

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ**
Федеральное государственное автономное образовательное учреждение
высшего образования
**«Нижегородский государственный университет им. Н.И. Лобачевского»
Национальный исследовательский университет**

**Институт информационных технологий, математики и механики
Кафедра математического обеспечения и суперкомпьютерных технологий**

**ОТЧЕТ ПО УЧЕБНОЙ ПРАКТИКЕ
«Редактирование текстов»**

Выполнил:
студент группы 361706-1
Резанцев Сергей Алексеевич
_____ Подпись

Научный руководитель:
ассистент каф. МОСТ ИИТММ
_____ Лебедев И. Г.

Нижний Новгород
2019 г.

Содержание	
Введение	3
Постановка задачи	5
Руководство пользователя	6
Руководство программиста	7
Описание структуры программы	7
Описание структур данных	7
Описание алгоритмов	9
Заключение	11
Литература	12

1. Введение

Обработка текстовой информации на компьютере широко применяется в различных областях человеческой деятельности: образование, наука, документооборот, кадровый и бухгалтерский учет и др. Вне зависимости от назначения текста типовыми операциями обработки являются создание, просмотр, редактирование и сохранение информации. В связи с тем, что объем текстовой информации может являться очень большим, для эффективного выполнения операций с ней необходимо выбрать представление текста, обеспечивающее структурирование и быстрый доступ к различным элементам текста. Так, текст можно представить в виде линейной последовательности страниц, каждая из которых есть линейная последовательность строк, которые в свою очередь являются линейными последовательностями слов. Такое представление можно осуществлять с любой степенью детализации в зависимости от особенностей прикладной задачи. В рамках лабораторной работы рассматривается задача разработки учебного редактора текстов, в котором для представления данных используется иерархический связный список. Подобная иерархическая структура представления может применяться при компьютерной реализации математических моделей в виде деревьев и, тем самым, может иметь самое широкое применение в самых различных областях приложений. Так можно графически представить себе эту структуру:

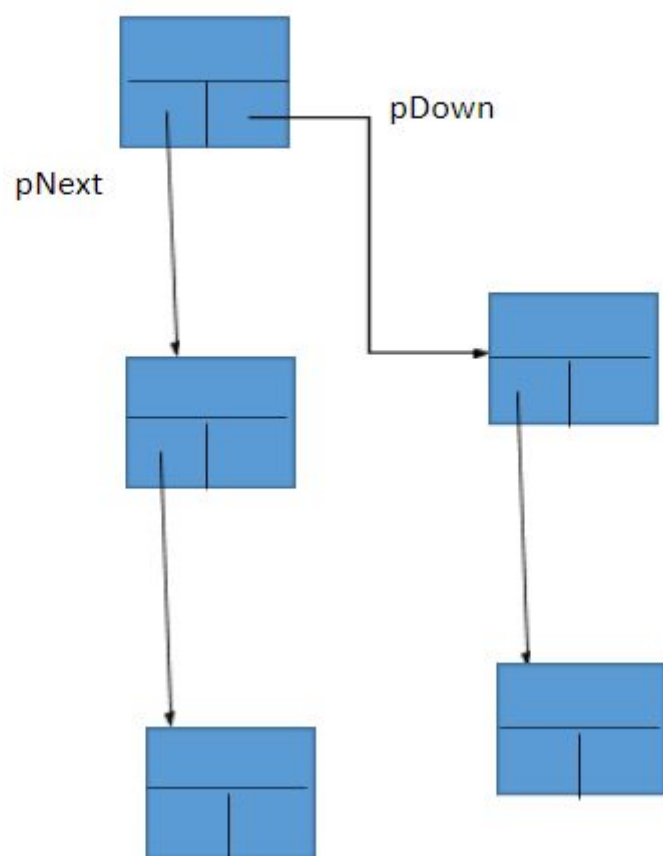


Рис1. Иерархический список

2. Постановка задачи

Необходимо разработать структуру данных представляющую собой иерархический список особым видом работы с памятью, который позволял бы хранить и редактировать текст. Должны быть реализованы функции добавления заголовка, подзаголовка, секции и их удаление, чтение текста из файла и его сохранение в файл.

3. Руководство пользователя

Чтобы начать работу с программой запустите приложение Text.

На экране появится интуитивно понятный интерфейс для работы с текстом:

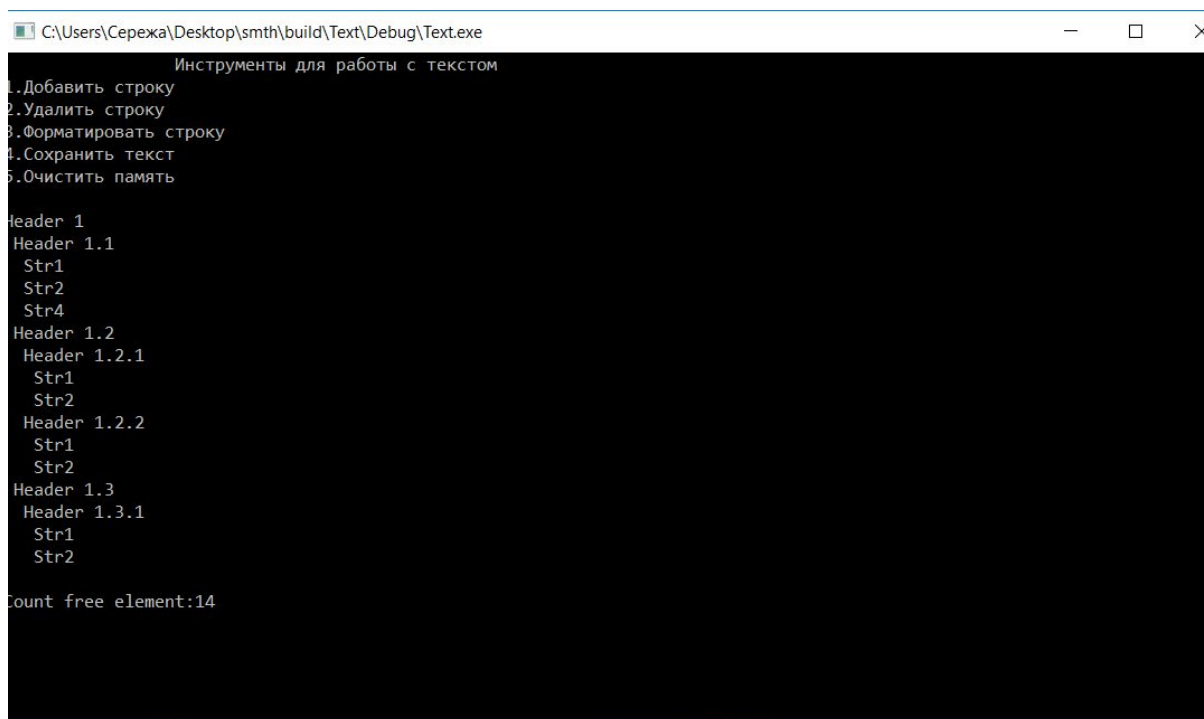


Рисунок 1. Текстовый редактор.

С которым весело, интересно и удобно работать.

4. Руководство программиста

4.1. Описание структуры программы

Список модулей программы

Программа состоит из следующих модулей:

- TLink.h – заголовочный файл класса «список».
- TText.h - заголовочный файл класса «текст».
- TLink.cpp – файл, содержащий реализации функций класса «список».
- TText.cpp – файл, содержащий реализации функций класса «текст».
- main.cpp – файл, содержащий тело программы, которая выполняет задачи поставленные ранее в пункте «Постановка задачи».

4.2. Описание структур данных

Модуль TLink.h

Модуль содержит структуру

struct TMem { TLink*pFirst, *pFree, *pLast;}; - структура для «разметки» памяти.

Модуль содержит class TLink, который содержит поля:

char str[80]; - поле для хранения строки,

TLink*pNext, *pDown; - указатели на следующие элементы списка,

static TMem mem; - поле, служащее для выделения памяти списку,

bool flag; - служебное поле, используемое при очистке памяти.

Класс содержит следующие прототипы функций:

TLink(char*s = 0, TLink*pN = 0, TLink*pD = 0) – конструктор.

char* GetLine() – функция получения поля str.

TLink* GetDown() - функция получения поля pDown.

TLink* GetNext() - функция получения поля pNext.

bool GetFlag() - функция получения поля flag.

void SetFlag(bool f) - функция установки значения flag.

void SetNext(TLink *pN) - функция установки значения pNext.

void SetDown(TLink *pD) - функция установки значения pDown.

void SetLine(char* s)- функция установки значения str.

static void InitMem(size_t s); - функция инициализации памяти.
static void MemClean(TText &t); - функция очистки памяти.
static int PrintFree(TText &t); - функция печати незанятых ячеек памяти.
void* operator new(size_t s); - перегрузка оператора new для выделения памяти.
void operator delete(void*p); - перегрузка оператора delete для удаления памяти.
~TLink(); - деструктор.

Модуль TText.h

Модуль содержит class TText, который содержит поля:

TLink*pFirst, *pCurr; - указатели на первый элемент текста и на текущий элемент,
stack<TLink*> st; - стек,
int level; - служебная переменная, используемая в алгоритме печати текста.
int CountLink - количество списков

Класс содержит следующие прототипы функций:

TText(); - конструктор
int GetCount() - функция получения поля CountLink .
TLink* GetFirst() функция получения поля pFirst.
void SetCurr(char* s) - Вставить строку по указателю pCurr
int GoNextLink(); - Сдвиг указателя pNext вниз
int GoDownLink(); - Сдвиг указателя pDown вниз
int GoPrevLink(); - Возврат сдвинутого указателя на шаг назад
void insNextLine(char*s); - Вставить строку по указателю pNext
void insNextSection(char*s);
void insDownLine(char*s); - Вставить строку по указателю pDown
void insDownSection(char* s);
void DelNext(); - Удалить строку по указателю pNext
void DelDown(); - Удалить строку по указателю pDown
void MarkCurr() { pCurr->SetFlag(true); }
void Reset();
bool IsEnd();
void GoNext();

TLink*ReadRec(istream& file); - Рекурсивное чтение из файла
void Read(char *fn);

void PrintText(TLink *tmp); - Вывод текста в консоль
void Print();
void Save(char* fn);
void SaveText(TLink* tmp, ostream& f); - Сохранение текста в файл.

4.3. Описание алгоритмов

Алгоритм очистки памяти TLink::MemClean(TText &t)

1. Сначала обходим все элементы экземпляра класса TText, передаваемого в функцию и поднимаем у каждого флаг.
2. Обходим все свободные элементы памяти и поднимаем в каждом флаг.
3. Обходим все элементы массива памяти и если флаг не поднят, значит элемент потерян при удалении отцовского, значит добавляем его к свободным элементам.

Алгоритм рекурсивного чтения из файла *TText::ReadRec(istream& file)

Пока не конец файла:

1. Если строка равна '}', то выходим из функции.
2. Иначе если строка равна '{', то tmp->pDown = ReadRec(file).
3. Иначе если first не указывает никуда, то tmp->pNext = new TLink(buf) (создаем элемент, со считанной строкой), tmp = first.
4. Иначе tmp->pNext = new TLink(buf) (Создаем элемент по указателю pNext со считанной строкой) и сдвигаем указатель по pNext.

Возвращаем указатель first.

Алгоритм вывода текста в консоль

void TText::Print()

```
{  
    level = 0;  
    PrintText(pFirst);  
    pCurr = pFirst;  
}
```

void TText::PrintText(TLink* tmp)

```
{  
    if (tmp != NULL)  
    {  
        for (int i = 0; i < level; i++)  
            cout << " ";  
        cout << tmp->GetLine() << endl;  
        level++;  
        PrintText(tmp->GetDown());  
        level--;  
        PrintText(tmp->GetNext());  
    }
```

```
}  
}
```

Идея алгоритма состоит в работе с переменной `lvl`, которая показывает сколько отступов необходимо сделать для правильного форматирования. Переменная `lvl` инкрементируется при рекурсивном вызове `ViewText` с передачей указателя `pDown` и декрементируется при вызове функции с передачей указателя `pNext`.

5. Заключение

В ходе выполнения лабораторной работы был разработан редактор текста.

Предоставлено описание примера работы с текстом в разделе «Руководство пользователя».

Также разработаны и доведены до успешного выполнения тесты, проверяющие корректность методов класса TLink.

6. Литература

1. Васильев А.Н. Самоучитель С++ с примерами и задачами. -СПб.: Наука и Техника, 2016. -480с.
2. Т. А. Павловская С/С++ Программирование на языке высокого уровня. - СПб.:Питер, 2011. - 461 с.
- 3.Крапенко С. Н. и др. Методы объектно-ориентированного программирования. <http://e-learning.unn.ru/course/view.php?id=251>.
- 4.Страуструп. Б. Курс «Язык программирование С++ для профессионалов» <http://www.intuit.ru/studies/courses/98/98/info>
- 5.Гергель В.П. Методические материалы по курсу “Методы программирования 2”:
[<http://www.itmm.unn.ru/files/2018/10/Primer-1.1.-Struktury-hraneniya-mnozhestva.pdf>], 2015.
6. Барышева И.В., Мееров И.Б., Сысоев А.В., Шестакова Н.В. ЛАБОРАТОРНЫЙ ПРАКТИКУМ 2017. - 105 с