Министерство образования и науки Российской Федерации

Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский**

**политехнический университет»**

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т**

**по лабораторной работе №19**

Дисциплина: «Основы алгоритмизации и программирования»

Тема: Стек через классы

Вариант 18

Выполнил:

Студент группы РИС-20-1б

Тараканов Д. М.

Проверила:

Доцент кафедры ИТАС

Полякова О. А.

Пермь

2021 год

# **Цель работы**

1) Получить практические навыки работы со стеками;

# **Постановка задачи**

Задача состоит в разработке программы, в которой создаётся стек через классы, и выполнить обработку стека в соответствии со своим вариантом:

(18 вариант)

1. Сформировать стек с информационным полем типа double.
2. Удалить из стека все элементы с четными индексами (2, 4, 6 и т.д.).

# **Анализ задачи**

1. Определить какие предстоит выполнить действия:

* Создать структуру Node в виде стека

struct Node {

double data;

Node\* prev;

};

* Создать основной стек .

Stack\* stack = new Stack;

* Создать стек в котором буду хранится элементы стека без элементов с четными индексами.

Stack\* temp\_stack = new Stack;

2. С какими типами данных действие надо сделать, в каком виде эти данные будут представлены:

* Для запоминания длины стека используется переменная size типа int.

int size=-1;

* Для инициализации стека используется переменная Stack.

Stack\* stack = new Stack;

* Для инициализации стека в котором будут хранится элементы стека без элементов с четными индексами.

Stack\* temp\_stack = new Stack;

3. Для решения задачи данные были представлены в виде множества элементов типа double.

4. Реализация операций ввода и вывода данных:

* Ввод данных с консоли реализован при помощи цикла for и оператора cin.

for (int i = 0; i <= N; i++) {

cout << "Введите " << i + 1 << " элемент: ";

cin >> el;

Stack.push(el);

}

* Вывод данных на консоль реализован при помощи цикла while и оператора cout.

Node\* temp = top;

while (temp != nullptr) {

cout << temp->data << " ";

temp = temp->prev;

}

cout << endl;

5. Поставленные задачи будут решены при помощи следующих действий:

* Программа запрашивает у пользователя размер стека.

while (size < 0) {

if (size < 0) {

cout << "Введите размер стека: ";

cin >> size;

}

}

* При помощи оператора for в стек добавляются элементы, введенные с консоли.

for (int i = 0; i < size; i++) {

cout << "Введите " << i + 1 << " элемент: ";

cin >> el;

Stack.push(el);

}

* При помощи оператора for программа передает элементы с четными индексами из Stack в StackP при помощи методов top, push и удаляет элементы из Stack при помощи метода pop. Программа копирует элементы из StackP в Stack.

int lenght = stack->Size();

if (lenght % 2 == 0) {

for (int i = 0; i < lenght; i++) {

if (i % 2 != 0) {

temp\_stack->Push(stack->Top());

}

stack->Pop();

}

}

else {

for (int i = 1; i <= lenght; i++) {

if (i % 2 != 0) {

temp\_stack->Push(stack->Top());

}

stack->Pop();

}

}

lenght = temp\_stack->Size();

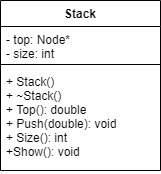
for (int i = 0; i < lenght; i++) {

stack->Push(temp\_stack->Top());

temp\_stack->Pop();

}

**UML диаграмма**



# **Код программы на языке C++**

**Stack.h**

#pragma once

#include <iostream>

using namespace std;

struct Node{

double data;

Node\* prev;

};

class Stack {

private:

Node\* top;

int size;

public:

Stack();

~Stack();

double Top();

void Push(double);

void Pop();

int Size();

void Show();

};

**Stack.cpp**

#include "Stack.h"

Stack::Stack() {

top = nullptr;

size = 0;

}

Stack::~Stack() {

Node\* temp = top;

while (temp != nullptr) {

Pop();

}

}

double Stack::Top() {

return top->data;

}

void Stack::Push(double value) {

Node\* temp = new Node;

temp->data = value;

temp->prev = top;

top = temp;

size++;

}

void Stack::Pop() {

Node\* temp = top;

top = top->prev;

delete temp;

size--;

}

int Stack::Size() {

return size;

}

void Stack::Show(){

cout << "\nСтек:\n";

Node\* temp = top;

while (temp != nullptr) {

cout << temp->data << " ";

temp = temp->prev;

}

cout << endl;

}

**Main**

#include <iostream>

#include "Stack.h"

void main() {

system("chcp 1251");

int size = -1;

cout << "\nВведите размер стека:\n> ";

while (size < 0) {

cin >> size;

if (size < 0) {

cout << "\nВведите положительное число:\n> ";

}

}

if (size == 0) {

cout << "\nСтек пуст!\n";

}

else {

Stack\* stack = new Stack;

double el;

for (int i = 0; i < size; i++) {

cout << "\nВведите " << i + 1 << " элемент:\n> ";

cin >> el;

stack->Push(el);

}

stack->Show();

int lenght = stack->Size();

Stack\* temp\_stack = new Stack;

if (lenght % 2 == 0) {

for (int i = 0; i < lenght; i++) {

if (i % 2 != 0) {

temp\_stack->Push(stack->Top());

}

stack->Pop();

}

}

else {

for (int i = 1; i <= lenght; i++) {

if (i % 2 != 0) {

temp\_stack->Push(stack->Top());

}

stack->Pop();

}

}

lenght = temp\_stack->Size();

for (int i = 0; i < lenght; i++) {

stack->Push(temp\_stack->Top());

temp\_stack->Pop();

}

stack->Show();

}

system("pause");

}

# **Скриншоты тестов**

