

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение высшего
образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА ПРИКЛАДНОЙ МАТЕМАТИКИ

10 *Б. А. Галанин*

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

доц., канд. техн. наук

должность, уч. степень, звание

Бег, 07.11.21

подпись, дата

В. А. Галанина

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №4

Создание контейнерного класса Vector.

по курсу: ИНФОРМАТИКА

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

M023

07.11.21 *Д.А. Трегуб*

подпись, дата

Д.А. Трегуб

инициалы, фамилия

Санкт-Петербург 2021

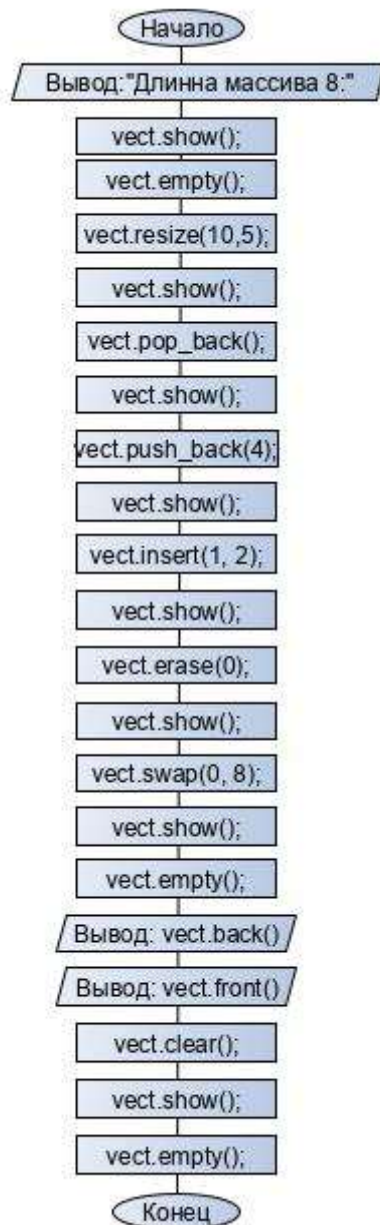
1)Цель работы:

Изучение принципов построения и использования контейнерных классов, совершенствование навыков объектно-ориентированного программирования на языке C++.

2)Задание на программирование:

Разработать класс, реализующий работу с динамическими массивами. Написать в классе методы, соответствующие аналогичным методам стандартного класса vector:

3)Блокхема:



4)Код:

```
#include "iostream"
#include <cassert>
using namespace std;

template < typename T >
```

```

class Vect {
private:
    T* data;
    int len = 0;
public:
    Vect(){
        len = 0;
        data = nullptr;
    }
    Vect(int q, int val = 0) {
        len = q;
        assert(q >= 0);

        if (len >= 0) {
            data = new T[len];
            for (int i = 0; i < len; i++)
                data[i] = val;
        }
        else
            data = nullptr;
    }
    ~Vect() {
        delete[] data;
        data = nullptr;
        len = 0;
    }
    int& operator[](int index)
    {
        return data[index];
    }
    void leng() {
        cout << len << endl;
    }
    void create(int q, int val = 0) {
        len = q;
        if (len >= 0) {
            data = new T[len];
            for (int i = 0; i < len; i++)
                data[i] = val;
        }
    }
    void clear() { //Полная отчистка массива
        delete[] data;
        data = nullptr;
        len = 0;
    }
    int front() { //Просмотр первой ячейки
        return (data[0]);
    }
    int back() { //Просмотр последней ячейки
        return (data[len-1]);
    }
    void resize(int n, int val = 0) { //изменяет размер массива на n элементов и
добавляет новые со значение val
        if (n == len) //Если совпадает
            return;
        if (n <= 0) { //Если нужно обнулить массив
            clear();
            return;
        }
        T* new_data = new T[n];
        if (n > len) { //Если новый размер массива больше
            for (int i = 0; i < len; i++)
                new_data[i] = data[i];
            for (int i = len; i < n; i++)

```

```

        new_data[i] = val; //изменение
        clear();
        len = n;
        data = new_data;
    }
    if (n < len) { //Если новый размер массива меньше
        for (int i = 0; i < n; i++)
            new_data[i] = data[i];
        clear();
        len = n;
        data = new_data;
    }
    new_data = nullptr;
}

void empty() { //если контейнер пуст возвращает true, если нет - false
    if (len == 0) {
        cout << "Вектор не содержит данных" << endl;
        return;
    }
    for (int i = 0; i < len; i++)
        if (data[i] != 0) {
            cout << "Вектор содержит данные" << endl;
            return;
        }
    cout << "Вектор не содержит данных" << endl;
    return;
}

void push_back(int val = 0) { //добавляет заданный элемент в конец вектора
    len = len + 1;
    T* new_data = new T[len];
    for (int i = 0; i < len - 1; i++)
        new_data[i] = data[i];
    new_data[len-1] = val;
    clear();
    len = n;
    data = new_data;
    new_data = nullptr;
};

void pop_back() { //удаляет элемент из конца вектора
    len = len - 1;
    T* new_data = new T[len];
    for (int i = 0; i < len; i++)
        new_data[i] = data[i];
    clear();
    len = n;
    data = new_data;
    new_data = nullptr;
}

void insert(int val = 0, int n = 1, int it = 0) { //добавляет элементы в начало
вектора
    len = len + n;
    T* new_data = new T[len+1];
    for (int i = it; i < n; i++)
        new_data[i] = val;
    for (int i = n; i < len; i++)
        new_data[i] = data[i-n];
    clear();
    len = n;
    data = new_data;
    new_data = nullptr;
}

void erase(int n) { //удаляет выбранный элемент
    len = len - 1;
    T* new_data = new T[len+1];
    for (int i = 0; i < n; i++)

```

```

    new_data[i] = data[i];
        for (int i = n+1; i < len+1; i++)
            new_data[i - 1] = data[i];
    clear();
    len = n;
    data = new_data;
    new_data = nullptr;
}

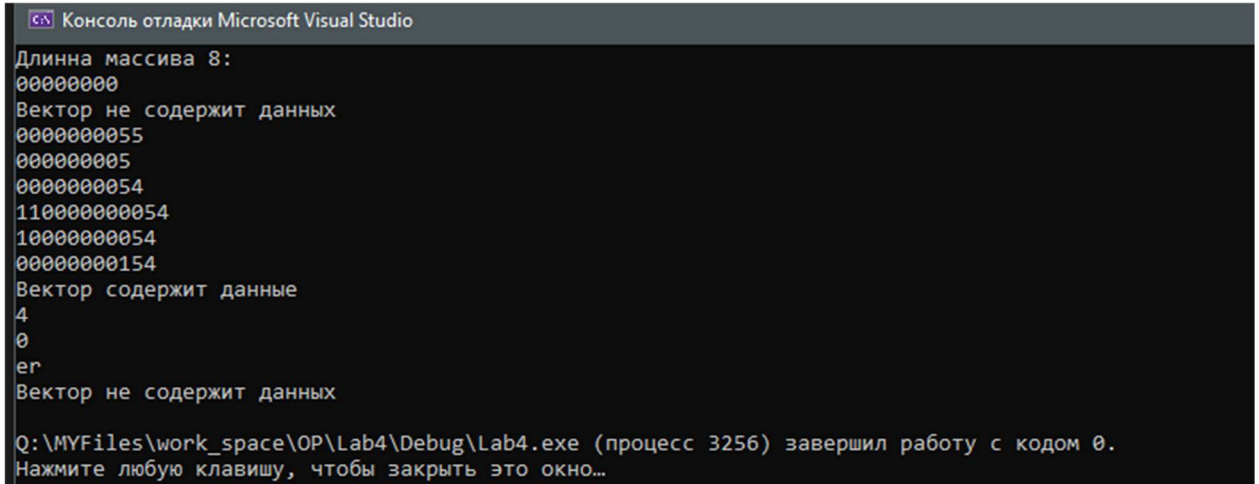
void swap(int n, int m) { //меняет два элемента местами
    T q = data[n];
    data[n] = data[m];
    data[m] = q;
}

void show() {
    if (len == 0)
        cout << "er";
    for (int i = 0; i < len; i++)
        cout << data[i];
    cout << endl;
}
};

int main() {
    setlocale(LC_ALL, "Russian");
    cout << "Длина массива 8:" << endl;
    Vect<double> vect(8);
    vect.show();
    vect.empty();
    vect.resize(10,5);
    vect.show();
    vect.pop_back();
    vect.show();
    vect.push_back(4);
    vect.show();
    vect.insert(1, 2);
    vect.show();
    vect.erase(0);
    vect.show();
    vect.swap(0, 8);
    vect.show();
    vect.empty();
    cout << vect.back() << endl;
    cout << vect.front() << endl;
    vect.clear();
    vect.show();
    vect.empty();
    return 0;
}

```

5)Пример работы программы:

A screenshot of the Microsoft Visual Studio debug console. The title bar reads 'Консоль отладки Microsoft Visual Studio'. The output text is as follows:

```
Длина массива 8:  
00000000  
Вектор не содержит данных  
0000000055  
000000005  
0000000054  
11000000054  
1000000054  
00000000154  
Вектор содержит данные  
4  
0  
er  
Вектор не содержит данных  
  
Q:\MYFiles\work_space\OP\Lab4\Debug\Lab4.exe (процесс 3256) завершил работу с кодом 0.  
Нажмите любую клавишу, чтобы закрыть это окно...
```

6)Вывод: В процессе выполнения лабораторной работы я изучил принцип построения и использования контейнерных классов, улучшил навыки в объектно-ориентированного программирование на языке C++.