Министерство высшего образования и науки Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования

**«Пермский национальный исследовательский политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЕТ**

**по выполненной лабораторной работе «CТЕК»**

Дисциплина: основы алгоритмизации и программирования

Выполнил:

Студент группы ИВТ

Голубцов Никита

Проверил:

Доцент кафедры ИТАС

Полякова Ольга Андреевна

**Пермь, 2023**

**Постановка задачи**

Реализовать стек.

Реализовать добавление элемента в стек.

Реализовать вхождение элемента, введенного пользователем, в стек.

**Анализ задачи**

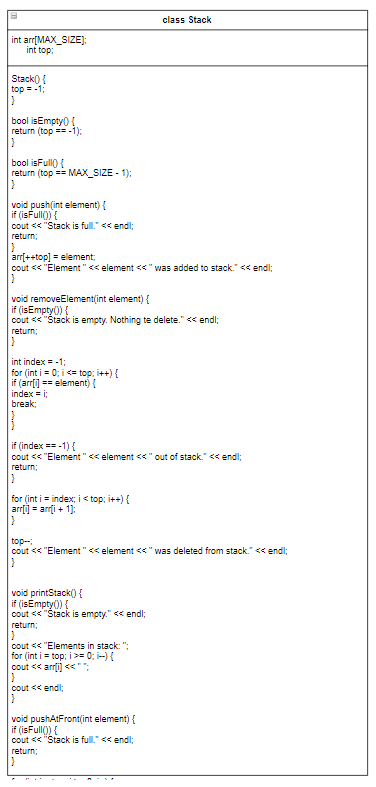
1. Класс Stack содержит методы проверки на наличие объектов стека и на его полноту вывода стека.

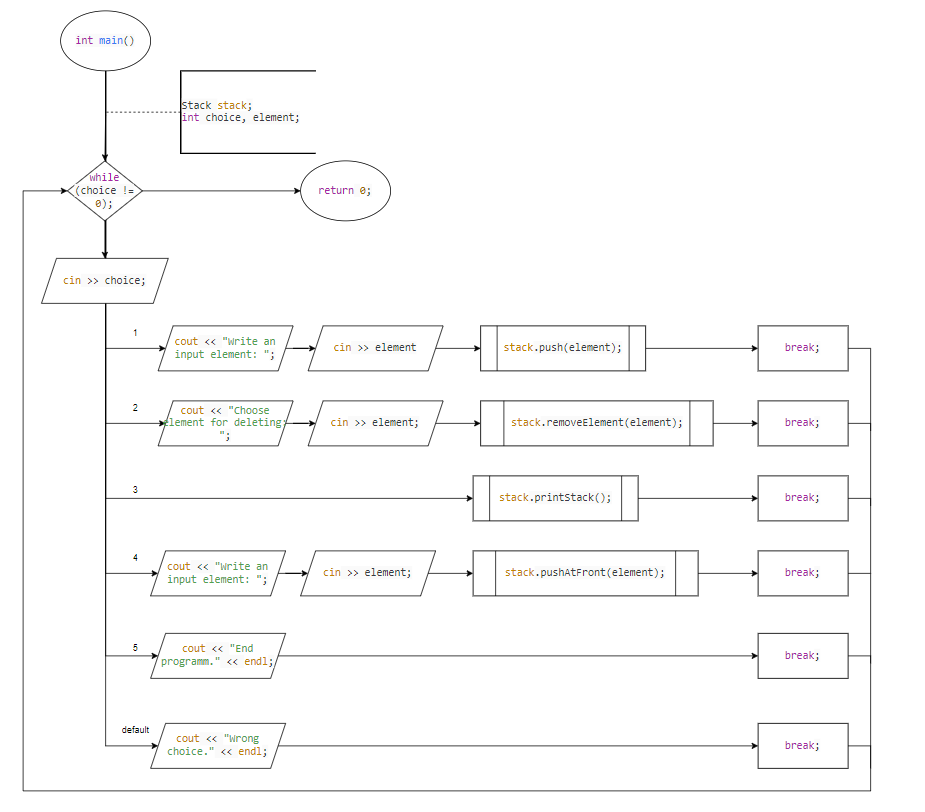
2. С помощью цикла while реализован выбор меню, необходимого для работы со стеком.

Если стек непустой и не его размер не больше, указанного изначально, с помощью метода push() добавляется новый элемент в конец стека. С помощью метода pushAtFront() добавляется элемент в начало стека. Если стек полон, добавление новых элементов невозможно.

3. Удаление введенного пользователем элемента происходит с помощью метода removeElement(). Элементы стека записываются в массив, происходит удаление выбранного элемента. Элементы стека выводятся на печать.

**Блок – схема**





**Код программы**

#include <iostream>

using namespace std;

const int MAX\_SIZE = 100;

class Stack {

private:

int arr[MAX\_SIZE];

int top;

public:

Stack() {

top = -1;

}

bool isEmpty() {

return (top == -1);

}

bool isFull() {

return (top == MAX\_SIZE - 1);

}

void push(int element) {

if (isFull()) {

cout << "Stack is full." << endl;

return;

}

arr[++top] = element;

cout << "Element " << element << " was added to stack." << endl;

}

void removeElement(int element) {

if (isEmpty()) {

cout << "Stack is empty. Nothing te delete." << endl;

return;

}

int index = -1;

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

if (arr[i] == element) {

index = i;

break;

}

}

if (index == -1) {

cout << "Element " << element << " out of stack." << endl;

return;

}

for (int i = index; i < top; i++) {

arr[i] = arr[i + 1];

}

top--;

cout << "Element " << element << " was deleted from stack." << endl;

}

void printStack() {

if (isEmpty()) {

cout << "Stack is empty." << endl;

return;

}

cout << "Elements in stack: ";

for (int i = top; i >= 0; i--) {

cout << arr[i] << " ";

}

cout << endl;

}

void pushAtFront(int element) {

if (isFull()) {

cout << "Stack is full." << endl;

return;

}

for (int i = top; i >= 0; i--) {

arr[i + 1] = arr[i];

}

arr[0] = element;

top++;

cout << "Element " << element << " was added to the begin of the stack." << endl;

}

};

int main() {

Stack stack;

int choice, element;

do {

cout << "Choose the operation" << endl;

cout << "1. Input new" << endl;

cout << "2. Delete" << endl;

cout << "3. Show" << endl;

cout << "4. Add Front Element" << endl;

cout << "0. Exit" << endl;

cout << "Your choice: ";

cin >> choice;

switch (choice) {

case 1:

cout << "Write an input element: ";

cin >> element;

stack.push(element);

break;

case 2:

cout << "Choose element for deleting: ";

cin >> element;

stack.removeElement(element);

break;

case 3:

stack.printStack();

break;

case 4:

cout << "Write an input element: ";

cin >> element;

stack.pushAtFront(element);

break;

case 0:

cout << "End programm." << endl;

break;

default:

cout << "Wrong choice." << endl;

break;

}

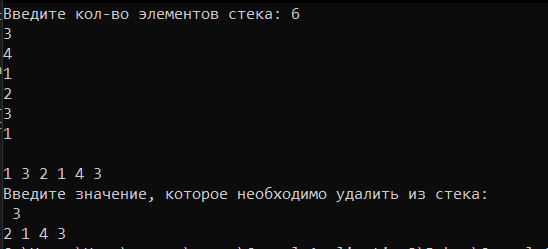
cout << endl;

} while (choice != 0);

return 0;

}

**Результаты работы программы**



**Анализ полученных результатов**

Добавление элементов реализовано корректно. Удаление элемента выполнено корректно.