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 №19**

Дисциплина: основы теории алгоритмов и структуры данных

Тема: Основные алгоритмы работы со стеками

Выполнил работу

студент группы РИС-20-1б

Еске Вячеслав Сергеевич

Проверила

Доцент кафедры ИТАС Полякова О.А.

Пермь, 2021

**Постановка задачи**

Реализовать стек, используя классы.

Тип информационного поля int. Удалить из стека последний элемент с четным информационным полем

**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

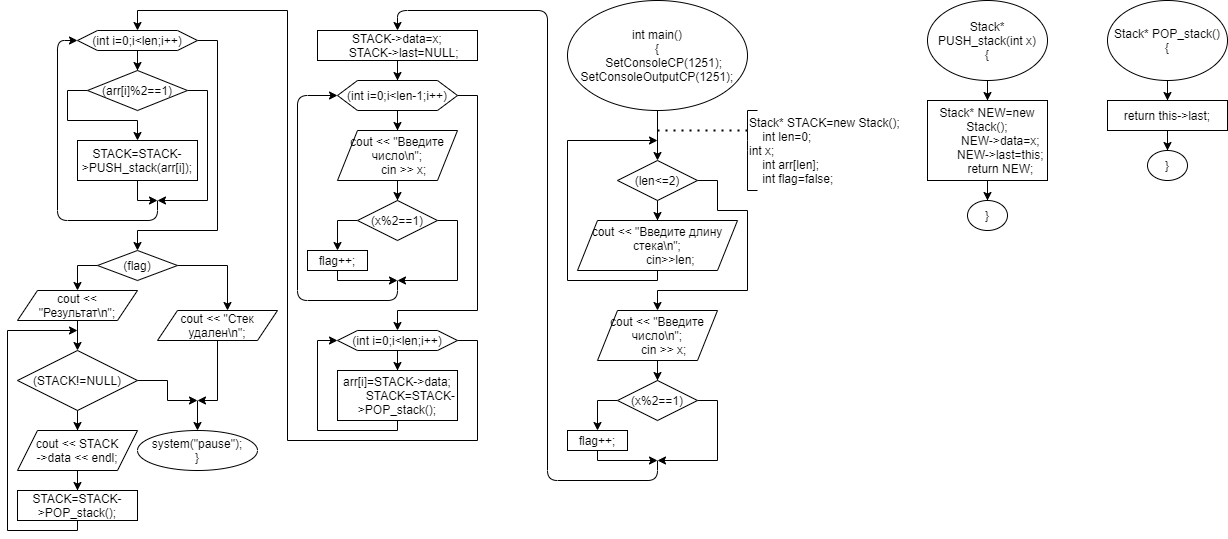
*Какие поля*

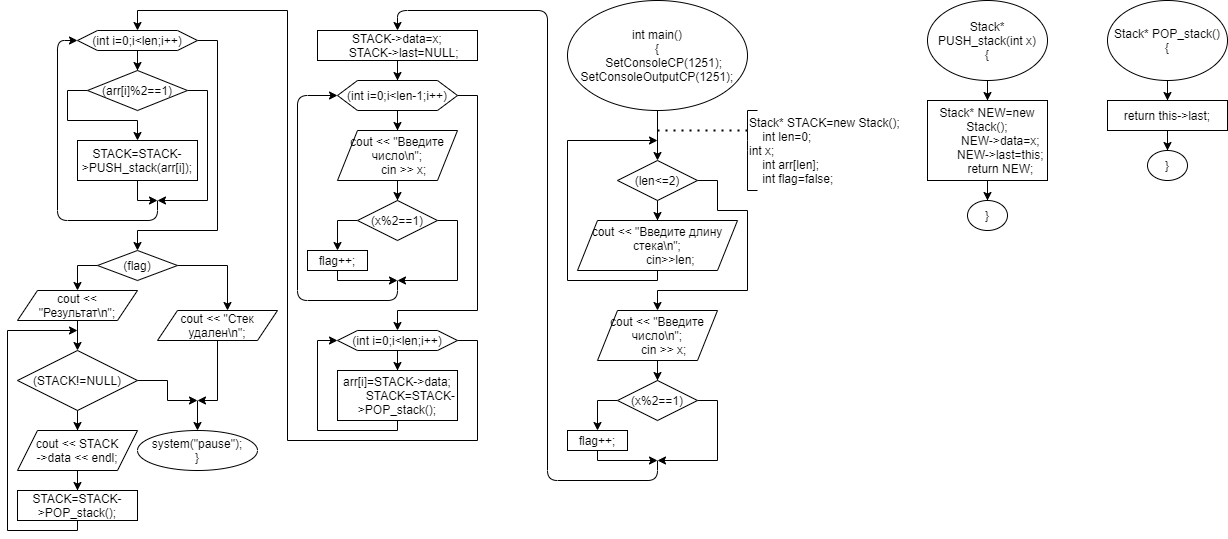
Поле int и поле указателя на такой же класс

*Операторы ввода/вывода*

В программе используются операторы cin/cout

**Блок-схема**

****

****

**Код**

#include <iostream>

using namespace std;

template <typename T>

class Node

{

private:

struct Stack

{

int data;

Stack\* prev;

};

Stack\* top;

int a;

bool f = true;

Stack\* x;

public:

Node() : top(nullptr) {}

Stack\* make\_stack(int n) {

if (n == 0) return NULL;

Stack\* p;

top = NULL;

p = new Stack;

cin >> a;

p->data = a;

if ((p->data % 2 == 0) && f) {

x = p;

f = false;

}

p->prev = NULL;

top = p;

for (int i = 2; i <= n; i++) {

Stack\* h = new Stack;

cin >> a;

h->data = a;

if ((h->data % 2 == 0) && f) {

x = h;

f = false;

}

h->prev = top;

top = h;

}

return top;

}

void print\_stack() {

if (top == NULL) {

cout << "Стек пуст" << endl;

}

else {

Stack\* p = top;

while (p != NULL) {

cout << p->data << " ";

p = p->prev;

}

cout << endl;

}

}

void delete\_stack() {

Stack\* p = top;

if (p == x) {

p = p->prev;

delete top;

top = p;

}

else {

while ((p != NULL)) {

if (p->prev == x) {

Stack\* q = p->prev;

p->prev = p->prev->prev;

delete q;

}

else p = p->prev;

}

}

}

};

int main()

{

setlocale(LC\_ALL, "rus");

Node<int> st;

int n;

do {

cout << "Введите количество элементов: ";

cin >> n;

} while (n <= 0);

cout << "Введите элементы: ";

st.make\_stack(n);

cout << "Стек: ";

st.print\_stack();

st.delete\_stack();

cout << "Стек: ";

st.print\_stack();

}

**Скриншоты тестов**

