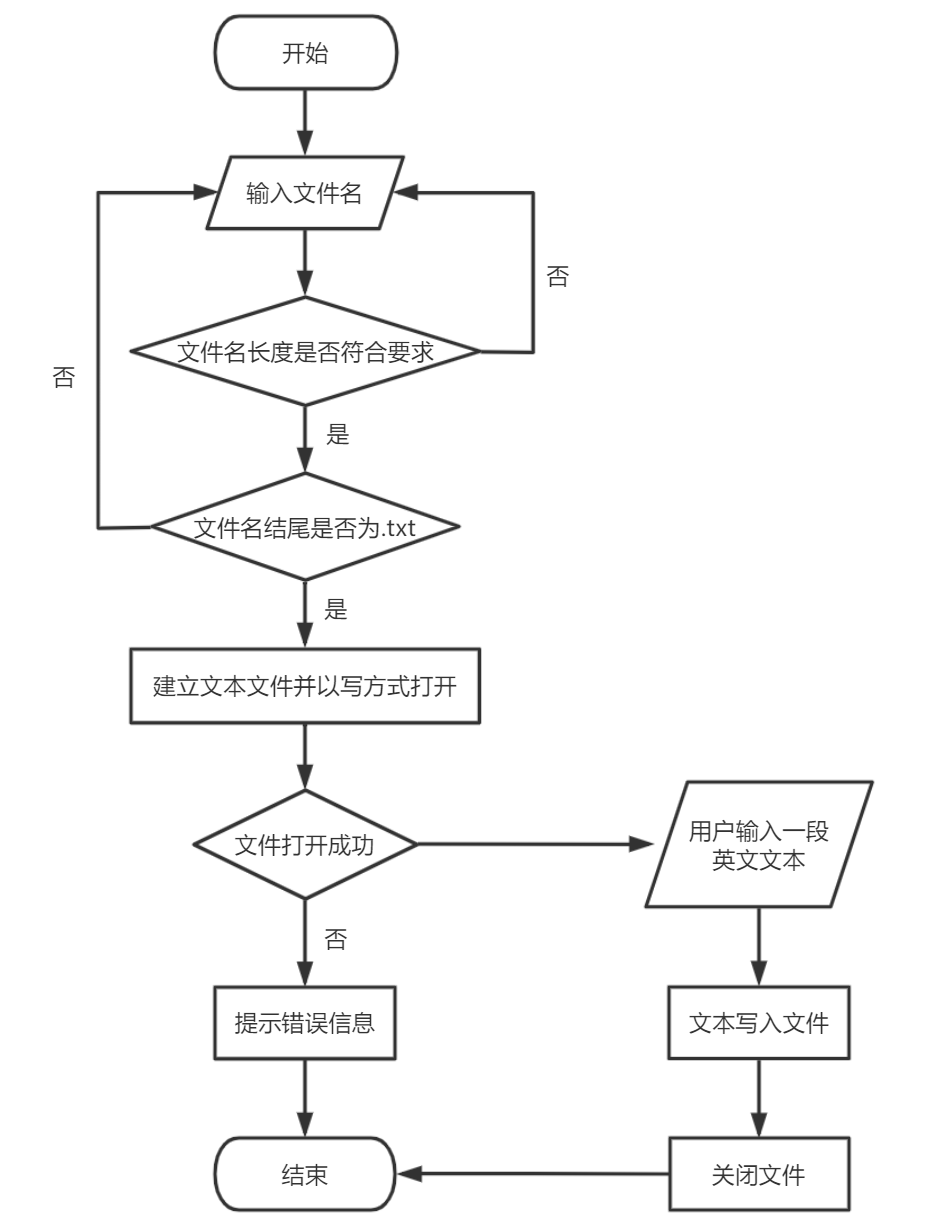
2.4 系统设计

程序开始后，首先进入KeywordSearchSystem类的构造函数，让用户输入新建的文件名，之后建立文件让用户输入一段英文（即后续要查询的文本），并将这段文本存入.txt文本文件。之后用户输入要检索的关键字，searchKeyword()函数把文本文件的内容读取到字符串，然后利用KMP算法遍历进行匹配、计数等操作。最后，程序输出该文本文件中关键字的匹配结果。

3 实现

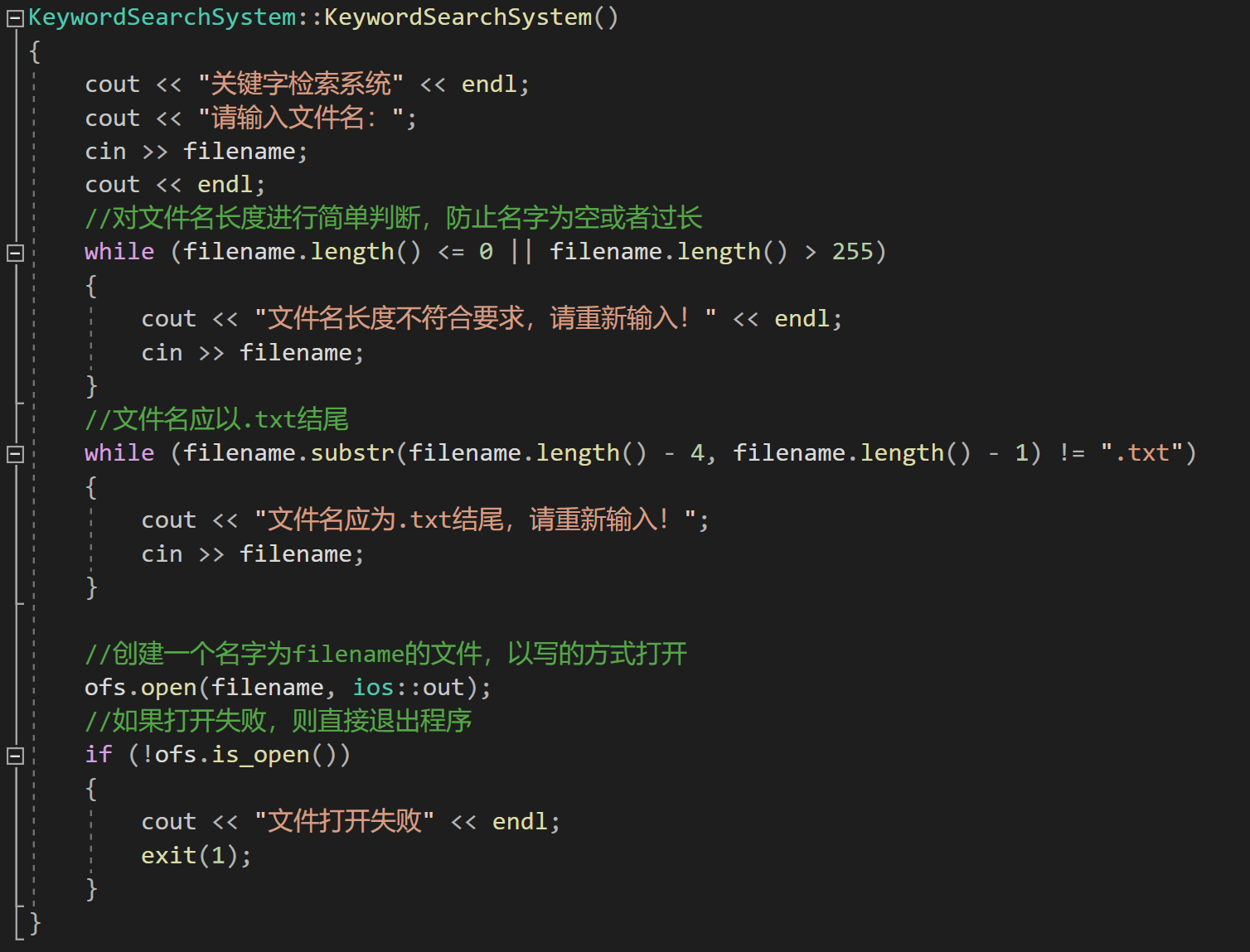
3.1 新建文件并写入文本功能的实现

3.1.1 新建文件并写入文本功能流程图

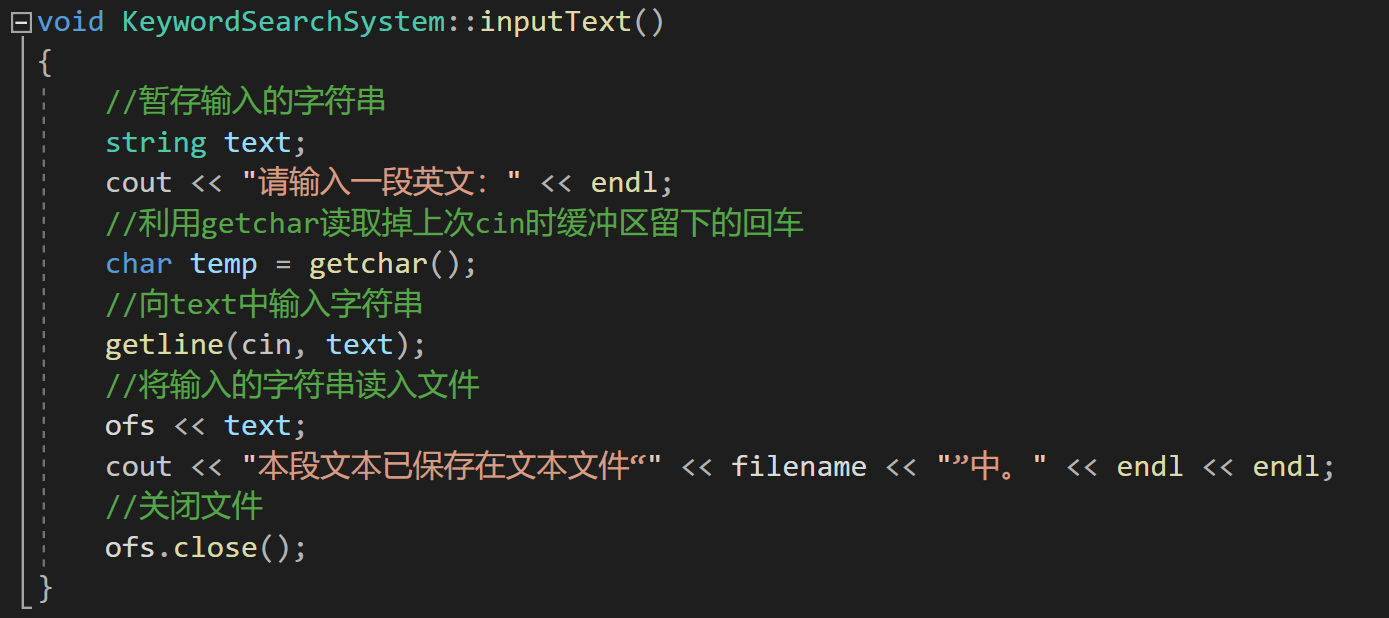


3.1.2 新建文件并写入文本功能核心代码

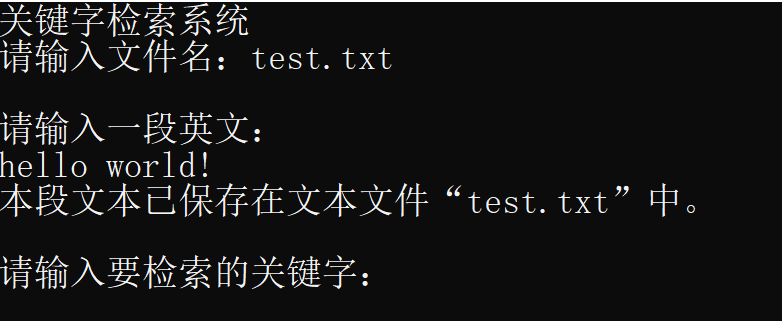
1.KeywordSearchSystem类的构造函数



2 KeywordSearchSystem类的inputText()函数.

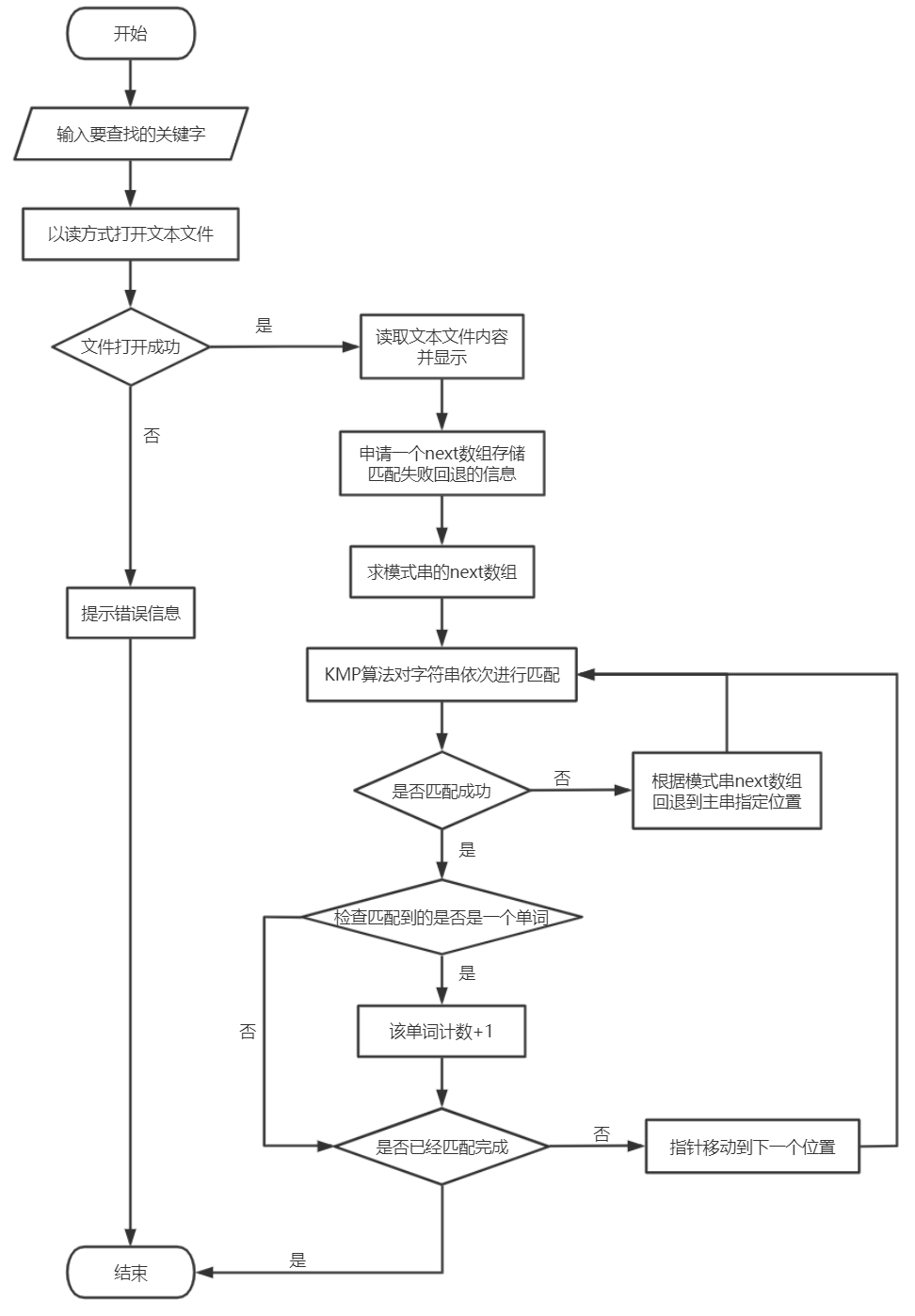


### 3.1.3 新建文件并写入文本功能截屏展示



3.2 关键字匹配和统计次数功能的实现

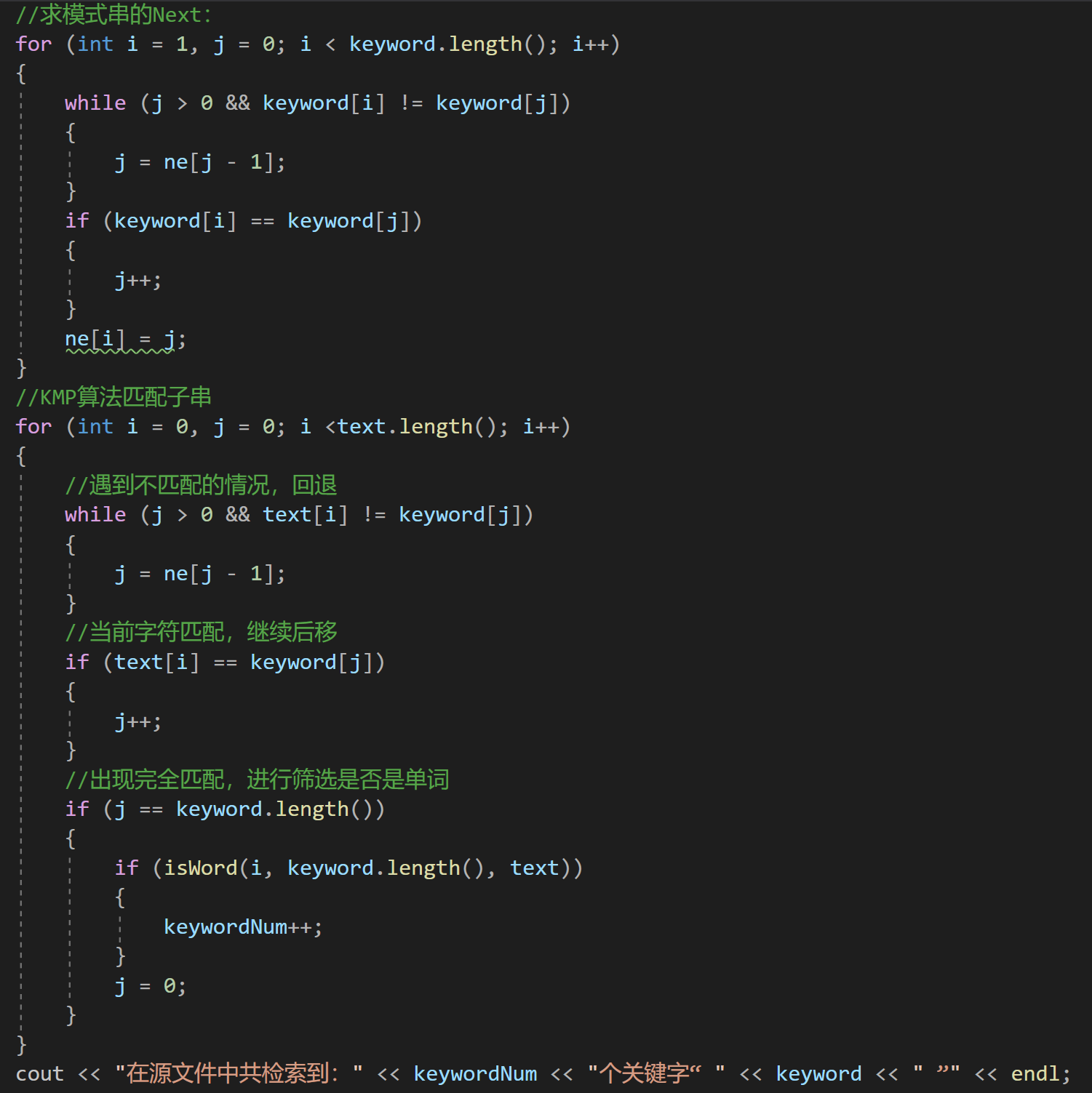
3.2.1 关键字匹配和统计次数功能流程图



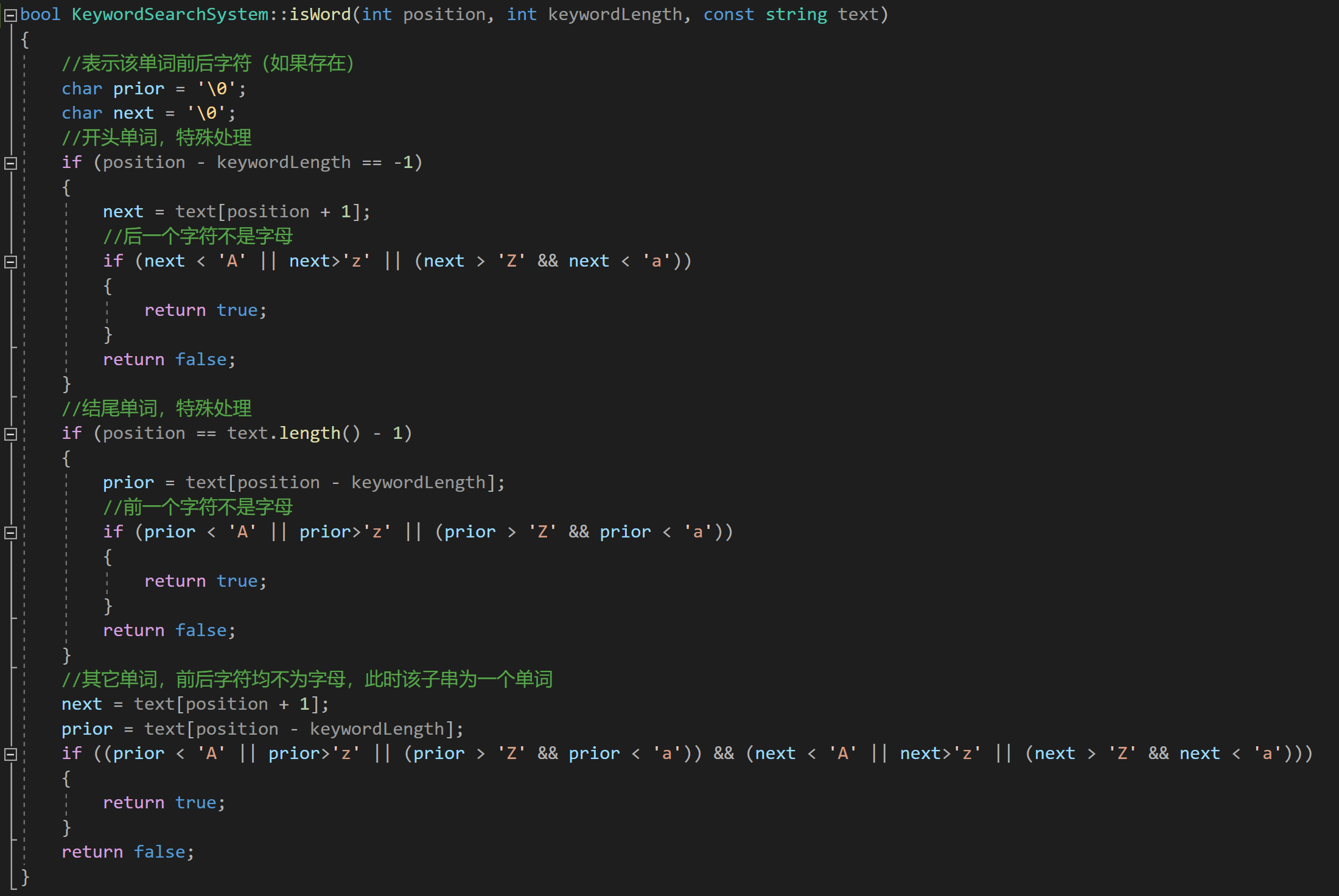
3.2.2 关键字匹配和统计次数功能核心代码

1.KeywordSearchSystem类的searchKeyword()函数

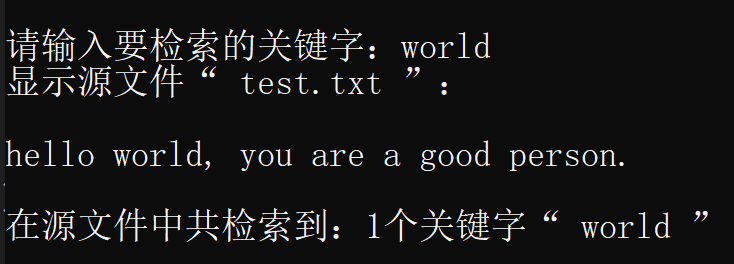
注：仅截取了求模式串next与匹配计数部分



2.KeywordSearchSystem类的isWord(int position, int keywordLength, const string text)函数

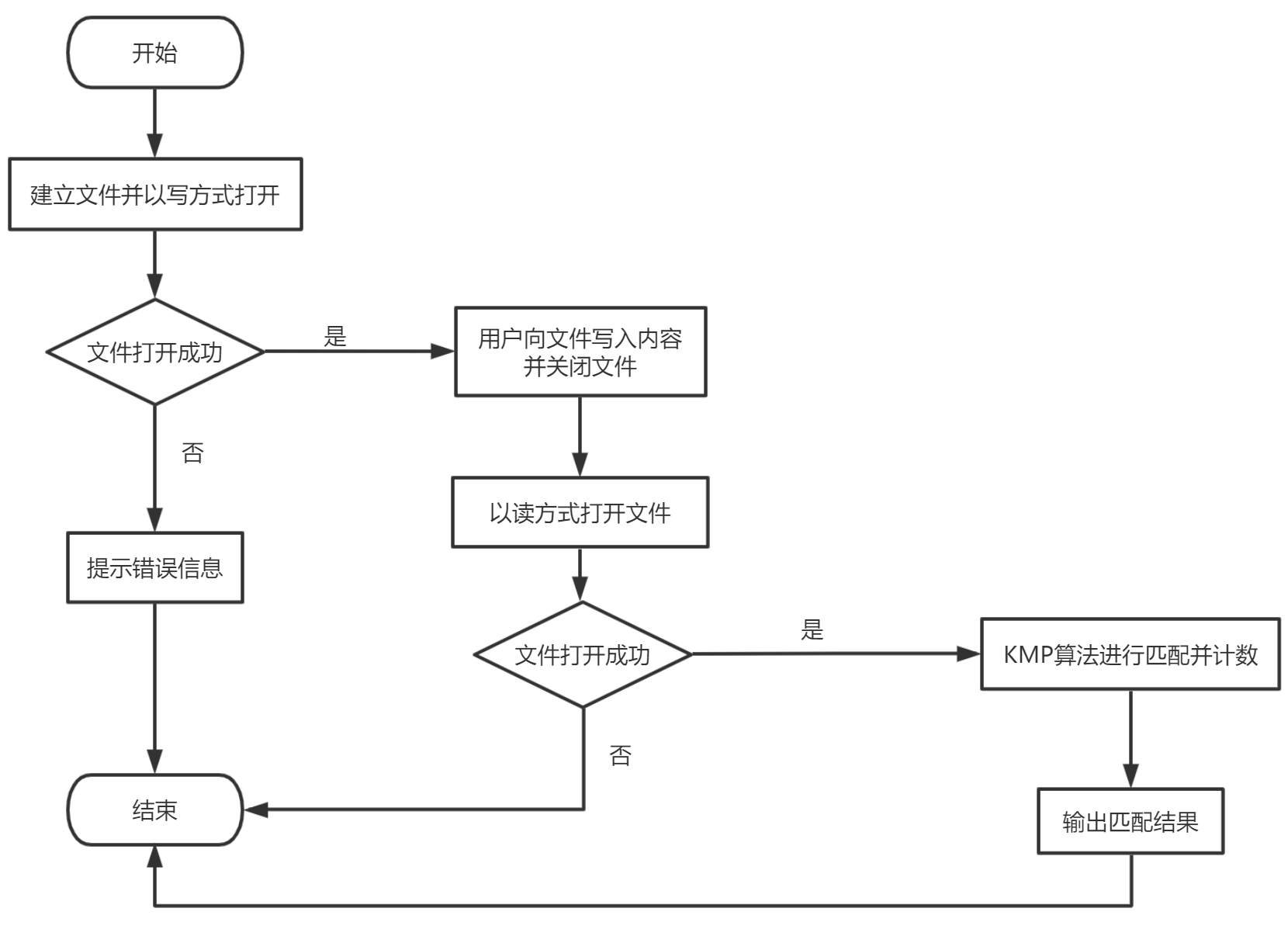


3.2.3 关键字匹配和统计次数功能截屏示例

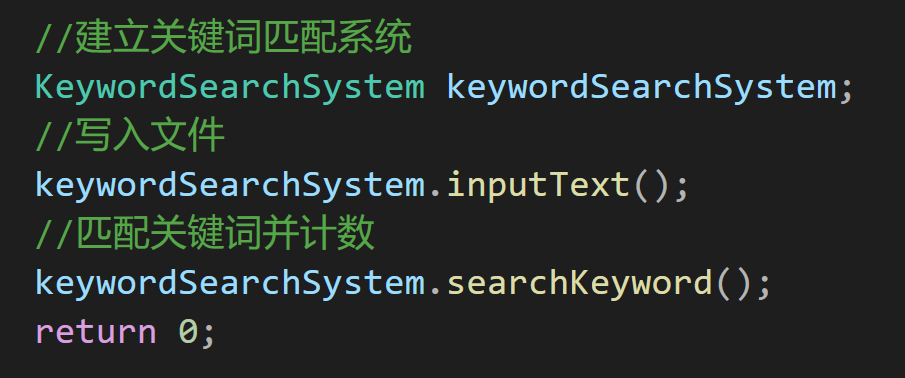


## 3.3 总体功能的实现

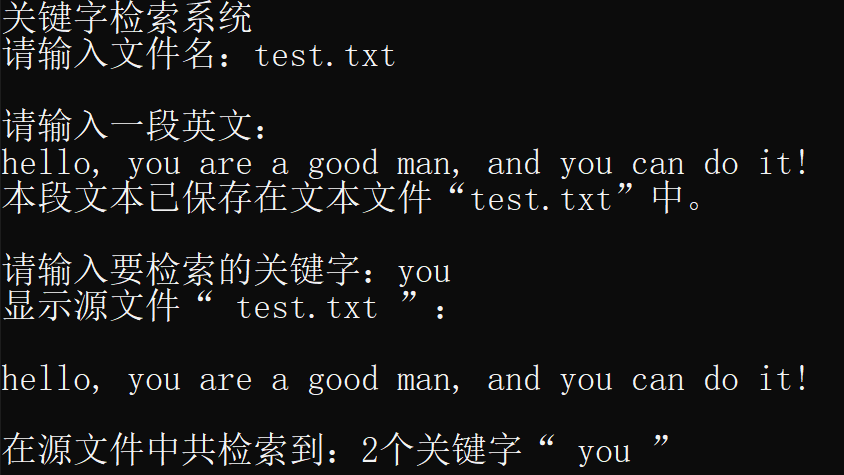
### 3.3.1 总体功能流程图



### 3.3.2 总体功能核心代码



### 3.3.3 总体功能截屏示例

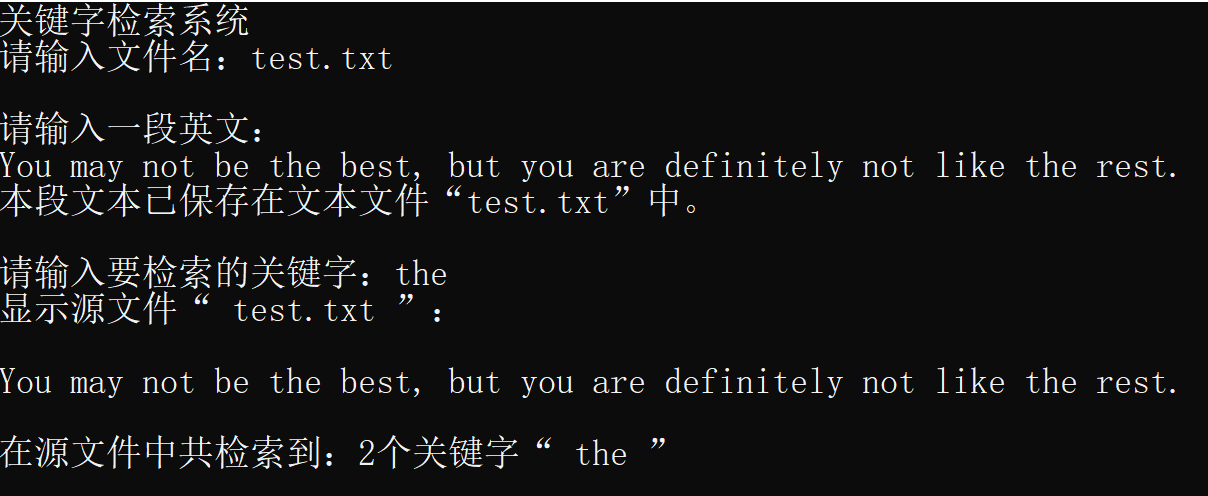


4 测试

4.1 正常功能测试

**测试用例**：文件名合法，关键词为一个正确单词

**实验结果：**符合预期（段落成功存入文本文件，关键字统计数目正确）

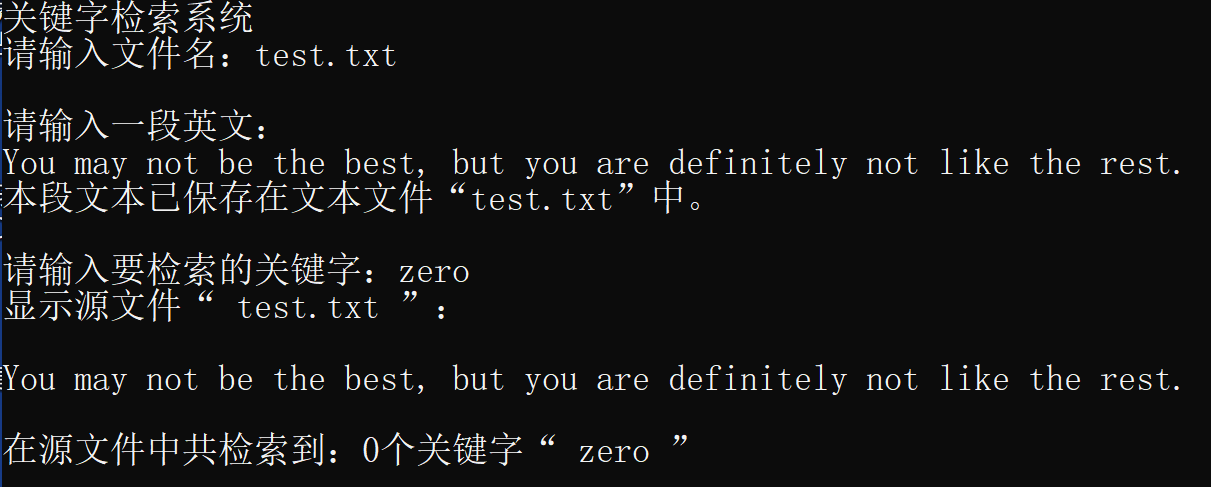


4.2边界测试

4.2.1 关键词不存在的情况

**测试用例**：文件名合法，关键词为文本中不存在的单词

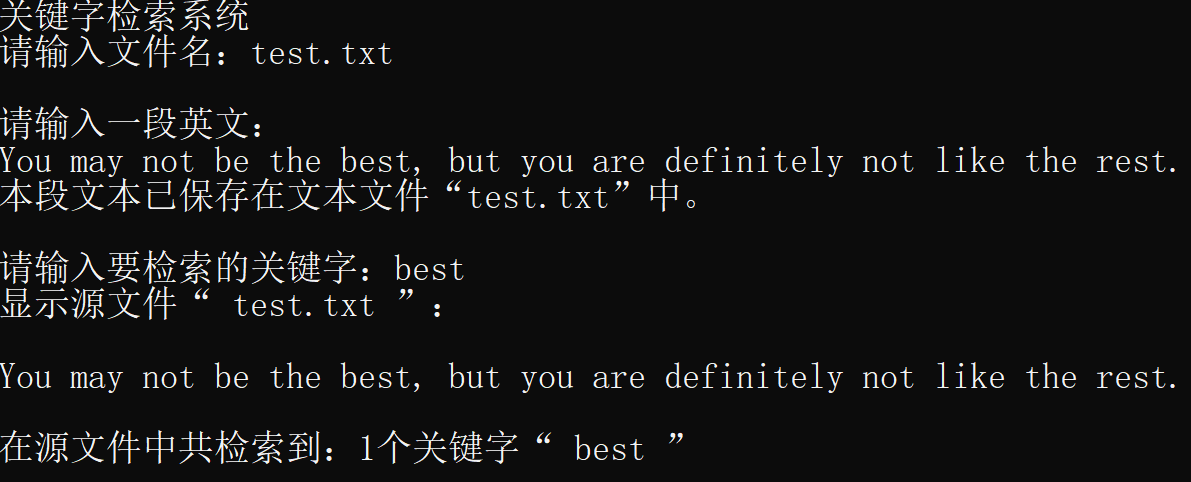
**实验结果：**符合预期（段落成功存入文本文件，关键字统计数目为0）



4.2.2 查询与标点符号相邻的单词

**测试用例**：文件名合法，关键词为文本中与标点符号相邻的单词

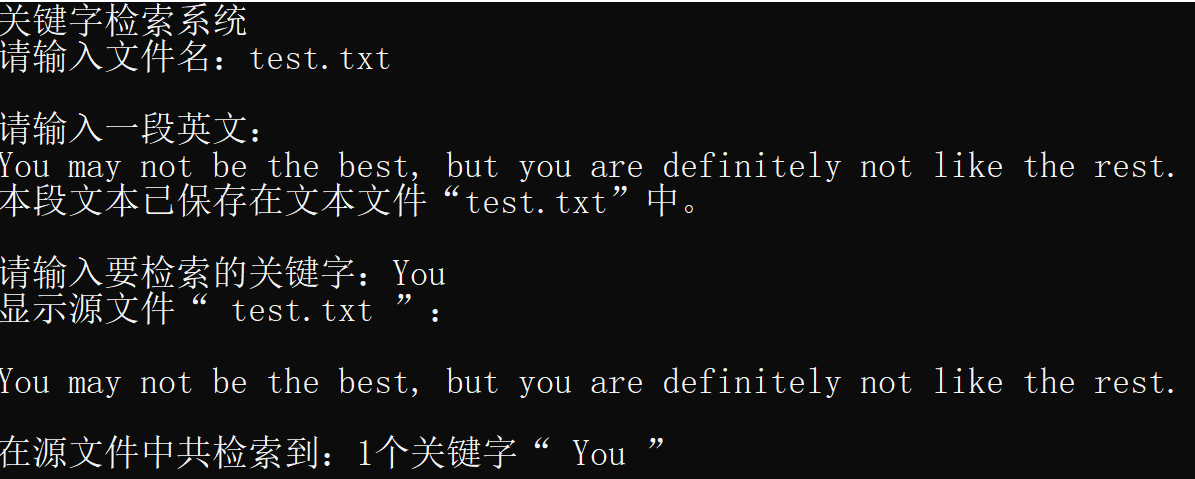
**实验结果：**符合预期（段落成功存入文本文件，关键字统计数目正确）

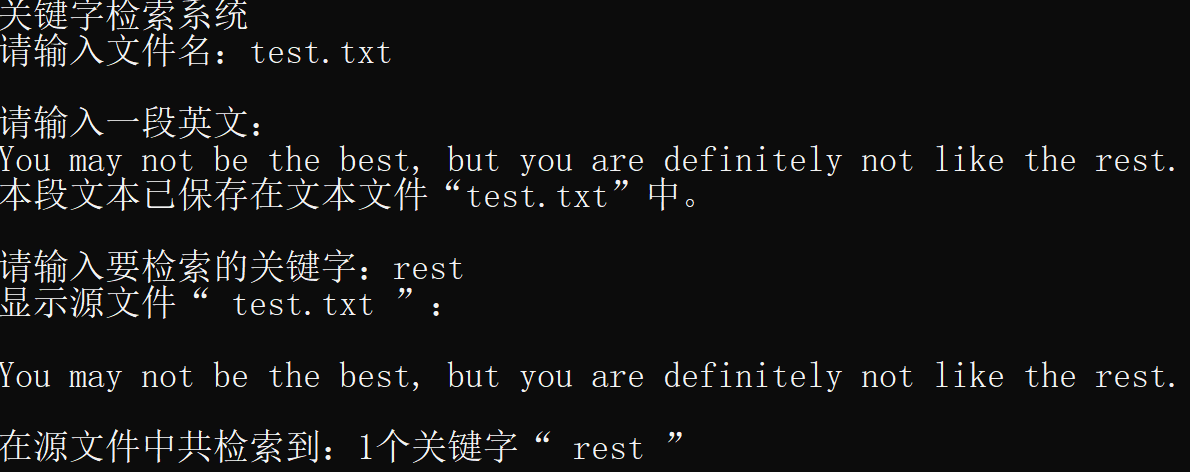


4.2.3 查询段落开头或结尾的单词

**测试用例**：文件名合法，关键词为文本中开头或结尾单词

**实验结果：**符合预期（段落成功存入文本文件，关键字统计数目正确）





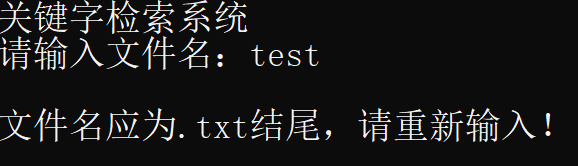
4.3 出错测试

4.3.1 输入的文件名不以.txt结尾

**测试用例**：文件名不以.txt结尾

**预期结果**：提示输入错误，并要求重新输入

**实验结果：**符合预期



4.3.2 输入的文件名长度不合法

**测试用例**：文件名长度非法

**预期结果**：提示输入错误，并要求重新输入

**实验结果：**符合预期

