Министерство науки и высшего образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №3

по дисциплине: «Объектно - ориентированное программирование»

на тему: «Изучить механизмы наследования в языке C++»

Выполнили:

студенты группы 22ВВВ2

Ипполитов И. Д.

Хоссейни Нежад С. А. С. М.

Приняли:

Евсеева Ю.И.

Гудков А.А.

Пенза, 2024

**Цель работы:** Изучить механизмы наследования в языке C++.

1. Получить индивидуальное задание у преподавателя.
2. Написать программу, выполняющую действия, согласно заданию.
3. Данные в программе должны вводиться с клавиатуры
4. Программу требуется отладить и протестировать на различных наборах данных
5. Продемонстрировать работу программы на экране.
6. Сдать теоретическую часть по отчету преподавателю.

**Задания**

Создать иерархию классов **Организация–Страховая компания–Завод**. Классы должны содержать поля для описания свойств объектов и методы для работы с ними. Основная программа должна создавать массивы объектов производных классов и выводить их на экран.

**Листинг**

#include <iostream>

#include <windows.h>

#include <string>

#include <vector>

using namespace std;

class Organization

{

private:

string organization;

string date;

public:

string getOrganization() { return organization; }

void setOrganization(string organization) { this->organization = organization; }

string getDate() { return date; }

void setDate(string date) { this->date = date; }

void showOrganization()

{

cout << "Организация: " << getOrganization() << endl;

}

void showDate()

{

cout << "Дата создания: " << getDate() << endl;

}

};

class InsuranceCompany : public Organization

{

private:

string insuranceCompany;

public:

string getInsuranceCompany() { return insuranceCompany; }

void setInsuranceCompany(string insuranceCompany) { this->insuranceCompany = insuranceCompany; }

void showInsuranceCompany()

{

cout << "Страховая компания: " << getInsuranceCompany() << endl;

}

void showDate()

{

cout << "Дата создания: " << getDate() << endl;

}

};

class Factory : public InsuranceCompany

{

private:

string nameFactory;

int performance;

public:

Factory(string organization, string insuranceCompany, string nameFactory, int performance, string date)

{

setOrganization(organization);

setInsuranceCompany(insuranceCompany);

setDate(date);

this->nameFactory = nameFactory;

this->performance = performance;

}

string getNameFactory() { return nameFactory; }

int getPerformance() { return performance; }

void setNameFactory(string nameFactory) { this->nameFactory = nameFactory; }

void setPerformance(int performance) { this->performance = performance; }

void show()

{

showOrganization();

showInsuranceCompany();

showDate();

cout << "Завод: " << getNameFactory() << endl;

cout << "Производительность завода: " << getPerformance() << endl << endl;

}

};

int main()

{

setlocale(LC\_ALL, "rus");

SetConsoleOutputCP(1251);

SetConsoleCP(1251);

string organization;

string insuranceCompany;

string date;

string nameFactory;

int performance;

vector<Factory> v1;

for (size\_t i = 0; i < 2; i++)

{

cout << "Введите организацию: "; cin >> organization;

cout << "Страховую компанию: "; cin >> insuranceCompany;

cout << "Введите дату создания: "; cin >> date;

cout << "Название завода: "; cin >> nameFactory;

cout << "Производительность завода: "; cin >> performance;

cout << endl;

Factory factory{ organization, insuranceCompany, nameFactory, performance, date};

v1.push\_back(factory);

}

for (size\_t i = 0; i < v1.size(); i++)

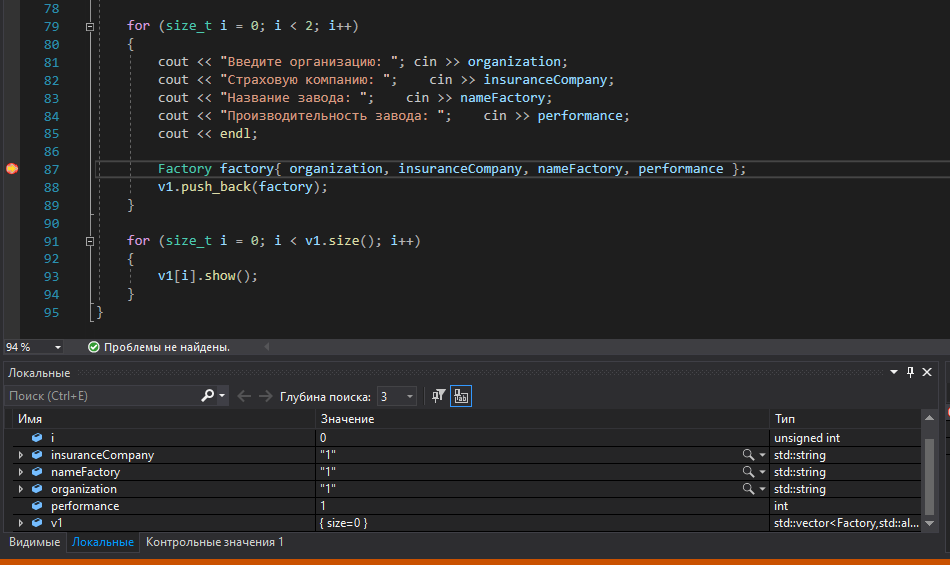
{

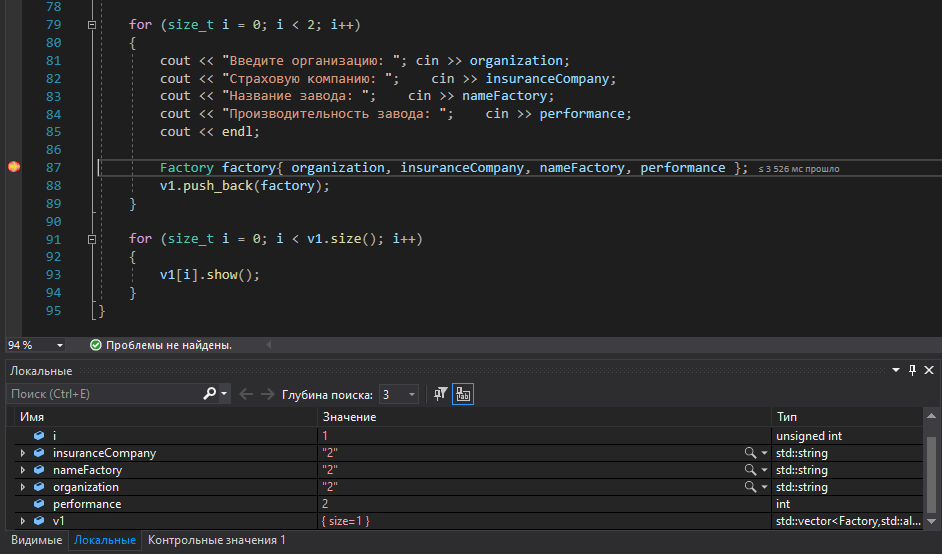
v1[i].show();

}

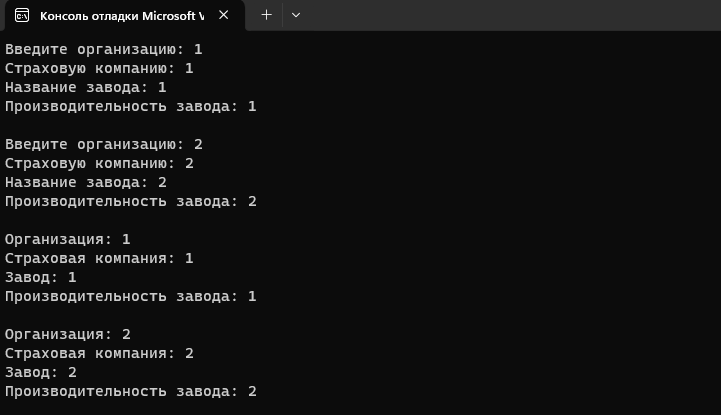
}

**Отладка программы**





**Результат работы программы**

****

**Вывод:** изучили механизмы перегрузки операций и дружественных функций в языке C++.