Федеральное агентство по образованию Российской Федерации Государственное образовательное учреждение

Высшего профессионального образования Нижегородский государственный университет им. Н.И. Лобачевского

Институт информационных технологий математики механики

Создание и обработка текста

Отчет по лабораторной работе

Выполнил:

студент ИИТММ гр. 381706-2

Антипин А.С.\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Проверил:

ассистент каф. МОСТ, ИИТММ

Лебедев И.Г\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

Нижний Новгород

2018 г.

Введение

Хранение и обработка текста является очень актуальной задачей на протяжении всего существования компьютеров. Линейное хранение массива символьных элементов не является удобным, так как для доступа к последним элементам массива потребуется пройти все предыдущие элементы, а если появляется необходимость вставить какой-либо элемент внутрь массива, то необходимо сдвигать оставшуюся часть, или если текущей выделенной памяти не хватает, выделять её заново. В связи с этим, для эффективного выполнения операций с тестовой информацией необходимо выбрать её представление, обеспечивающее структурирование и быстрый доступ к различным элементам текста.

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.

Постановка целей и задач

Целью лабораторной работы является создание структуры хранения текста типа «TTree» и обработки теста типа «TText» и методов работы с ними, таких как:

• Создание текста или добавление в уже существующий;

* Удаление элементов текста;
* Копирование;

• Поиск по тексту.

Для реализации алгоритмов будет использоваться 3 шаблонных класса:

• TTree;

* TTreeIterator;
* TText.

Для проверки правильности работы этих классов будут написаны тесты с использованием фреймворка Google Test, а также тестовый образец программы, которая будет использует класс

Руководство программиста

Описание структуры программы:

Класс TTree будет реализован с помощью линейного списка элементов дерева. Каждый элемент имеет параметр уровня:

* 1 уровень — уровень текста;
* 2 уровень — уровень строки;
* 3 уровень — уровень слова;
* 4 уровень — уровень буквы.

Все элементы дерева на разных уровнях имеют одинаковую структуру, но только на уровне буквы можно хранить буквы. Остальные уровни служат для правильной связи теста.

Класс TTreeIterator будет осуществлять проход по дереву, используя метод поиска вглубину.

Класс TText будет реализовывать методы связанные с обработкой текста.

Описание алгоритмов:

Перевод из дерева в массив символьного типа:

* Создание стека с указателями на элементы дерева и помещение в него корня.
* Прохождение по дереву и подсчёт количества букв в переводимой строке.
* Создание массива символов необходимого размера и помещение в стек корня дерева.
* Прохождение по дереву, если буква, то она записывается в результирующий массив.
* Возвращение результирующего массива.

Нахождение дерева:

* Создаётся итератор для текста и создаётся указатель на результирующее дерево;
* Считается количество букв в переданной строке и заводится счётчик;
* Пока в итераторе присутствуют элементы:
* Если встречается буква, которая совпадает с первой буквой переданной строки, то запоминается этот элемент дерева;
* Если буква не совпадает с какой либо буквой из переданной строки, то все обнуляется, а в стек итератора вновь заносится данный элемент дерева;
* Если количество совпавших букв совпадает с длинной переданной строки, то цикл заканчивается и метод возвращает результирующий элемент дерева.

Заключение

В заключение можно сказать, что все поставленные цели и задачи были выполнены, то есть у нас получилось создать достаточно функциональный класс для работы с текстом, который по своему принципу работы схож с более профессиональными текстовыми редакторами. Все методы данных классов успешно прошли проверку с помощью фреймворка GoogleTest, а значит в правильности их работы можно не сомневаться.