* + 1. *Перекрестные ссылки. Бинарное дерево поиска.*

***3530904/90002 Мэн Цзянин***

1. **Общая постановка задачи**
   1. Для разрабатываемого словаря реализовать основные операции:
      * INSERT (ключ, значение) – добавить запись с указанным ключом и значением
      * SEARCH (ключ)- найти запись с указанным ключом
      * DELETE (ключ)- удалить запись с указанным ключом
   2. Предусмотреть обработку и инициализацию исключительных ситуаций, связанных, например, с проверкой значения полей перед инициализацией и присваиванием.
   3. Программа должна быть написана в соответствии со стандартом программирования: C++

Programming Style Guidelines (<http://geosoft.no/development/cppstyle.html>).

* 1. Тесты должны учитывать как допустимые, так и не допустимые последовательности входных данных.

1. **Требования**

Разработать и реализовать алгоритм формирования перекрестных ссылок:

* прочитать текст и вывести его с добавлением последовательных номеров строк;
* собрать все слова, встречающиеся в тексте;
* сформировать таблицу, в которой все слова будут расположены в алфавитном порядке и для каждого слова будет указан список строк его нахождения (по возрастанию номеров строк)

Для реализации задания использовать бинарное дерево поиска, узел которого может содержать:

* Ключ – слово
* Информационная часть – ссылка на список, содержащий номера строк

1. **Спецификация**
   1. Файл должен быть открыт
      1. В противном случае выводим ошибку, содержащую текст: «cant\_open\_the\_file»
   2. root, который в бинарном дереве поиска нельзя было nullptr
   3. все данные должны иметь правильный тип
      1. Поля Номер строки(row), столбца(col), счетчик(count) и длина слова(length) должны иметь тип данных unsigned int
      2. Номер строки, столбца и длина слова должны больше чем 1
   4. Если пустая строка пропускается, счетчик не накапливается
   5. Для повторяющих слов
      1. В одной статье может быть несколько повторяющихся слов,и номера строк и столбцов этих слов должны храниться в определенном контейнере
      2. При печати функции(printWordAndInfo), которая выводит слова, должен быть предоставлен внешний интерфейс, чтобы пользователь мог выбрать, следует ли печатать повторяющиеся слова
         1. Если «bool repetitive\_word = true»: выводит все повторяющиеся слова вместе с их информацией
         2. Если «bool repetitive\_word = false»: выводит это слово, номер строк и столбцов, которые появляются в первый раз
   6. Для одного и того же слова, но с большой буквы или с “ ’s ”. Все буквы должны быть преобразованы в маленькую букву.

**Пример:**

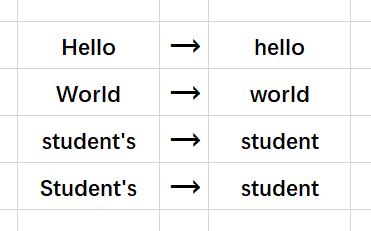


Рис.1

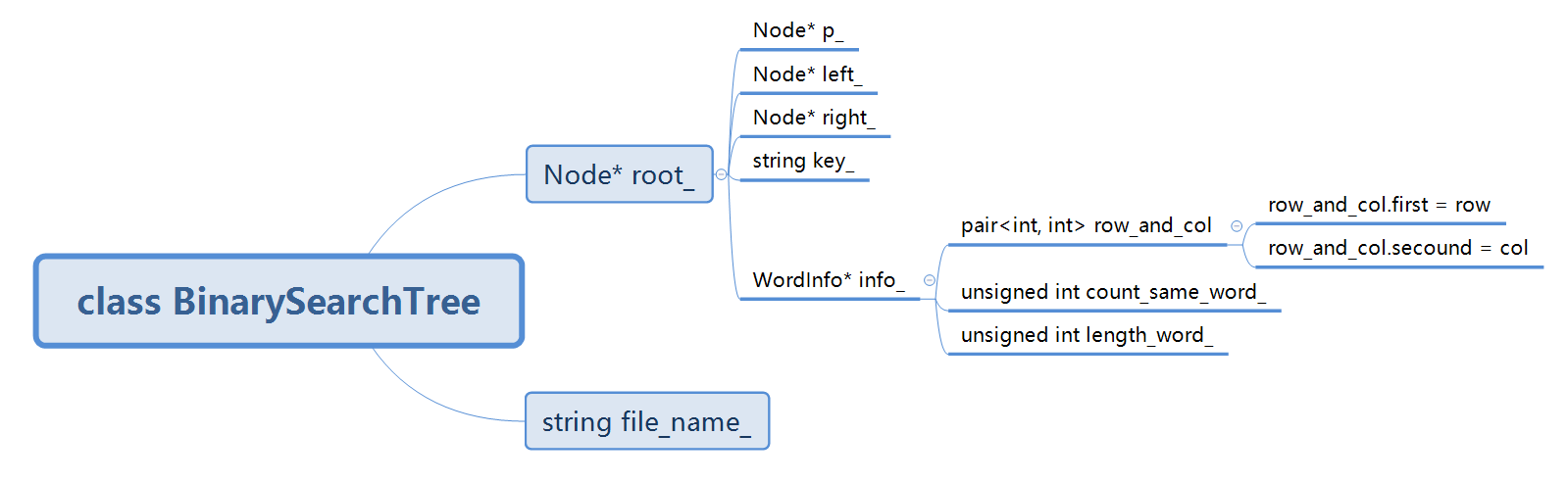
1. **Структура данных**

Рис.2

1. **Функции**

* Сохранить имя файла, которое пользователь вводит

void openFile(string file\_name);

* добавить запись с указанным ключом и значением

void printTextWithRowNum()

* найти запись с указанным ключом

searchWordInTree();

* удалить запись с указанным ключом

deleteNode ();

* собрать все слова, встречающиеся в тексте

makeWordTree();

* прочитать текст и вывести его с добавлением последовательных номеров строк

addRowNum();

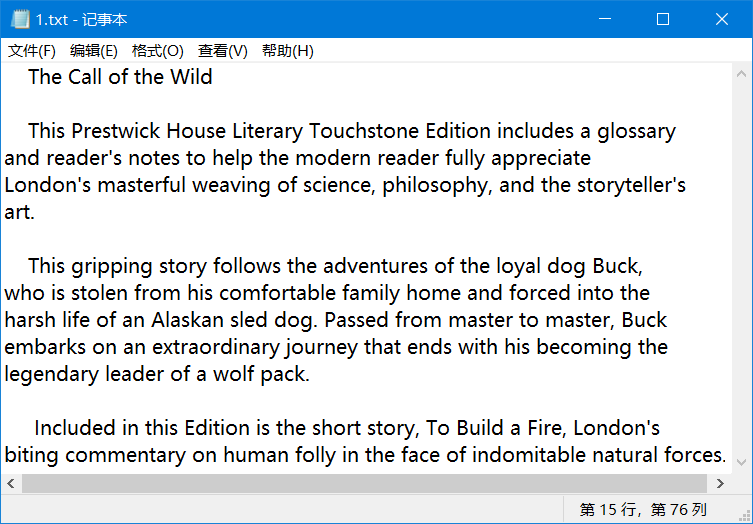
* Выводить в фале

outputInFile();

1. **Тест план**

\*для тестового примера используется файл с название 1.txt. Текст (1.txt) представлена для правильного ввода

Рис.3 (1.txt)

Ввод:

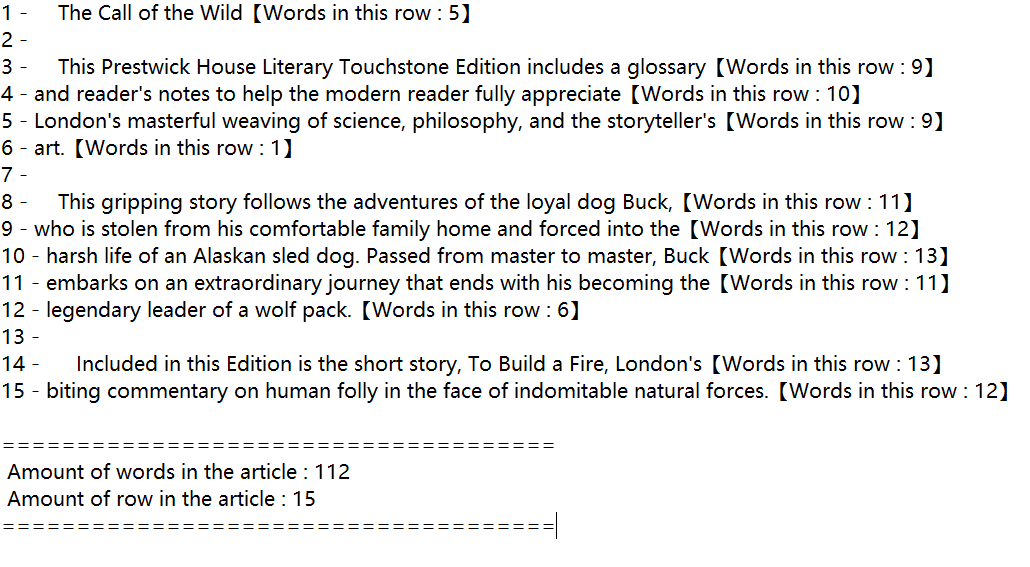
Вывод (Пример):

Рис.4 **(Часть 1 - Пример)**

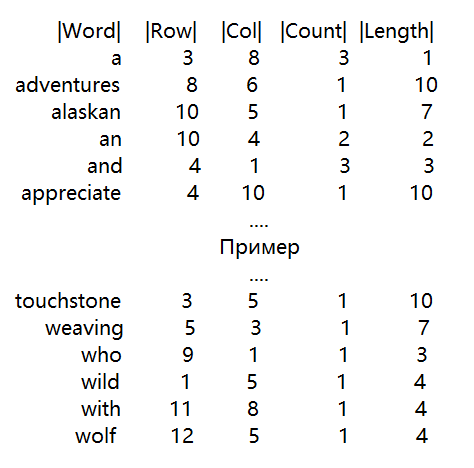
****

Рис.5 **(Часть 2 - Пример)**

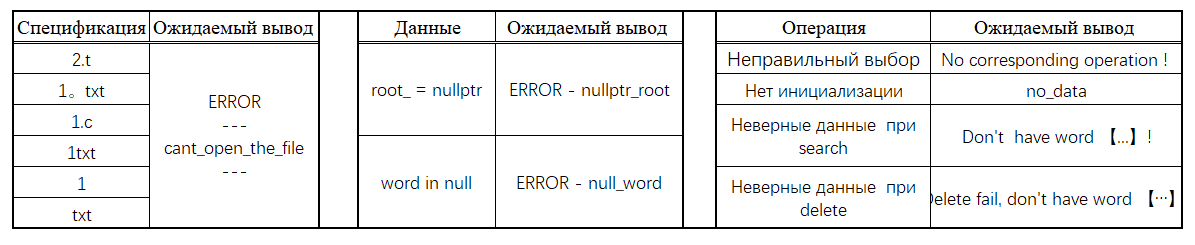
1. **Для некорректного ввода данных:**

Таблица 1