МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное автономное образовательное учреждение

высшего образования

**«Национальный исследовательский**

**Нижегородский государственный университет им. Н.И. Лобачевского»**

**Институт информационных технологий, математики и механики**

**ОТЧЕТ**

по учебной практике

**Стек, список, стек на списке, польская запись**

**Выполнил:** студент группы 381906-3  
Захаров А. А. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Нижний Новгород

2021

**Содержание**

[**1** **Введение** 3](#_Toc61091947)

[**1.** **Постановка задачи** 4](#_Toc61091948)

[**2** **Руководство пользователя** 6](#_Toc61091949)

[**3** **Руководство программиста** 6](#_Toc61091950)

[3.1 Описание структуры программы 6](#_Toc61091951)

[3.2 Описание структуры данных 6](#_Toc61091952)

[3.3 Описание программной реализации 7](#_Toc61091953)

[**Заключение** 9](#_Toc61091954)

[**Список литературы** 10](#_Toc61091955)

# **Введение**

Данная работа направлена на изучение способов программной обработки стеков, стеков на списках, самих списков и польской записи. Каждая программа должна выполнять определенные действия: хранить данные особенным способом или же работать с входящими данными. Основная цель данной лабораторной работы – понять на практике в чем плюсы и минусы каждого из способов хранения, а также разобраться с польской записью. В ходе выполнения лабораторной работы разрабатывается общая форма представления стеков, стеков на списках, списков. Также разрабатываются функции нужные для работы польской записи.

# **Постановка задачи**

*Цель:*

Разработать программы способные имитировать минимальную деятельность стека, стека на списке, списков, а также разработать программу способную проверять правильность заданных выражений, переводить их в польскую запись и высчитывать их. Для написания программы используется язык C++.

*Исходные данные:*

В самих программах уже записаны все нужные операции, поэтому пользователю ничего не надо вводить.

*Требуемый результат:*

Правильность выполнения всех функций.

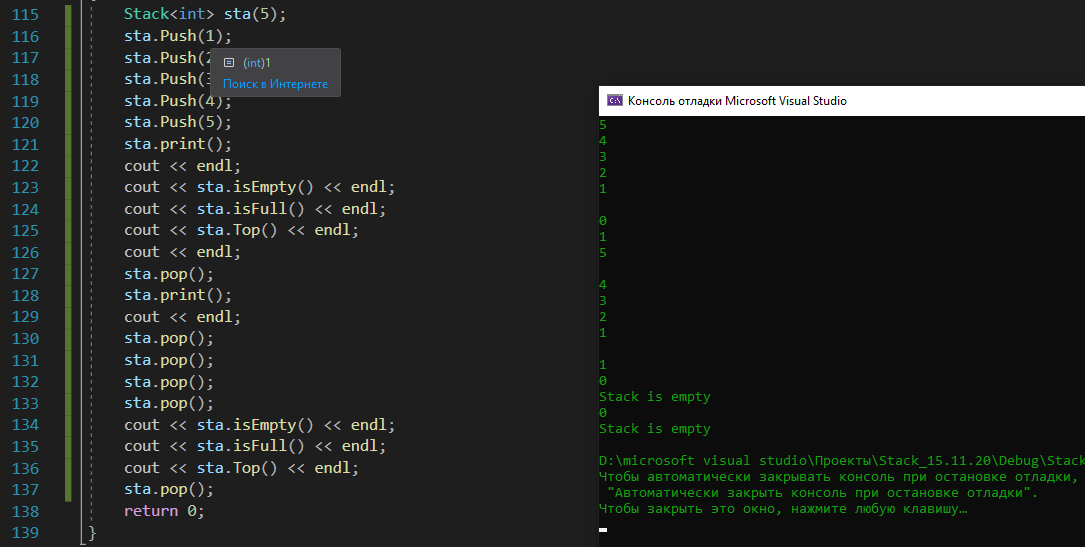


Рис. 1. Пример работы стека

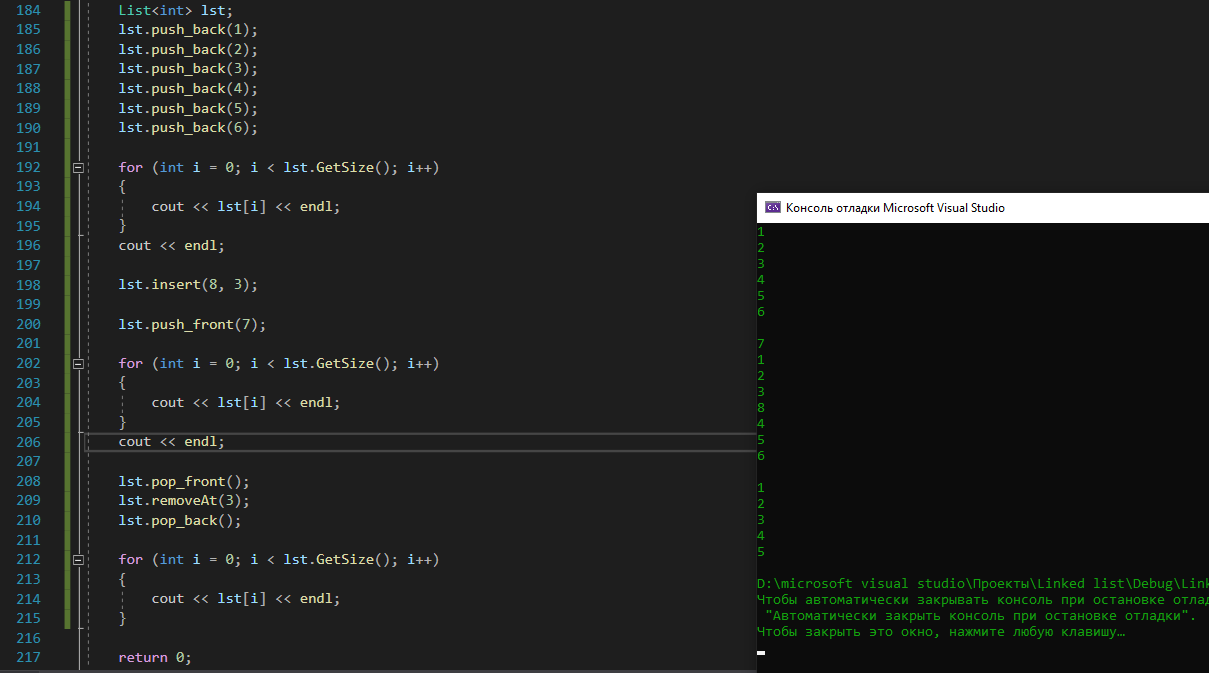


Рис. 2. Пример работы листа

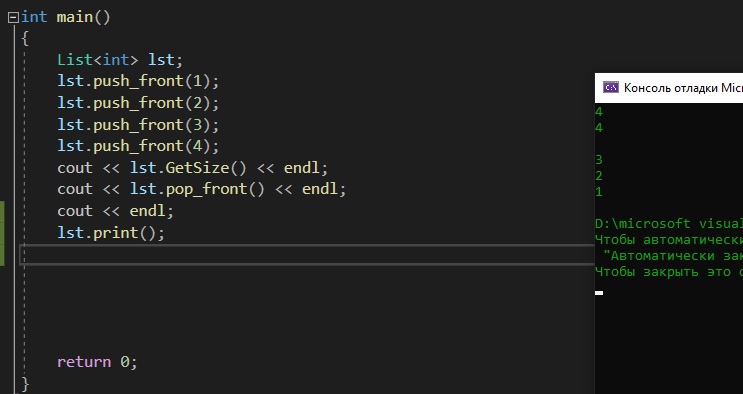


Рис. 3. Пример работы стека на листе

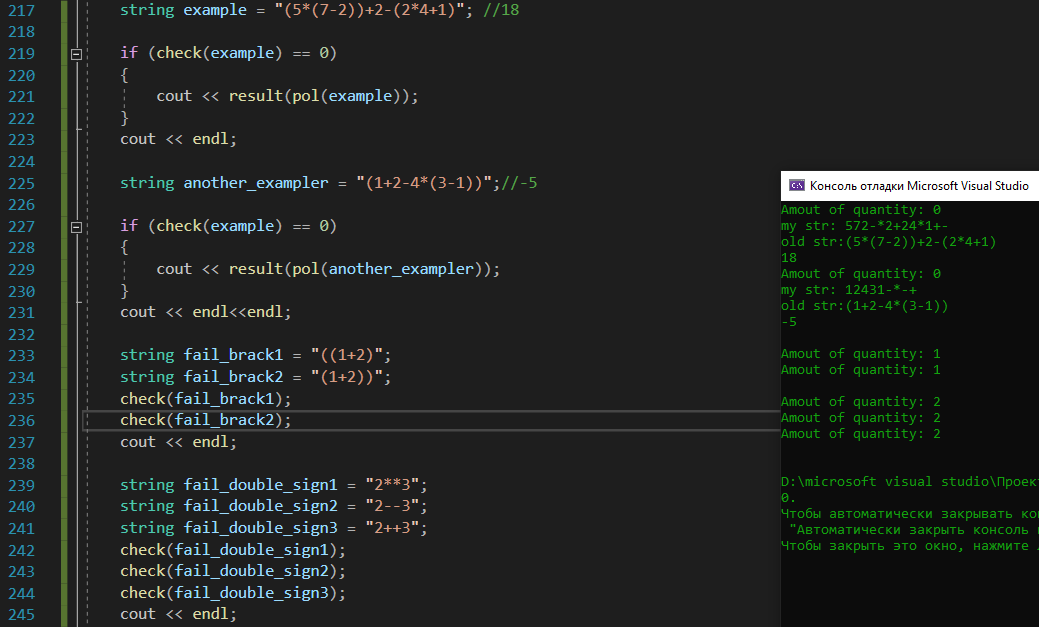


Рис. 4. Пример работы польской записи

# **Руководство пользователя**

* 1. Для начала работы программы её нужно включить.
  2. Дальше выводится результаты, которые показаны на Рис. 1, Рис. 2, Рис. 3, Рис. 4.

# **Руководство программиста**

## Описание структуры программы

Каждой программе свой проект, поэтому имеется 4 проекта.

1. **Стек**

Source.cpp – файл, в котором реализован стек

1. **Стек на списке**

Source.cpp – файл, в котором реализован стек на списке

1. **Список**

Source.cpp – файл, в котором реализован список

1. **Польская запись**  
   Source.cpp – файл, в котором реализована польская запись

## Описание структуры данных

Стек реализован с помощью динамического массива. Список имеет в себе реализацию класса Node, в котором хранится указатель на такой же тип и определенные данные. Также в списке есть указатель на первый Node (head), начало списка. Стек на списке – это тот же список, но без некоторых методов. Польская запись реализована с помощью трех функций. Первая функция должна проверять правильность выражения, вторая переводить её в польскую запись, и третья должна высчитывать результат с помощью стека.

**Алгоритмы.**

Алгоритмы у польской записи.

*Алгоритм проверки правильности введенного выражения.*

Подается строка, которая проходится слева-направо. Считается количество открывающихся и закрывающихся скобок. Также смотрится если ли рядом со знаком еще знак или происходит ли деление на ноль (но только если там фигурирует само деление и ноль (1-1) не считается за ноль).

*Алгоритм перевода в польскую запись.*

Из первоначальной строки копируются в новые числа. Пока программа не дойдет до закрывающиеся скобки. Все знаки при этом действии отправляются в стек. После определенных манипуляций над умножение происходит вывод из стека в новую строчку. Данное действие происходит только, когда программа доходит до закрывающиеся скобки.

*Алгоритм счета с помощью стека.*

Данный алгоритм взят с ютуб ролика <https://www.youtube.com/watch?v=RyylxcuJ6LE&t=132s>. Там лучшее объяснение.

## Описание программной реализации

**Стек:**

Методы:

Stack(int size); -конструктор по умолчанию

~Stack();- деструктор

void Push(T data); - добавление элемента

void pop();- удаление элемента

T Top();- функция возвращает верхний элемент стека

bool isEmpty();- проверка на пустоту

bool isFull();- проверка на полноту

void print();- вывод всех элементов

Поля:

T\* arr; - шаблонный динамический массив

int Size; - целочисленное число, которое будет отвечать за размер массива

int top; - индекс в стеке  
**Лист:**

Методы:

List();- конструктор по умолчанию

~List();- деструктор

void push\_back(T data); - добавление элемента в конец

int GetSize();- функция возвращает количество элементов

T& operator[](const int index); -оператор скобок[]

void pop\_front();-удаление элемента сначала

void clear();-удаление всех элементов

void push\_front(T data); -добавление элемента в начало

void insert(T data, int index); -добавление элемента в какой-то определенный индекс

void removeAt(int index); -удаление элемента по определенном индексу

void pop\_back();-удаление элемента с конца

Поля:

int Size; -целочисленное число, которое будет отвечать за количество элементов

Node<T> \*head; -указатель на первый элемент

Класс:

class Node{};-содержит указатель на такой же тип и имеет определенные данные

**Стек на листе:**

Методы:

List();- конструктор по умолчанию

~List();- деструктор

int GetSize();- функция возвращает количество элементов

T pop\_front();- возвращает верхний элемент и удаляет его

void clear();- удаление всех элементов

void push\_front(T data); - добавление элемента в начало

void print();- показывает все элементы

Поля:

int Size; -целочисленное число, которое будет отвечать за количество элементов

Node<T> \*head; -указатель на первый элемент

Класс:

class Node{};-содержит указатель на такой же тип и имеет определенные данные

**Польская запись:**

Функции:

int check(string test) – функция проверяет на правильность введенное в него выражение

string pol(string str) – функция переводит в польскую запись выражение

int result(string str) – функция считает с помощью стека выражение, которое записано в польской записи

# **Заключение**

В лабораторной работе были реализованы программы: стек, стек на списке, список, польская запись. Выполняющие свои задачи: хранение данных особым способом или же перевод выражения в польскую запись, чтобы посчитать особым способом. Программы реализованы с минимальными требованиями. И выполняют необходимые операции.

# **Список литературы**

1. Лабораторный практикум. Составители: Барышева И.В., Мееров И.Б., Сысоев А.В., Шестакова Н.В. Под редакцией Гергеля В.П. Учебно-методическое пособие. – Нижний Новгород: Нижегородский госуниверситет, 2017
2. С++. Руководство для начинающих – Герберт Шилдт.