

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
ФГБОУ ВО «АЛТАЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт цифровых технологий, электроники и физики
Кафедра вычислительной техники и электроники (ВТиЭ)

Отчет по лабораторной работе № 0
Работа с текстовыми данными.

Выполнил: студент 595 гр.

_____ А.В. Лаптев

Проверил: доц. каф. ВТиЭ

_____ П.Н. Уланов

Лабораторная работа защищена

«__» _____ 2022 г.

Оценка _____

Барнаул 2022

Постановка задачи:

В рамках данной лабораторной работы мне необходимо было реализовать консольное приложение для реализации следующего функционала:

1. Просмотр текстового файла с кодировкой UTF-8 построчно с вертикальной промоткой при достижении конца экрана; строки выводятся на ширину окна, выступающие части обрезаются; ширина окна в символах – 80, количество строк – 24 (вариант 1).
2. Поиск слова/фразы в файле, вывод всех упоминаний с номером строки и номером символа в строке, с которого начинается упоминание искомого (вариант 6).

Описание алгоритма:

Вся реализация была разделена на 3 блока:

1. Основной блок (функция main).

В этом блоке реализовано простое меню для выбора того варианта задания, который захочет выполнить пользователь. Помимо этого, в данном блоке выполняется преобразование исходного файла для того, чтобы получить каждую отдельную его строку, которая затем помещается в вектор типа string.

Это сделано для более удобного подсчета номера строки, который используется в функции searchTextInFile. В данном случае для разбиения на строки использовался буфер, а также функция getLine, которая входит в библиотеку fstream, необходимую для работы с файлами.

2. Блок для просмотра текстового файла построчно (функция readFile).

В этом блоке реализован весь необходимый функционал для просмотра исходного текстового файла. Работа пользователя с файлом заключается в пролистывании файла при помощи стрелок “вверх” и “вниз”, кроме того, предусмотрен выход из программы по нажатию клавиши “Esc”. Обработка нажатия клавиш осуществляется с помощью метода _getch() (является частью заголовочного файла conio.h). Для того, чтобы обрезать строку до 80 символов (ширина окна) используется функция substr для работы со строками.

Вывод текста происходит блоками по 23 строки (на 24 строке всегда находится только каретка, поскольку, после вывода каждой строки внутри цикла осуществляется переход на следующую строку). В программе предусмотрена обработка граничных случаев (для вывода самого первого и самого последнего блоков с текстом) при нажатии клавиш стрелок.

3. Блок для поиска слова/фразы в файле (функция searchTextInFile).

В данном блоке реализована возможность поиска по файлу определенного слова/фразы. Здесь основную задачу выполняет функция `find`, которая является частью библиотеки `algorithm`.

Эта функция внутри цикла, который проходит по всем элементам вектора (отдельным строкам файла), находит подстроку, заданную пользователем с клавиатуры, в каждом элементе. Если таковая имеется, то пользователю выводится номер элемента в векторе (непосредственно номер строки в тексте, в которой находится это слово/фраза), а также номер символа с которого начинается это слово/фраза (подстрока).

Программный код:

```
#include <iostream>
#include <fstream>
#include <vector>
#include <algorithm>
#include <conio.h>

using namespace std;

void readFile(vector <string> repeatText)
{
    system("mode con cols=80 lines=24");    // Задаем размер
    консольного окна

    int numberKey;

    string s;
    for (size_t firstDisplay = 0; firstDisplay < 23; firstDisplay++)
    {
        s = repeatText[firstDisplay];
        cout << s.substr(0, 80) << endl;
    }

    for (size_t countString = 0; countString < repeatText.size() -
22; countString++)
    {
        numberKey = _getch();
        if (numberKey == 27)    // Завершаем по нажатию Esc
            return;
        numberKey = _getch();
        system("cls");    // Очищаем дисплей
        switch (numberKey)
        {
```

```

        case 80:
            // Нажатие стрелки вниз
            if (countString == repeatText.size() - 24) // Если это
последний блок
            {
                cout << countString << endl;
                for (size_t firstDisplay = countString; firstDisplay
< countString + 24; firstDisplay++)
                {
                    // Выводим строки блоками по 23 - количество
строк в окне (минус одна пустая)
                    s = repeatText[firstDisplay];
                    cout << s.substr(0, 80) << endl;
                }
                countString--;
            }
            else if (countString == 0) // Если это самый первый
блок
            {
                for (size_t firstDisplay = countString; firstDisplay
< countString + 24; firstDisplay++)
                {
                    s = repeatText[firstDisplay];
                    cout << s.substr(0, 80) << endl;
                }
            }
            else // Все остальные блоки
            {
                countString++;
                for (size_t firstDisplay = countString; firstDisplay
< countString + 23; firstDisplay++)
                {
                    s = repeatText[firstDisplay];
                    cout << s.substr(0, 80) << endl;
                }
                countString--;
            }
            break;
        case 72:
            // Нажатие стрелки вверх
            if (countString == 0) // Первый блок
            {
                for (size_t firstDisplay = countString; firstDisplay
< countString + 23; firstDisplay++)
                {

```

```

        s = repeatText[firstDisplay];
        cout << s.substr(0, 80) << endl;
    }
    countString--;
}
else // Все остальные блоки
{
    countString--;
    for (size_t firstDisplay = countString; firstDisplay
< countString + 23; firstDisplay++)
    {
        s = repeatText[firstDisplay];
        cout << s.substr(0, 80) << endl;
    }
    countString--;
}
break;
default:
    break;
}
}
}

void searchTextInFile(vector <string> repeatText)
{
    cout << "Find text: ";
    string findText;
    cin >> findText;

    for (size_t countString = 0; countString < repeatText.size();
countString++)
    {
        string s = repeatText[countString];
        int resultFind = 0;

        for (resultFind = s.find(findText, resultFind++); resultFind
!= string::npos; resultFind = s.find(findText, resultFind + 1))
            // Находим все совпадения подстроки для каждого элемента
вектора

            cout << "Номер строки: " << countString + 1 << " Номер
элемента: " << resultFind + 1 << endl;
    }
}

```

```

int main()
{
    setlocale(0, "");

    int number;
    string filename = "test.txt";
    do
    {
        cout << "Введите номер задания, которое надо выполнить:" <<
endl;

        cout << "1. Просмотр текстового файла." << endl;
        cout << "2. Поиск слова/фразы в текстовом файле." << endl;

        cin >> number;
    } while (number < 1 or number > 2);

    char buffer[2048];
    vector <string> repeatText;
    ifstream inputFile(filename);
    if (!inputFile.is_open())
    {
        cout << "File not found" << endl;
    }
    else
    {
        while (!inputFile.eof())
        {
            // Проходимся по файлу для подсчета количества строк
            // Каждую новую строку добавляем в вектор
            inputFile.getline(buffer, 2048, '\n');
            repeatText.push_back(buffer);
        }
    }
    inputFile.close();

    switch (number)
    {
    case 1:
        readFile(repeatText);
        break;
    case 2:
        searchTextInFile(repeatText);
        break;
    default:

```

```
        break;
    }
    system("pause");

    return 0;
}
```