Министерство цифрового развития, связи и массовых коммуникаций РФ

Федеральное государственное бюджетное общеобразовательное учреждение

высшего образования

“Сибирский Государственный Университет Телекоммуникаций и Информатики”

(СибГУТИ)

Кафедра телекоммуникационных систем и вычислительных средств

Отчет по лабораторной работе на тему: Контейнерные типы данных в языке С++. Векторы

Выполнил: студент 1 курса группы ИА-231

Зырянов Иван Александрович

Проверил: Вейлер Андрей Игоревич

Новосибирск, 2023

**ЗАДАНИЕ**

**Задание.** Составить алгоритм решения задачи согласно своему варианту и написать программу на языке С++, используя **Класс vector.**

**Создать вектор из объектов класса, разработанного вами в лабораторной работе №4 по дисциплине «Технологии программирования».**

**Создать новый вектор, переписав в него некоторые объекты исходного вектора (критерий для перезаписи выбрать самостоятельно). Проверить**, не оказался ли новый вектор пустым.

**Если не пуст, провести сортировку** объектов в новом векторе. (критерий сортировки выбрать самостоятельно, исходя из атрибутов объектов вашего класса). **Размерности векторов задаются с клавиатуры.**

**Примечание**. **В программе МАКСИМАЛЬНО использовать стандартную библиотеку.**

В отчете предоставить перечень использованных методов класса и функций стандартной библиотеки и текст программы со скриншотами результатов ее работы.

перечень использованных методов класса и функций стандартной библиотек:

Использованные методы класса и функции стандартной библиотеки:

* <iostream>: для стандартного ввода и вывода данных через консоль.
* <vector>: для работы с динамическим массивом объектов класса MedicalInstitution.
* <algorithm>: для использования функции сортировки вектора и поиска максимального элемента вектора.

Методы класса MedicalInstitution:

* outputTo(): для вывода информации об объекте на консоль.

Функции стандартной библиотеки:

* sort(): для сортировки вектора по заданному критерию (sortByAddress в данном случае).
* push\_back(): для добавления элемента в конец вектора.
* empty(): для проверки, пуст ли вектор.
* begin(): для получения итератора на начало вектора.
* end(): для получения итератора на конец вектора.

**Код**

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

class MedicalInstitution {

public:

    // поля

    string type;

    string address;

    // конструктор

    MedicalInstitution(string t, string a) : type(t), address(a) {}

    // Способ вывода информации на консоль

    void outputTo() const {

        cout << "Type: " << type << endl;

        cout << "Address: " << address << endl;

    }

};

bool sortByAddress(const MedicalInstitution& a, const MedicalInstitution& b) {

    return a.address < b.address;

}

int main() {

    unsigned int size; // измененный тип переменной i

    cout << "Enter vector size: ";

    cin >> size;

    // Создать вектор объектов медицинского учреждения

    vector<MedicalInstitution> medicalInstitutions;

    for (unsigned int i = 0; i < size; i++) { // измененный тип переменной i

        string type, address;

        cout << "Enter type and address for Medical Institution " << i + 1 << ": ";

        cin >> type >> address;

        medicalInstitutions.emplace\_back(type, address);

    }

    // Создайте новый вектор и скопируйте выбранные элементы

    vector<MedicalInstitution> newVector;

    for (unsigned int i = 0; i < medicalInstitutions.size(); i++) { // измененный тип переменной i

        if (medicalInstitutions[i].type == "clinic") {

            newVector.push\_back(medicalInstitutions[i]);

        }

    }

    // проверка пуст ли новый вектор

    if (newVector.empty()) {

        cout << "New vector is empty" << endl;

        return 0;

    }

    // Отсортировать новый вектор по адресу

    sort(newVector.begin(), newVector.end(), sortByAddress); // добавлен вызов sort

    // Вывести новый вектор

    cout << "New vector size: " << newVector.size() << endl;

    for (unsigned int i = 0; i < newVector.size(); i++) { // измененный тип переменной i

        newVector[i].outputTo();

    }

    // Найти максимальный адрес в новом векторе

    MedicalInstitution maxAddress = newVector[0];

    for (unsigned int i = 1; i < newVector.size(); i++) { // измененный тип переменной i

        if (newVector[i].address > maxAddress.address) {

            maxAddress = newVector[i];

        }

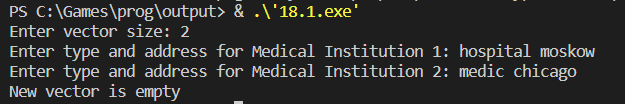
    }

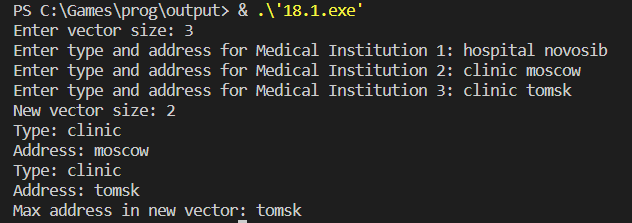
    cout << "Max address in new vector: " << maxAddress.address << endl;

    return 0;

}

**Выводы программы:**



****