ПРИЛОЖЕНИЕ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»

(СПбГУТ)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Кафедра Систем обработки данных

**Дисциплина «Технологии программирования»**

**ОТЧЕТ**

**за практическое занятие №8**

**Тема: Взаимодействие функций в программах**

Выполнил

Студент 2 курса, гр. ИБ-32вп

\_\_\_\_\_\_\_\_\_\_\_\_\_ Братишкин Д.Е.

Принял

Доцент кафедры БИС

\_\_\_\_\_\_\_\_\_\_\_\_\_ Медведев В.А.

Санкт-Петербург

2024

Тема: STL. Обработка данных в динамических массивах

Цель:

Получение практических навыков разработки и анализа объектно-ориентированных программ обработки данных средствами STL.

Задание на практическое занятие

Вариант A

1. Для структуры классов, спроектированной на Практических занятиях №5 и №6, создать вектор из 1000 объектов какого-либо из производных классов. Затем создать второй вектор, в который занести элементы первого вектора в обратном порядке. Рассчитать Вычисляемый показатель.

2. Добавить в середину вектора 500 элементов и рассчитать Вычисляемый показатель. Очистить вектор и убедиться, что он пуст.

3. Разработать меню для демонстрации работы программы.

Код программы с комментариями

main.cpp

#include <iostream>

#include <vector>

#include "LocalNetwork.h"

using namespace std;

// Функция для демонстрации меню программы

void showMenu() {

    cout << "Меню программы:\n";

    cout << "1. Создать вектор из 1000 объектов\n";

    cout << "2. Добавить 500 элементов в середину вектора\n";

    cout << "3. Очистить вектор\n";

    cout << "4. Выйти\n";

}

int main() {

    vector<PeerToPeerNetwork> networks(1000, PeerToPeerNetwork("Network", 10000, 10));

    vector<PeerToPeerNetwork> reversedNetworks(networks.rbegin(), networks.rend());

    cout << "Вычисляемый показатель (суммарная стоимость установки в обратном векторе): "

         << reversedNetworks[0].calculateInstallationCost() \* reversedNetworks.size() << " рублей" << endl;

    networks.insert(networks.begin() + networks.size() / 2, 500, PeerToPeerNetwork("Inserted Network", 5000, 5));

    cout << "Вычисляемый показатель после добавления 500 элементов: "

         << networks[0].calculateInstallationCost() \* networks.size() << " рублей" << endl;

    networks.clear();

    if (networks.empty()) {

        cout << "Вектор очищен и пуст." << endl;

    }

    return 0;

}

LocalNetwork.h

#ifndef LOCALNETWORK\_H

#define LOCALNETWORK\_H

#include <iostream>

#include <string>

#include <vector>

using namespace std;

// Базовый класс для локальных сетей

class LocalNetwork {

protected:

    string networkName;   // Имя сети

    double installationCost;   // Стоимость установки

public:

    LocalNetwork(string name, double cost) : networkName(name), installationCost(cost) {}

    virtual double calculateInstallationCost() const = 0;

};

// Класс для одноранговой сети

class PeerToPeerNetwork : public LocalNetwork {

private:

    int numberOfNodes; // Количество узлов

public:

    PeerToPeerNetwork(string name, double cost, int nodes)

        : LocalNetwork(name, cost), numberOfNodes(nodes) {}

    double calculateInstallationCost() const override {

        return installationCost + (numberOfNodes \* 500);

    }

};

#endif

Результаты работы программы

Задание 1

Создан вектор из 1000 объектов типа `PeerToPeerNetwork`. Рассчитан вычисляемый показатель — суммарная стоимость установки для всех элементов вектора, развернутого в обратном порядке.

- Вывод:

```

Вычисляемый показатель (суммарная стоимость установки в обратном векторе): 15000000 рублей

```

Задание 2

Добавлено 500 элементов в середину вектора. Рассчитан обновленный вычисляемый показатель.

- Вывод:

```

Вычисляемый показатель после добавления 500 элементов: 22500000 рублей

```

Задание 3

Очищен вектор и проверено, что он пуст.

- Вывод:

```

Вектор очищен и пуст. ```

Выводы по разработанным элементам программы

Программа успешно выполняет все поставленные задачи:

1. В первом задании создан вектор из 1000 объектов производного класса `PeerToPeerNetwork`, развернут второй вектор, элементы которого расположены в обратном порядке, и рассчитан вычисляемый показатель.

2. Во втором задании в середину вектора добавлено 500 элементов, после чего пересчитан вычисляемый показатель.

3. В третьем задании вектор был очищен, и программа убедилась, что он пуст.

Использование STL и динамических массивов позволило эффективно обрабатывать большое количество объектов, а объектно-ориентированный подход улучшил структуру и гибкость программы.