

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра ВТ

ОТЧЕТ
по Лабораторной работе №1
по дисциплине «Программирование»
Тема:
Модельное представление строки с маркером

Студент гр. 9005

Самуйлов. Е.С.

Преподаватель

Калмычков В.А.

Санкт-Петербург

2020

Цель работы.

Изучение и практическое освоение приемов программирования функций для представления и выполнения элементарных операций с текстом (символьной информацией) из строк с маркером на основе структур (классов) с использованием массива (обычного/динамического) и организации работы с файлами (чтение/запись текста) с использованием библиотек `stdio` или `fstream`, оформления фрагментов программы (типов и функций) в виде многомодульной реализации (набор файлов в проекте).

Purpose of work.

Studying and practical mastering of methods of programming functions for representing and performing elementary operations with text (symbolic information) from lines with a marker based on structures (classes) using an array (regular / dynamic) and organizing work with files (reading / writing text) using `stdio` or `fstream` libraries, designing program fragments (types and functions) in the form of a multi-module implementation (a set of files in a project).

Основные теоретические положения.

Для выполнения было получено задание на разработку программы. В этой программе необходимо использовать модельное представление строки с маркером. Общим функционалом является работа с файлами и вывод полученных результатов.

Задание

В заданной строке, которая состоит из слов, разделенных одним или несколькими пробелами определить количество слов, начинающихся с заданного символа.

Внешний формат хранения данных

Для работы используется 2 файла, путь до которых вводится вручную

- 1) для взятия данных
- 2) для вывода результатов работы

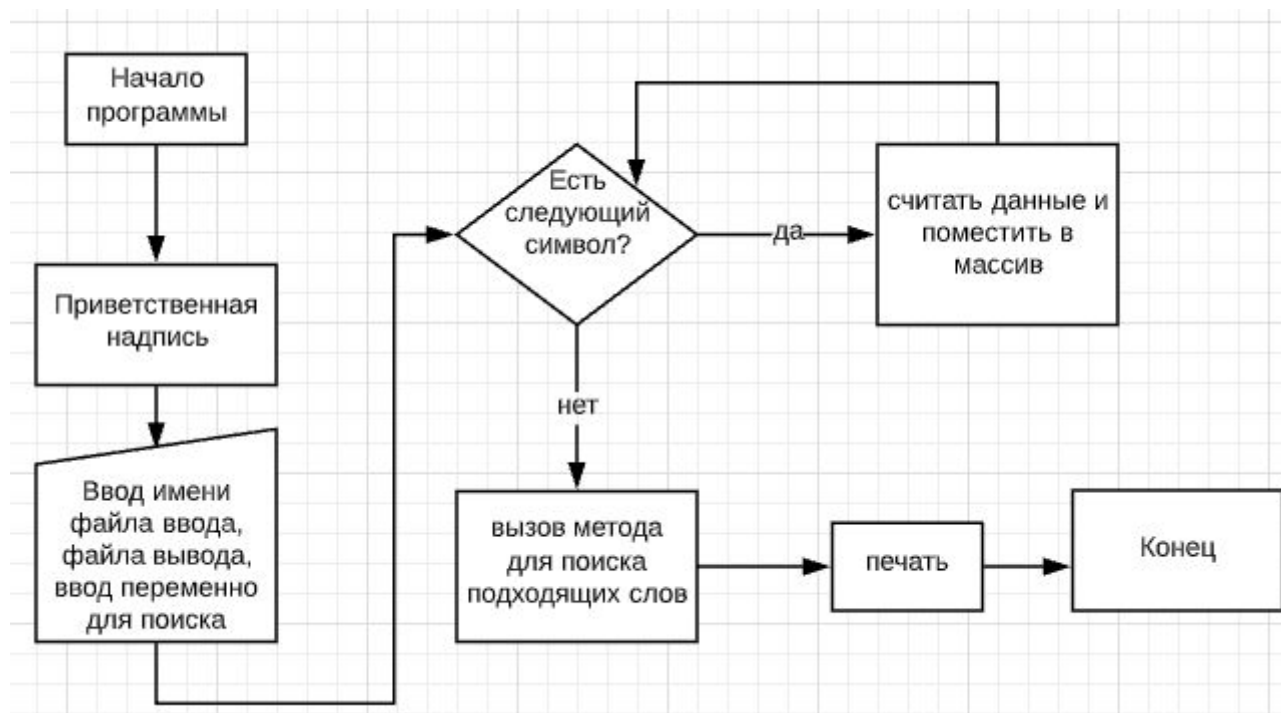
Внутренний формат хранения данных

Вычитанная строка помещается в массив `char text[MAX_SIZE]{};`, далее вся обработка (для вычисления и для печати) просто перебирает каждый элемент и обрабатывает.

Алгоритм обработки данных

После запуска программы из консоли пользователь видит приветственную надпись с описанием программы, далее ему будет предложено ввести файл, откуда происходит считывание, имя файла вывода и букву, с которой необходимо сравнить. Далее данные считываются и заносятся в массив, по окончании ставится маркер конца строки. Затем вызывается метод обработки, который ищет все слова, которые начинаются с введенного символа. Затем вызывается метод для печати.

Блок-схема обработки данных



Код программы

```
#include <iostream>
#include <fstream>

using namespace std;

const int MAX_SIZE = 2000;

struct Sentence {
public:
    char litera;
    char text[MAX_SIZE]{};
    int index = 0;
    char marker = '%';
    void addChar(const char &inputChar) {
        if (inputChar == '\n') {
            return;
        }

        this->text[this->index] = inputChar;
        this->index++;
    }

    bool isMarker(const char &inputChar) {
        return inputChar == this->marker;
    }

    int countOfWord()
    {
        char alfa; // буква
        int index = 0;
        int counter = 0;
        while (alfa != marker)
        {
            alfa = text[index];
            if(alfa == litera)
                if(text[index - 1] == ' ' || index == 0)counter++;

            index++;
        }
        return counter;
    }

    void printer (string nameOfFile, int counter)
    {
        ofstream out;
        out.open(nameOfFile, ios::app);
        if (out.is_open()) {
            out << "for letter " << litera << ", count = " << counter;
        }
    }
};
```

```

        out.close();
    }
};

int main() {
    cout
        << "hello, this program can check how many word in this sentence have on first position have a certain letter\n"
        "if you want start you need write name of file in, on next line name of file out, and on next line write letter\n";

    string input;
    string outName;
    Sentence line;
    cin >> input;
    cin >> outName;
    cin >> line.litera;
    ifstream inputFile(input);
    if (inputFile.is_open()) {
        char inputChar;

        while (inputFile >> noskipws >> inputChar) {
            line.addChar(inputChar);
        }

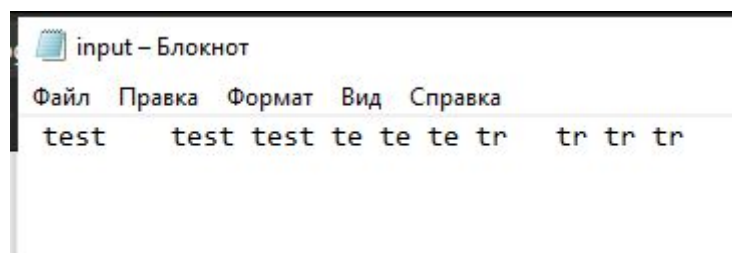
        line.addChar( inputChar: '%');
        inputFile.close();
    }
    int result = line.countOfWord();
    line.printer(outName,result);
}

```

Примеры работы программы

Пример 2

Файл ввода



Приветствие и входные данные

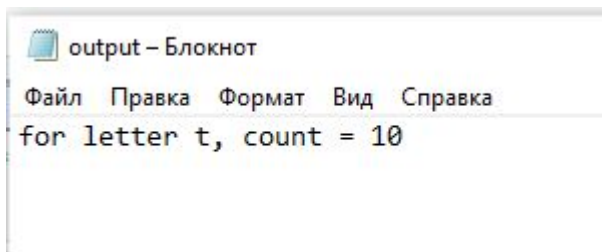
```

C:\tmp\lab_1_19B\cmake-build-debug\lab_1_19B.exe
hello, this program can check how many word in this sentence have on first position have a certain letter
if you want start you need write name of file in, on next line name of file out, and on next line write letter
C:\tmp\courseTes\input.txt
C:\tmp\courseTes\output.txt
t

```

Здесь я ожидаю увидеть ответ 10

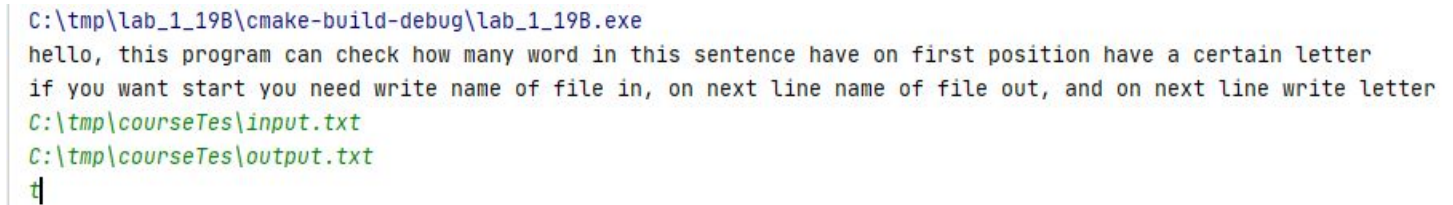
Результат работы



```
output - Блокнот
Файл  Правка  Формат  Вид  Справка
for letter t, count = 10
```

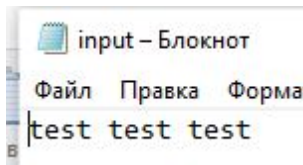
Пример 2

Приветствие и входные данные



```
C:\tmp\lab_1_198\cmake-build-debug\lab_1_198.exe
hello, this program can check how many word in this sentence have on first position have a certain letter
if you want start you need write name of file in, on next line name of file out, and on next line write letter
C:\tmp\courseTes\input.txt
C:\tmp\courseTes\output.txt
t
```

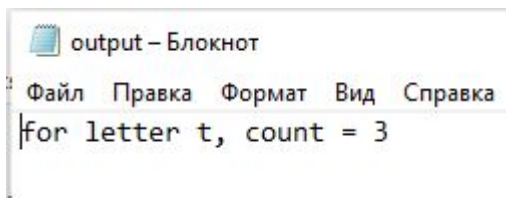
Файл ввода



```
input - Блокнот
Файл  Правка  Формат
test test test
```

Здесь я ожидаю увидеть ответ 3

Результат работы



```
output - Блокнот
Файл  Правка  Формат  Вид  Справка
for letter t, count = 3
```

Выводы.

В ходе работы над программой были освоены приемы работы с файлами и их библиотеками, служащими для ввода вывода. Получены навыки разработки алгоритмов, а также улучшены навыки работы с модельным представлением строки с маркером.