Министерство образования и науки Российской Федерации  
Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский  
политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЁТ**

**по лабораторной работе №18.19**

Дисциплина: основы теории алгоритмов и структуры данных

Тема: Основные алгоритмы работы со стеками

Выполнил работу

студент группы РИС-20-1б

Кузнецов Михаил Сергеевич

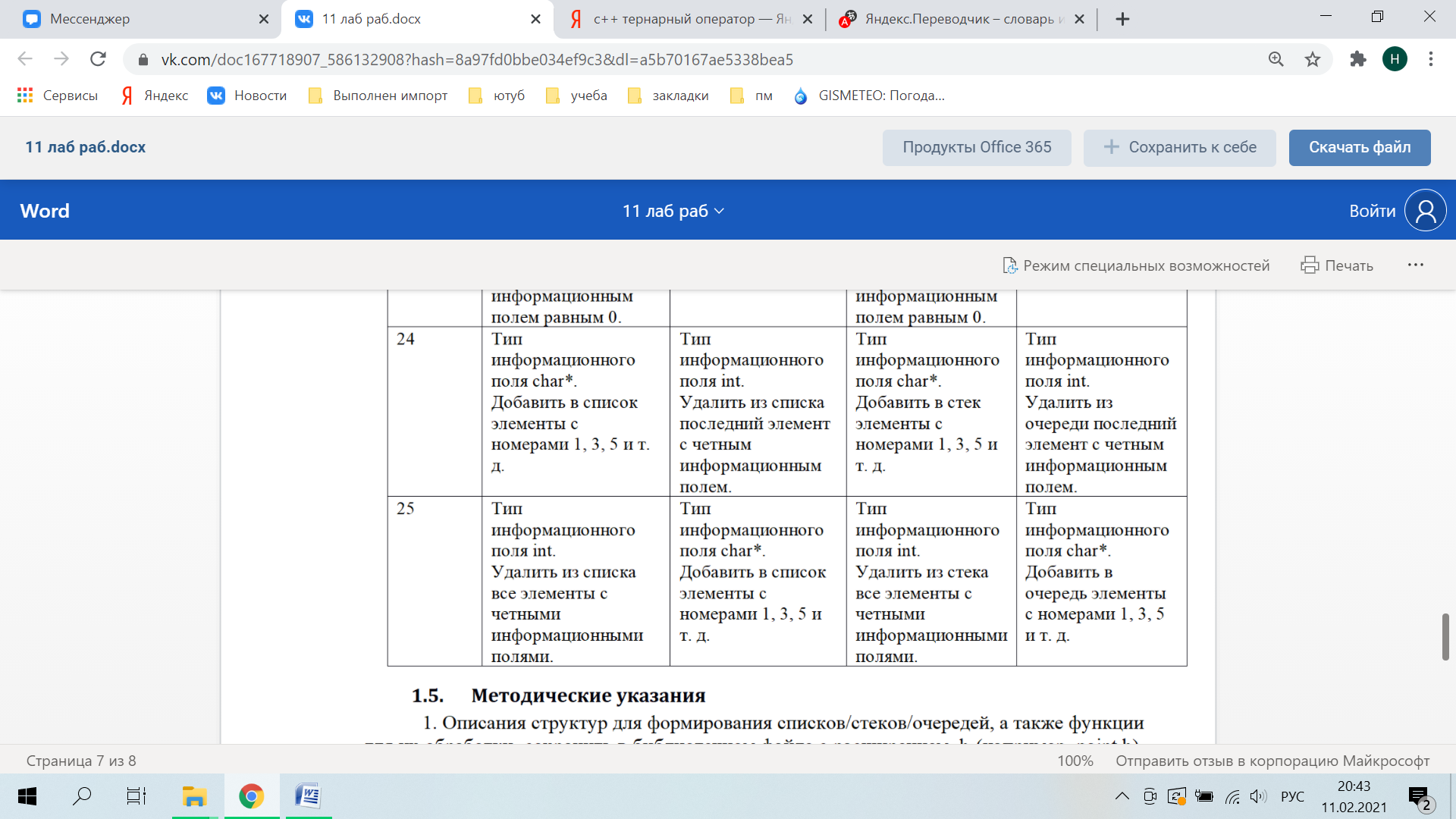
Проверила

Доцент кафедры ИТАС Полякова О.А.

Пермь, 2021

**Постановка задачи**

Реализовать стек, используя классы.



**Анализ задачи**

*Какие предстоит выполнить действия.*

Функция main считывает все необходимые данные и объявляет переменные

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");

}

*С каким типом данных действуем*

Целые числа

*В каком виде представлены данные*

Классы

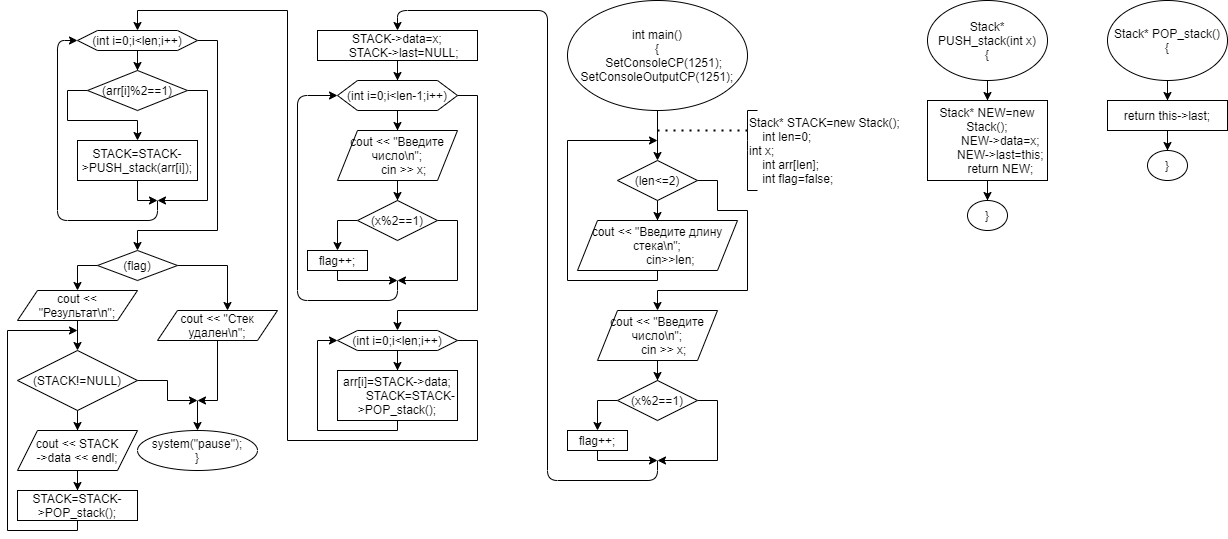
*Какие поля*

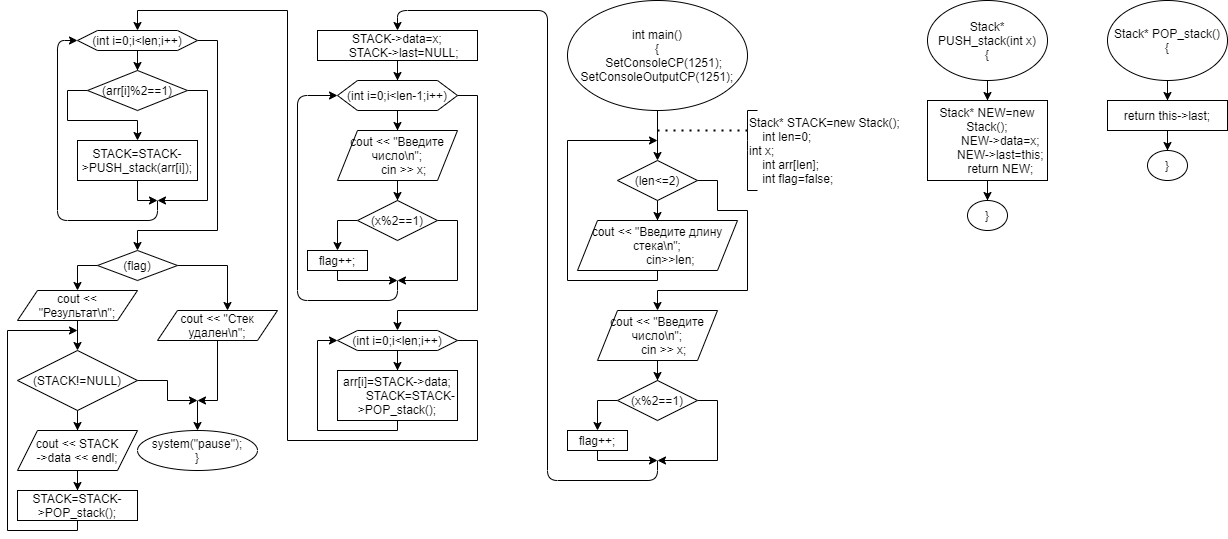
Поле int и поле указателя на такой же класс

*Операторы ввода/вывода*

В программе используются операторы cin/cout

**Блок-схема**

****

****

**Код**

#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");

}

#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 << " ";

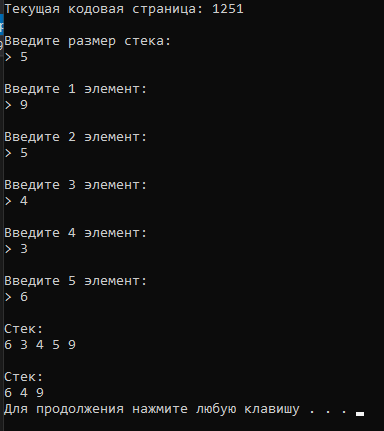
temp = temp->prev;

}

cout << endl;

}

**Скриншоты тестов**

****