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

«Структура хранения стек на списках»

Выполнил: студент группы 381706-1
Митягина Дарья Сергеевна
_____ Подпись

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

Нижний Новгород
2018.

Оглавление

1. Введение

В данной лабораторной работе целью является разработка структуры хранения данных стек на списках.

Данная реализация является полезной для экономии времени, т.к. добавление и извлечение элементов из начала списка возможно за время O(1). При добавлении в стек нового элемента elem создается новый узел us, из us устанавливается ссылка на elem, и us добавляется в начало списка. Аналогично с извлечением элемента.

Таким образом, реализация стека на основе связного списка имеет существенное преимущество по сравнению с реализацией на основе массива.

2. Постановка задачи

В данной работе стояла задача разработать следующие элементы:

1. Класс стека на списках TStackUsingList;

2. Набор тестов для класса TStackUsingList с помощью GT;

3. Программа – пример использования реализованного класса стека на списках;

4. Класс исключений TException.

3. Руководство пользователя

При запуске программы пользователь сможет увидеть простейший пример работы со стеком на списках.

С пользователя запрашивается число n, затем создается стек размера n. Далее необходимо ввести элементы стека, после того, как все элементы заданы, полученный стек выводится на экран.

Затем из стека удаляется элемент, полученный в результате этого стек также выводится на экран. Выглядит это следующим образом:

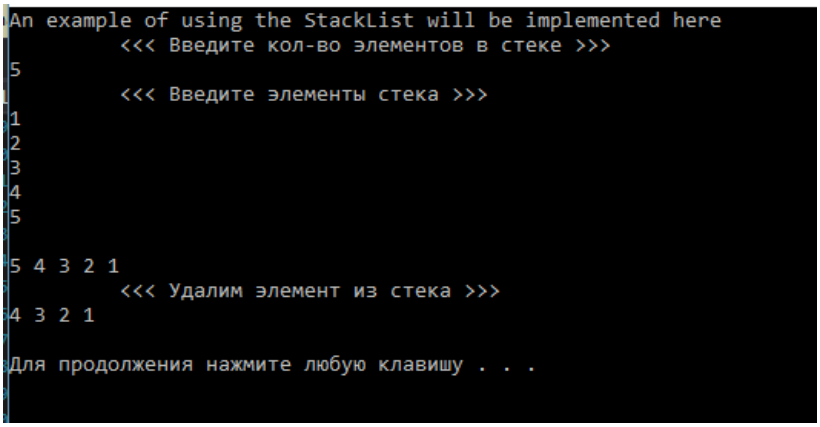

Рис.1 – пример использования стека на списках

4. Руководство пользователя

4.1. Описание структуры программы.

Программа состоит из модулей:

1. StackListLib – статическая библиотека, содержащая файлы StackList.h и StackList.cpp, в которых описан и реализован класс TStackUsingList.

2. ListLib – статическая библиотека, содержащая файлы TList.h, TList.cpp, TElement.h и TElement.cpp, в которых описаны и реализованы классы TElement и TList.

3. Exception – класс исключений.

4. StackListTest – содержит файл test.cpp с тестами для класса TStackUsingList (16 тестов).

5. StackList– содержит файл StackList.cpp, в котором реализован простой пример использования стека.

4.2. Описание структур данных.

Класс TStackUsingList

Является шаблонным классом, наследуется от TList.

Рассмотрим protected-часть:

1. int st_size – максимальный размер стека на списках.

Рассмотрим public-часть:

1. TStackUsingList <T>(int _size = 5) - конструктор;
2. TStackUsingList <T>(TStackUsingList<T> &A) – конструктор копирования;
3. ~TStackUsingList() - деструктор;
4. T Get() – получить значение с вершины стека;
5. void Del() – удалить значение с вершины стека;
6. void Put(T A) – положить значение на вершину стека;
7. void Show() – вывод содержимого стека на экран;
8. int GetSize() – получить размер (количество элементов);
9. int GetMSize() – получить максимальный размер;
10. bool IsFull() – проверка на полноту;
11. bool IsEmpty() – проверка на пустоту;

4.3. Описание алгоритмов

В данной части не будут рассматриваться тривиальные методы, внимание уделим лишь некоторым.

1. Добавление элемента в стек.

Так как разработана структура хранения данных стек, то рассуждать будем следующим образом:

При добавлении нового элемента в стек он размещается на вершине стека, в таком случае удобно использовать списки. Тогда вершина будет являться первый элемент списка.

2. Изъятие элемента из стека

Аналогично.

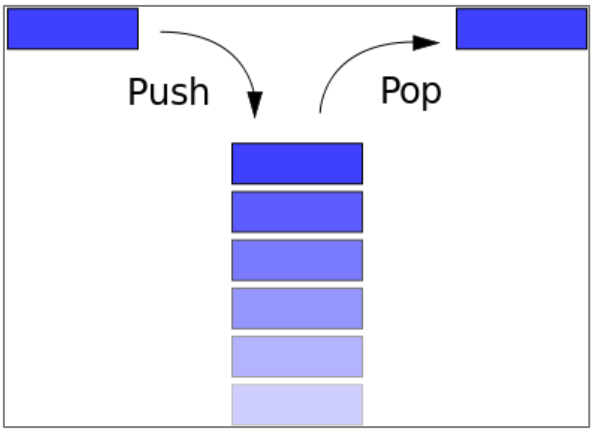

Рис.2 – принцип работы методов добавления и изъятия элемента для стеков

5. Заключение

В процессе работы над данной лабораторной работой мне удалось разработать такую структуру хранения данных как стек на списках.

Кроме того, было достигнуто более глубокое понимание принципов работы с данной структурой, принципов тестирования программы с помощью GT.

6. Литература

1. Гергель В.П. Методические материалы по курсу «Методы программирования 2», 2015. 2. Статья, посвященная теме списков [https://prog-cpp.ru/data-ols/]

2. Статья, посвященная теме списков [https://tim4ous.com/realizatsiya-odnosvyaznogo-lineynogo-spiska-v-si/]

3. Статья, посвященная теме списков [https://prog-cpp.ru/data-ols/]

4. Ахо А., Хопкрофт Д., Ульман Д. Структуры данных и алгоритмы.: Пер. с англ. М.: Издат. дом «Вильямс», 2000. С. 58–76

5. Статья о стеке в Викиконспектах [https://neerc.ifmo.ru/wiki/index.php?title=%D0%A1%D1%82%D0%B5%D0%BA]