

**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
«Реализация стека на списке»**

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

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

Нижний Новгород
2018 г.

Содержание	
Введение	3
Постановка задачи	4
Руководство пользователя	5
Руководство программиста	6
Описание структуры программы	6
Описание структур данных	6
Описание алгоритмов	7
Заключение	8
Литература	9

1. Введение

Стек на списках — структура данных, представляющая собой упорядоченный набор элементов одного типа, фиксированного количества, связанных между собой последовательно посредством указателей, организованных по принципу LIFO (англ. last in — first out, «последним пришёл — первым вышел»). Каждый элемент стека на списках имеет указатель на следующий элемент. Последний элемент стека на списках указывает на NULL. Элемент, на который нет указателя, является первым (головным) элементом списка. В стек на списках все новые элементы добавляются в начало. Вершиной стека на списках является первый элемент в списке.

2. Постановка задачи

Реализовать класс TSList для работы со стеками на списках. Поля классов должны быть закрыты.

В классах обязательно должны присутствовать методы:

- 1) добавление элемента в начало
- 2) удаление элемента в начале

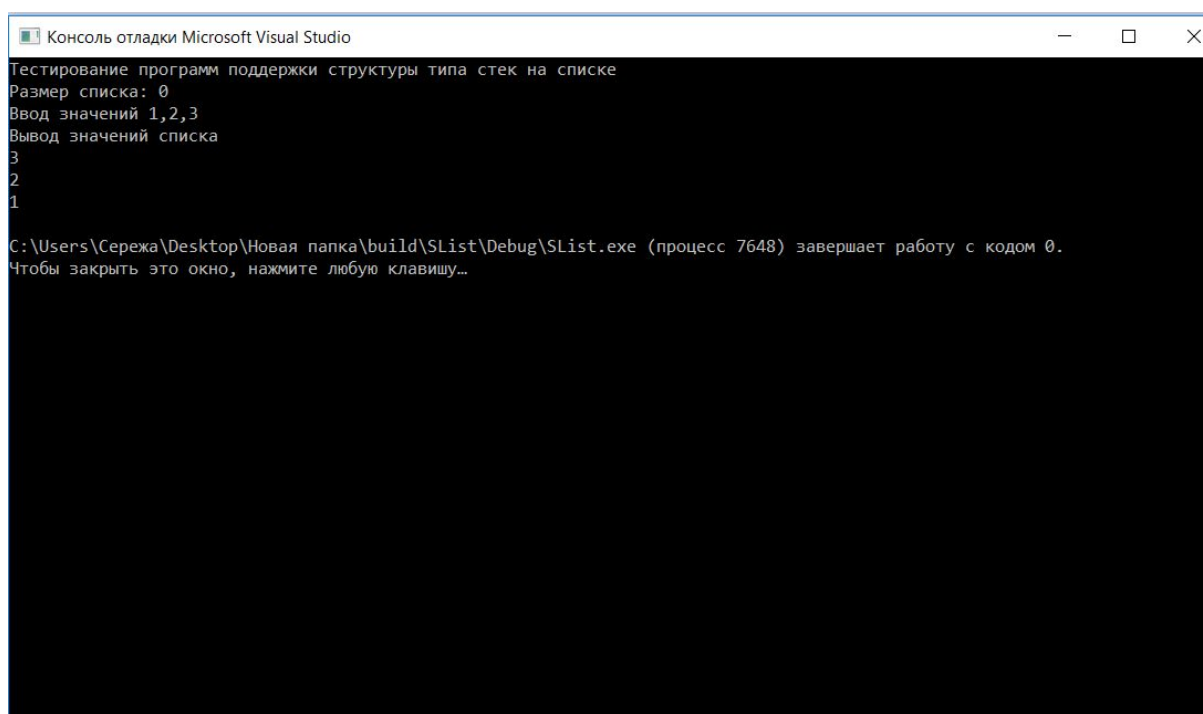
Должны быть реализованы конструкторы: копирования и инициализатор.

Предоставить пример использования и обеспечить работоспособность тестов, покрывающих все методы класса TSList.

3. Руководство пользователя

Чтобы начать работу с программой запустите приложение SList.

На экране появится следующее:



```
Консоль отладки Microsoft Visual Studio
Тестирование программ поддержки структуры типа стек на списке
Размер списка: 0
Ввод значений 1,2,3
Вывод значений списка
3
2
1
C:\Users\Сергея\Desktop\Новая папка\build\SList\Debug\SList.exe (процесс 7648) завершает работу с кодом 0.
Чтобы закрыть это окно, нажмите любую клавишу...
```

Рисунок 1. Результат программы, выведенный на консоль.

Затем программа завершится.

4. Руководство программиста

4.1. Описание структуры программы

Для реализации алгоритмов будут использованы классы TList и TSList.

Лабораторная работа состоит из следующих модулей:

ListLib

Библиотека, содержащая заголовочный файл TList.h, в котором содержится класс TList и реализация его методов, и файл TList.cpp

SListLib

Библиотека, содержащая заголовочный файл TAlis.h, наследник класса TList, в котором содержится класс TAlis, и реализация его методов, и файл TAlis.cpp

SList

Пример использования программы.

test

В файле test_slist.cpp прописаны тесты, покрывающие каждый метод класса TSList.

4.2. Описание структур данных

В программе описан шаблонный класс

TSList:

В нем 1 поле:

int size;

И реализованы следующие функции:

TList(int _size = 10); - конструктор по умолчанию

TList<T>(TList<T> &A); - конструктор копирования

~TList(); -деструктор

void Put(T A); - положить элемент в начало списка

int GetSize(); - узнать размер списка

int GetMaxSize(); - узнать максимальный размер списка

T Get(); - взять элемент из начала списка

bool IsEmpty(); - проверка на пустоту

bool IsFull(); - проверка на полноту

4.3. Описание алгоритмов

Добавление и изъятие элемента в стеке на списках.

При добавлении элемента в стек, он размещается на вершине стека. При изъятии элемента из стека берется элемент, находящийся на вершине стека. Поскольку в данной работе стеки хранятся на списках, то целесообразно вершиной стека считать первый элемент списка. То есть добавление и изъятие элементов происходит только из начала списка. В этом случае сложность этих операций составит $O(1)$. Иначе, для добавления или изъятия элементов из конца списка, пришлось бы обходить все список поэлементно, и сложность операций составила $O(n)$.

5. Заключение

В ходе выполнения лабораторной работы была разработана библиотека, реализующая шаблонный класс стека на списке. Она позволяет при работе со стеком выполнять базовые операции извлечения/добавление элементов стека.

Предоставлено описание примера работы со стеком в разделе «Руководство пользователя».

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

6. Литература

1. Васильев А.Н. Самоучитель C++ с примерами и задачами. -СПб.: Наука и Техника, 2016. -480с.
2. Т. А. Павловская C/C++ 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 C++ 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.