Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Пермский национальный исследовательский**

**политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т**

**по лабораторной работе №19**

Дисциплина: основы алгоритмизации и программирования

Тема: Организация стека через классы

Вариант 7

Выполнил работу

студент группы РИС-20-1б

Нечаев Д.А.

Проверил

Доцент кафедры ИТАС

Полякова О.А.

Пермь, 2021

**Цель задачи**

Реализовать построение стека через класс

Написать консольное приложение для демонстрации работы программы

Для достижения поставленной цели необходимо решить следующие задачи:

* Изучить динамическую структуру данных стек
* Создать класс и методы класса для организации стека

Постановка задачи

1. Создать стек через классы
2. Создать метод увеличения стека в 2 раза

Анализ задачи

1. Определить какие операции должны быть выполнены по заданию:

* Создание класса

class stack

{

public:

char data;

stack\* prev;

};

Создание контейнера

stack\* sozd\_stack(int n)

Создание метода печати стека

void print(stack\* top)

Для решения задачи используются переменные:

Несколько целочисленных переменных для управления циклами, контроля размеров массивов и контроля длины строк

int len = 0;

int adress;

people x;

string user\_search;

int error = 0;

Указатели

people\* data\_base = new people[len];

string\* key = new string[len];

Структура для ввода и хранения названий городов, численности их населения

struct people

{

string fio;

string cardnum;

string summ;

};

1. Ввод данных осуществляется посредством функции

getline(F1, x.fio);

Вывод данных осуществляется посредством функции

cout << i << ": " << key[i] << endl;

1. Для решения задачи будут использованы циклы. Например, цикл для ввода информации в структуры

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

{

cout << "Город: ";

cin >> city[i].name;

cout << "Численость населения: ";

cin >> city[i].population;

}

1. Для защиты от случайных ошибок используются циклы на корректный ввод информации

while (size < 1)

{

cout << "Введите кол-во городов: ";

cin >> size;

cout << endl;

}

while (menu != 1 && menu != 2)

{

cout << "Выбери метод сортировки: \n 1. Хоара \n 2. Шелла \n Ввод: ";

cin >> menu;

}

Код

#include <iostream>

using namespace std;

char a;

class stack

{

public:

char data;

stack\* prev;

};

stack\* sozd\_stack(int n)

{

if (n == 0)

{

return NULL;

}

stack\* top, \* p;

top = NULL;

p = new stack;

cout << "Введите элементы: ";

cin >> a;

p->data = a;

p->prev = NULL;

top = p;

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

{

stack\* h = new stack;

cin >> a;

h->data = a;

h->prev = top;

top = h;

}

return top;

}

void print(stack\* top)

{

if (top == NULL)

{

cout << "Стек пуст" << endl;

}

else

{

stack\* p = top;

while (p != NULL)

{

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

p = p->prev;

}

cout << endl;

}

}

void add(stack\*& first, int n)

{

stack\* p = first;

stack\* h = new stack;

cout << "Введите элемент для добавления: ";

cin >> h->data;

h->prev = p;

first = h;

while (p->prev != NULL)

{

stack\* h = new stack;

cout << "Введите элемент для добавления: ";

cin >> h->data;

h->prev = p->prev;

p->prev = h;

if (p->prev->prev != NULL)

{

p = p->prev->prev;

}

else {

p = p->prev;

}

}

}

int main()

{

setlocale(LC\_ALL, "Rus");

cout << "Введите кол-во элементов: ";

int n;

cin >> n;

stack\* st = sozd\_stack(n);

cout << "Стек: ";

print(st);

add(st, n);

cout << "Изменённый стек: ";

print(st);

return 0;

}

Блок-схема:



Работа кода

