

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение
высшего образования

«Национальный исследовательский

Нижегородский государственный университет им. Н.И. Лобачевского»
(ННГУ)

Институт информационных технологий, математики и механики

Кафедра математического обеспечения и суперкомпьютерных
технологий

ОТЧЕТ ПО УЧЕБНОЙ ПРАКТИКЕ

«Создание и обработка текста»

Выполнил: студент группы
381706-2

Крюков Дмитрий Алексеевич

_____ Подпись

Научный руководитель:

ассистент каф. МОСТ ИИТММ

Лебедев Илья Геннадьевич

_____ Подпись

Нижний Новгород

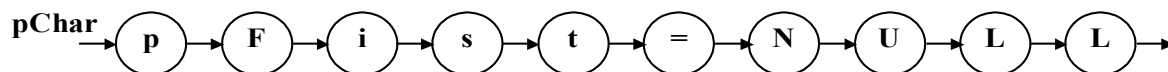
2019

Содержание

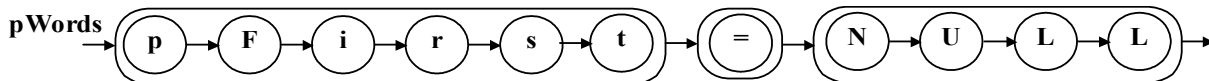
<u>Содержание.....</u>	<u>2</u>
<u>1. Введение.....</u>	<u>3</u>
<u>2. Постановка задачи.....</u>	<u>4</u>
<u>3. Руководство пользователя.....</u>	<u>5</u>
<u>4. Руководство программиста.....</u>	<u>6</u>
<u>4.1 Описание структуры программы.....</u>	<u>6</u>
<u>4.2 Описание структур данных.....</u>	<u>6</u>
<u>4.3 Описание алгоритмов.....</u>	<u>8</u>
<u>5. Заключение.....</u>	<u>9</u>
<u>6. Литература.....</u>	<u>10</u>

1. Введение

Текст – линейная последовательность символов

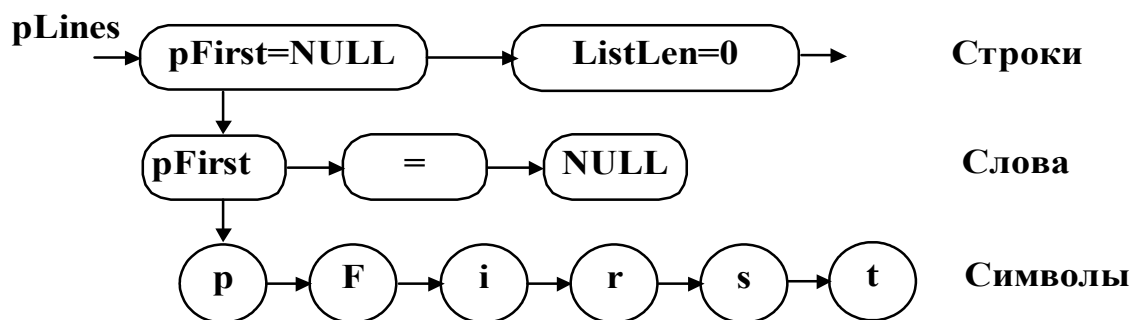


Текст – линейная последовательность слов (слово - линейная последовательность символов)



Текст – линейная последовательность строк, строки состоят из слов, слова – из символов и т.д.

Математическая модель текста – иерархическая структура представления (дерево)

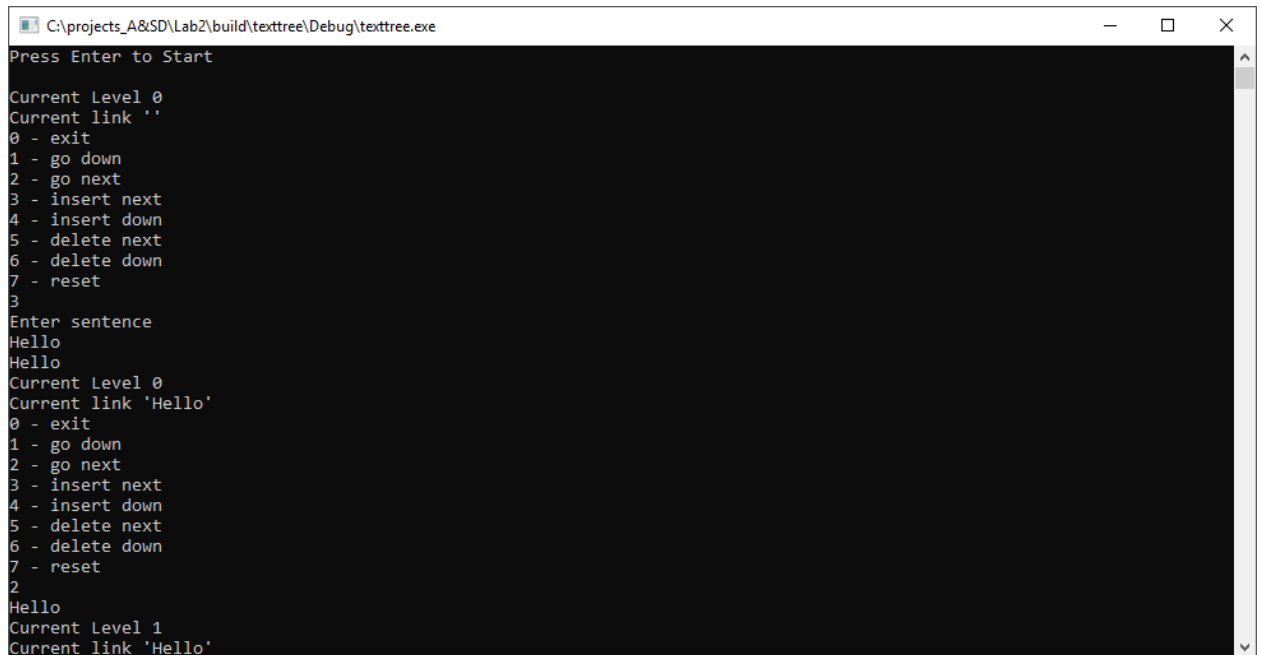


2. Постановка задачи

- Выбор модели представления текста
- Выбор структуры хранения текста
- Реализация
 - схема наследования,
 - навигация,
 - доступ,
 - модификация структуры,
 - алгоритмы обхода,
 - итератор,
 - копирование
- Повторное использование памяти (сборка мусора)

3. Руководство пользователя

Данная программа предназначена для тестирования динамической структуры текст



```
C:\projects_A&SD\Lab2\build\texttree\Debug\texttree.exe
Press Enter to Start

Current Level 0
Current link ''
0 - exit
1 - go down
2 - go next
3 - insert next
4 - insert down
5 - delete next
6 - delete down
7 - reset
3
Enter sentence
Hello
Hello
Current Level 0
Current link 'Hello'
0 - exit
1 - go down
2 - go next
3 - insert next
4 - insert down
5 - delete next
6 - delete down
7 - reset
2
Hello
Current Level 1
Current link 'Hello'
```

пользователю предлагается набор команд :

- переход на уровень ниже
- переход к следующему звену
- вставка после текущего звена
- вставка на уровень ниже
- удаление следующего звена
- удаление звена на уровень ниже
- переход в начало текста

4. Руководство программиста

4.1 Описание структуры программы

1. Модуль `texttreelib` (`TText.h.h`, `TtextLink.h`) – реализация класса текст, звено текстового дерева, память дерева
2. Модуль `texttree(main.cpp)` – реализация программы для тестирования динамической структуры текст
3. Модуль `texttreetest(texttree_test.cpp, textlink_test.cpp)` - тестирование класса текст при помощи Google C++ Testing Framework.

4.2 Описание структур данных

Структура `TextMem`

Поля:

`pFirst` - первое звено

`pLast` - последнее звено

`pFree` - первое свободное

Структура `TTextLink`

Поля:

`pDown` - следующий уровень

`pNext` - соседи

`d` - буква, только на нижнем уровне

`level` - уровень, 0 -текст, 1 -строка 2 - слово, 3 - буква

`TTextMem` мем – память под звенья

Методы:

`InitMemory(int size)` – инициализация памяти

`MemCleaner(TText &txt)` — сборка «мусора»

`operator new(const size_t size)` — перегрузка оператора `new`
`operator delete(void* pM)` — перегрузка оператора `delete`
`operator=(const TTextLink& a)` - присваивание
`operator+=(TTextLink& a)` – добавить звено
`operator+=(char c)`– добавить букву
`operator+=(char* c)` - – добавить строку
`operator<<(ostream& o, const TTextLink& a)` - вывод
`Clone()` - копирование звена

Структура Ttext

Поля:

`pFirst` – первое звено
`pCurrent` – текущее звено
`st` – стек итератора

Методы:

`GoNextLink()` - перейти к следующему звену
`GoDownLink()` - перейти на уровень ниже
`GetCurrent()` - вернуть указатель на текущий
`Reset()` - установить первое звено текущим
`GoNext()` - обход структуры
`IsEnded()` - является ли текущий элемент последним
`InsertNext(char* s)` – вставить строку после текущего
`InsertNext(TTextLink* s)` – вставить звено после текущего
`InsertDown(char* s)` – вставить строку ниже текущего
`InsertDown(TTextLink* s)` – вставить звено ниже текущего
`DelNext()` - удалить звено за текущим
`DelDown()` - удалить звено ниже текущего
`Print()` - печать

4.3 Описание алгоритмов

Вставить звено s после текущего

если (pCurrent == NULL)

ошибка

pN = pCurrent->pNext

pCurrent->pNext = s

s->pNext = pN

Вставить звено s ниже текущего

если (pCurrent == NULL)

ошибка

pN = pCurrent->pDown

если (pN == NULL)

pCurrent->pDown = s;

иначе

пока (pN->pNext != NULL)

pN = pN->pNext;

pN->pNext = s

5. Заключение

В ходе работы был реализован класс текст, в нем реализованы функции навигации, доступа, копирования, а также алгоритмы обхода и итератор

6. Литература

1. Гергель В.П. Методические материалы по курсу «Методы программирования 2», Нижний Новгород, 2015.