МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное автономное образовательное учреждение

высшего образования

**«Национальный исследовательский**

**Нижегородский государственный университет им. Н.И. Лобачевского»**

**Институт информационных технологий, математики и механики**

**ОТЧЕТ**

по учебной практике

**Иерархический связный список**

**Выполнил:** студент группы 381906-3  
Захаров А. А. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Нижний Новгород

2020

**Содержание**

[Введение 3](#_Toc59532060)

[1 Постановка задачи 4](#_Toc59532061)

[2 Руководство пользователя 5](#_Toc59532062)

[3 Руководство программиста 8](#_Toc59532063)

[3.1 Описание структуры программы 8](#_Toc59532064)

[3.2 Описание структуры данных 8](#_Toc59532065)

[3.3 Описание программной реализации 8](#_Toc59532066)

[Заключение 10](#_Toc59532067)

[Список литературы 11](#_Toc59532068)

# Введение

Данная работа направлена на изучение способов программной обработки иерархических списков. Эта программа должна выполнять определенные операции над текстом (такие как разделять файлы с текстом, разделять их уже на строки, а их на слова). Основная цель данной лабораторной работы – практическое применение таких структур хранения данных, как иерархический связный список. В ходе выполнения лабораторной работы разрабатывается общая форма представления иерархических связных списков, разрабатывается программа для работы с одним из них, а также создаются функции для работы с файлами (.txt разрешения).

# Постановка задачи

*Цель:*

Разработать и реализовать программу, способную сначала открывать .txt файлы и считывать данные из них, а затем выводить определенные куски текста (весь текст, строчки, слова), а также изменять этот текст. Для написания программы используется язык C++.

*Исходные данные:*

Выбирается в меню определенный пункт и по необходимости дописываются дополнительные уточнения, такие как номер книги, строки и слова или же названия .txt файла.

*Требуемый результат:*

Правильное выполнение функций, которые были выбраны в меню.

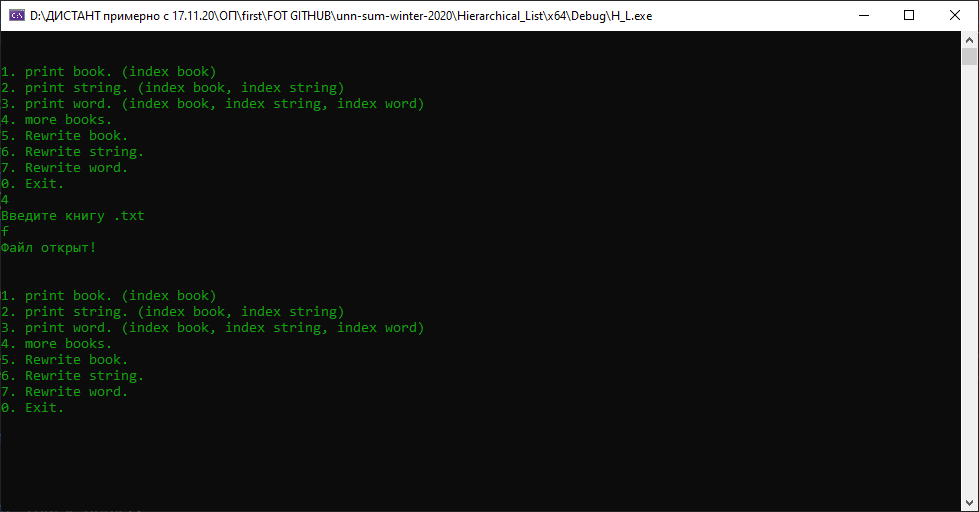


Рис. 1. Пример работы программы

# Руководство пользователя

1. Для начала работы программы необходимо открыть .txt файл. На [рис. 1](#Пример_работы_программы) хорошо показано данное действие.
2. Далее в меню нужно выбрать номер операции для дальнейших действий. В большинстве случаев нужно ввести ещё дополнительную информацию. Например, индекс книги, строки, слова. (Индекс в данной реализации начинается с 1).
3. Пользователю выводится текст или пользователь вводит новый текст, который сохраняется в файл.
4. Далее можно снова выбрать в меню операцию. Однако после изменения текста все с ним связанное удаляется из самой программы, поэтому нужно снова его добавить как в [пункте 1](#пункт_1).

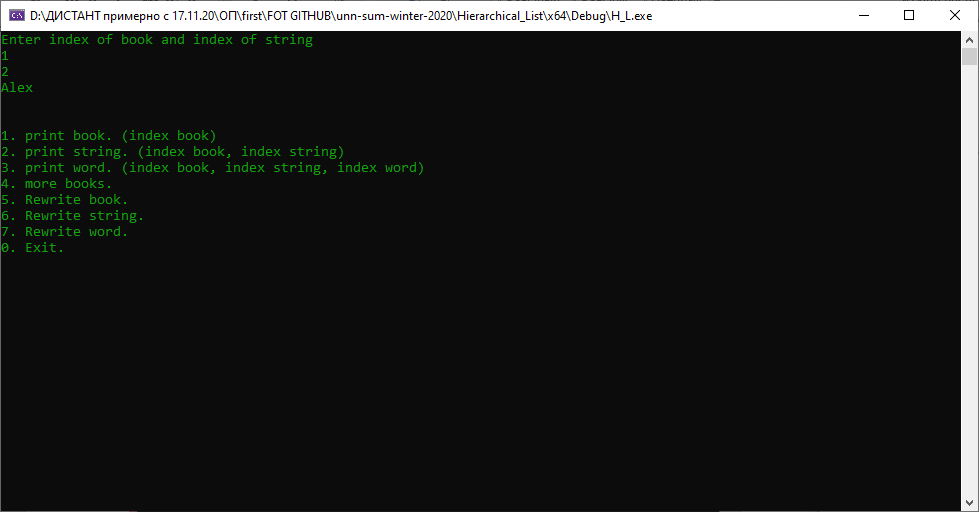


Рис. 2. Пример вывода строки

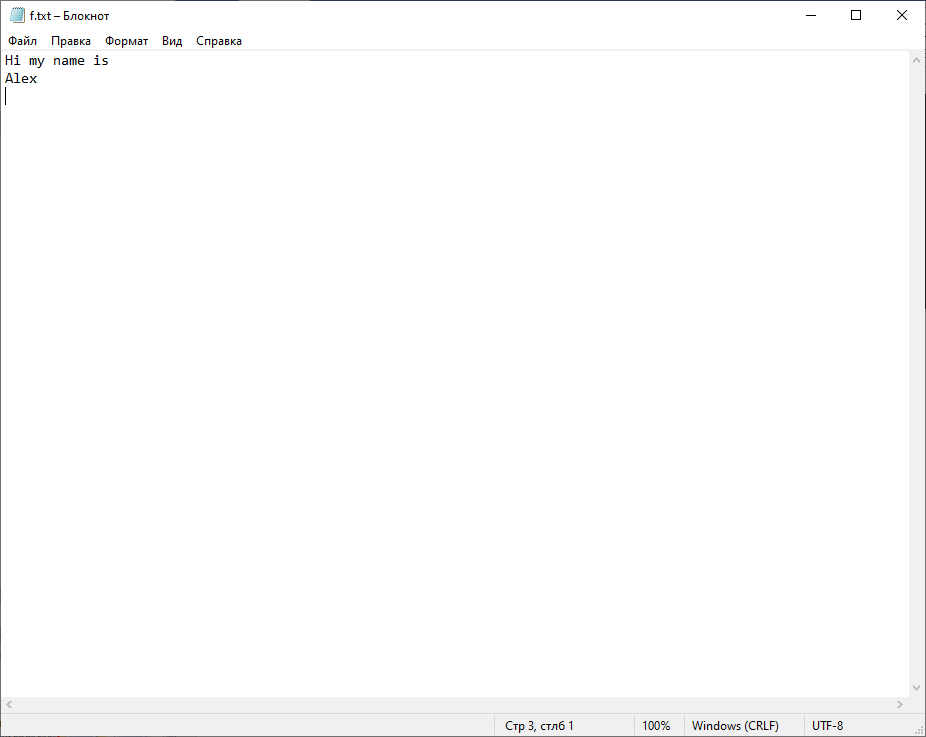


Рис. 3. f.txt файл

# Руководство программиста

## Описание структуры программы

Программа состоит из 1 проекта

**Hierarchical\_List**

Source.cpp - файл, в котором реализовано все, что нужно для выполнения этой задачи.

f.txt, s.txt, HP.txt, myStory.txt – файлы в которых хранится текст.

## Описание структуры данных

В программе реализован иерархический связный список, который имеет указатель на “голову”. Также реализован класс Node, в котором реализован указатель на следующий Node и указатель на нижний Node, а также хранилище для строчки.

**Алгоритмы.**

Вводится с клавиатуры название .txt файла, далее этот файл открывает и из него считываются данные, потом файл закрывается. Весь принятый текст записывается в один из главных Node-ов (head, head->pNext и так далее). Далее текст разбивается на строчки, а они уже на слова.

*Алгоритм разбиения на строки.*

Алгоритм осуществляется с помощью прохождения всей строки в поисках ‘\n’. При нахождении весь текст до него считается за строчку, далее снова идет в поисках этого символа и так далее, пока не будет конец всего текста.

*Алгоритм разбиения на слова.*

Строчка разбивается на слова так же, как весь текст разбивается на строчки, только ищется символ ‘ ‘ (пробел).

## Описание программной реализации

**Node:**

*Поля:*

Node\* pNext; - указатель на следующий Node

Node\* pDown; - указатель на нижний Node

string data; - данные

*Методы:*

Node(string data = string(), Node\* pNext = nullptr, Node\* pDown = nullptr); - конструктор по умолчанию

**Hierarchy:**

*Поля:*

int size\_book; - кол-во книг

int size\_string; - кол-во строк

int size\_words; - кол-во слов

Node\* head; - указатель на Node

*Методы:*

hierarchy(); - конструктор по умолчанию

~hierarchy(); - деструктор

void push\_back(string data); - распределения текста по Node-ам, также создание этих Node-ов

string print\_book(int book); - вывести определенную книгу

string print\_string(int book, int string); - вывести определенную строчку

string print\_word(int book, int stringg, int word); - вывести определенное слово

int get\_size\_books(); - вернуть кол-во книг

int get\_size\_string(int index\_book); - вернуть кол-во строк в книге

int get\_size\_word(int index\_book, int index\_string); - вернуть кол-во слов в строке в книге

void changer\_book(int index\_book); - изменить всю определенную книгу

void changer\_string(int index\_book, int index\_string); - изменить определенную строку в определенной книге

void changer\_word(int index\_book, int index\_string, int index\_word); - изменить определенное слово в определенной строке в определенной книге

*Классы:*

class Node – класс Node

**Остальное:**

stack<string> names; - хранилище имен (названий .txt файлов)

string open\_file(string path); - открыть файл и считать все данные (возвращает всю книгу)

void open\_file\_rewrite(string new\_string, int index\_b); - открыть файл, чтобы его переписать

void open\_file\_add(string new\_string, int index\_b); - открыть файл, чтобы дописать

hierarchy list; - объект иерархического списка

void plus\_books(); - добавление книги (для сокращения кода)

void work\_with\_hierarchy(); - интерфейс (для работы из командной строки)

# Заключение

В лабораторной работе была реализована программа, выполняющая работу с .txt файлом: взятие текста, изменение этого текста. Хранение текста было реализовано с помощью иерархического связного списка. Программа выполняет все необходимые операции над текстом (изменение всего текста, строки, слова).

# Список литературы

1. Лабораторный практикум. Составители: Барышева И.В., Мееров И.Б., Сысоев А.В., Шестакова Н.В. Под редакцией Гергеля В.П. Учебно-методическое пособие. – Нижний Новгород: Нижегородский госуниверситет, 2017