Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«**Пермский национальный исследовательский политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЕТ**

Дисциплина: «Информатика»

Тема: Стек

Семестр 2

Выполнил работу

Студент группы РИС-22-1Б

Тарасов C.В.

Проверил

Доцент кафедры ИТАС

Полякова О.А.

Г. Пермь-2023

**Постановка задачи**

Написать программу, в которой реализуется работа со стеком.

А именно:

1. Добавление элемента

2. Удаление элемента

3. Показ стека (его верхушки, само собой)

**Анализ решения**

Стек (от англ. stack — стопка) — структура данных, представляющая из себя упорядоченный набор элементов, в которой добавление новых элементов и удаление существующих производится с одного конца, называемого вершиной стека. Притом первым из стека удаляется элемент, который был помещен туда последним, то есть в стеке реализуется стратегия «последним вошел — первым вышел» (last-in, first-out — LIFO).

Для реализации поставленной задачи нужно 2 структуры: Node и Stack.

struct Node {

int data;

Node\* next = nullptr;

};

struct Stack {

int size = 0;

Node\* head = nullptr;

};

Функция удаления/добавления элемента всегда будут удалять/добавлять последний элемент, как и функция показа стека, то есть его последнего элемента.

**Код программы**

#include <iostream>

using namespace std;

struct Node {

int data;

Node\* next = nullptr;

};

struct Stack {

int size = 0;

Node\* head = nullptr;

};

void push(Stack\*, int value);

void pop(Stack\*);

void last(Stack\*);

int main()

{

setlocale(LC\_ALL, "Russian");

Stack\* stack = new Stack;

int n, value;

do {

system("cls");

cout << "Введите 0 для завершения работы\n";

cout << "Введите 1 для добавления элемента \n";

cout << "Введите 2 для удаление элемента\n";

cout << "Введите 3 для показа верхушки стэка\n";

cout << "============================================================================\n";

cin >> n;

switch (n)

{

case 1:

{

cout << " Введите значение элемента "; cin >> value;

push(stack, value);

break;

}

case 2:

{

pop(stack);

break;

}

case 3:

{

last(stack);

cout << " Введите чтобы продолжить ";cin >> value;

break;

}

case 0:

cout << " Завершение работы ";

break;

}

} while (n != 0);

delete stack;

}

void push(Stack\* stack, int value)

{

if (stack->head == nullptr)

stack->head = new Node{ value };

else

{

Node\* new\_node = new Node;

new\_node->data = value;

new\_node->next = stack->head;

stack->head = new\_node;

}

stack->size++;

}

void pop(Stack\* stack)

{

if (stack->head != nullptr)

{

Node\* temp = stack->head;

stack->head = temp->next;

stack->size--;

delete temp;

}

}

void last(Stack\* stack)

{

if (stack->head == nullptr)

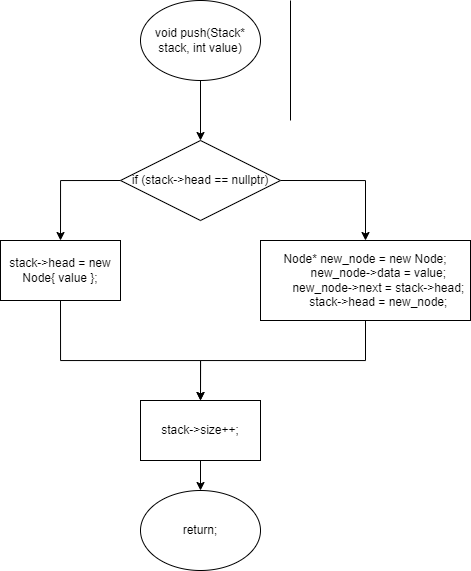
cout << "Стек пуст\n";

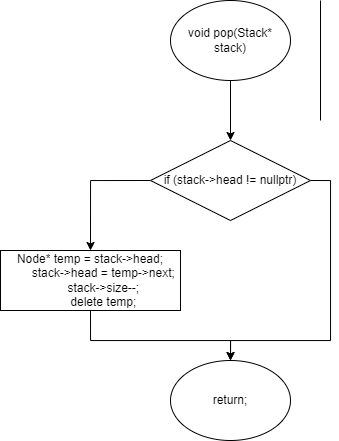
else

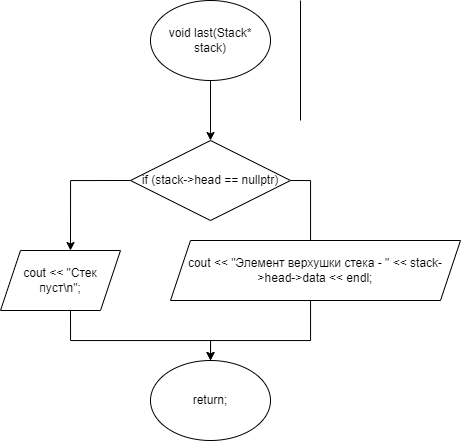
cout << "Элемент верхушки стека - " << stack->head->data << endl;

}

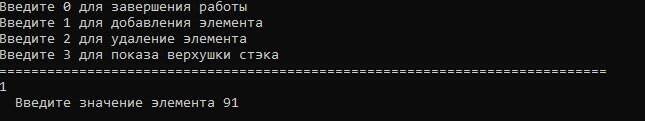
**Блок-схемы**

****

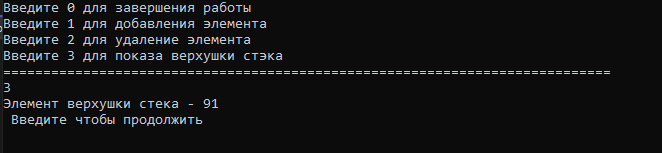
****

****

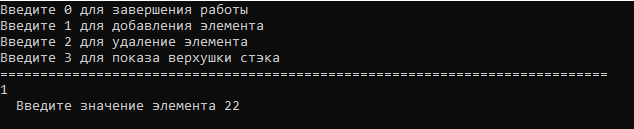
**Результат работы программы**



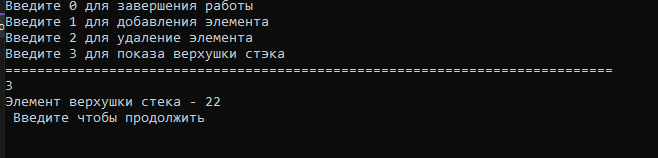
Добавление элемента



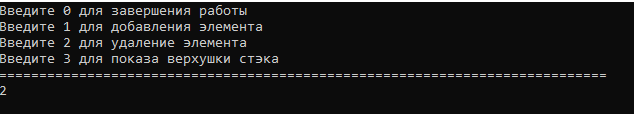
Показ верхушки стека



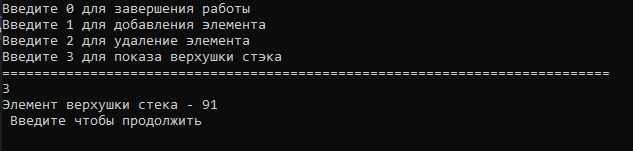
Добавление элемента



Показ верхушки стека



Удаление элемента



Показ стэка