МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

#### Факультет информационных технологий и компьютерной безопасности

(факультет)

### Кафедра Систем автоматизированного проектирования и информационных систем

# ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 6

по дисциплине Объектно-ориентированное программирование

Тема: «Наследование»

Выполнили студенты группы ИСТ-222 Е.Г. Гладнева

Подпись, дата Инициалы, фамилия

Е.В. Журавлёва

Подпись, дата Инициалы, фамилия

## Принял Б.Н. Тишуков

Подпись, дата Инициалы, фамилия

Защищена \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

дата

2023

Лабораторная работа № 6

«Наследование»

1 Цель работы

Целью лабораторной работы является изучение и практическое освоение организации взаимодействия между классами в языке программирования C++.

1. Задание на лабораторную работу

В задании для варианта 6 необходимо создать иерархию классов, используя наследование или композицию. В каждой программе необходимо соблюсти принцип разделения интерфейса и реализации класса. Необходимо написать программу, иллюстрирующую применение всех методов классов. Прежде чем приступить к написанию программ, нужно продумайте, какие необходимы функции в каждом из классов (может, где-то необходимо считать координаты, где-то площади и объемы, а где-то хранить фамилии и года поступления): как в базовом, так и в классах-наследниках. Также нужно продумайте, что следует поместить в закрытые (а, возможно, защищенные) переменные. Предусмотрите возможность переопределения методов базового класса в производном. Приветствуется демонстрация наследования перегруженных операций.

Факультет:

a) администрация;

b) преподаватель;

c) студент.

3 Ход выполнения

1. Создаём классы Administration, Teacher, Student, в которые и будет заключена информация о администрации, преподавателе и студенте.
2. Для класса Administration создаём переменную типа private timework, что будет означать время работы администрации, далее создаём метод в классе Administration(int time), в котором будет передаваться то время, которое назначит пользователь.
3. Для класса Teacher создаём переменные типа private: name, surname, subject. Далее создаём метод Teacher(string name1, string name2, string object), в котором будет передаваться имя, фамилия и предмет, который ведёт преподаватель, эти параметры назначает пользователь.
4. Для класса Student создаём переменные типа public: name, surname, average\_score. Далее создаём метод Student(string name1, string name2), в котором будет передаваться имя, фамилия и средний балл студента эти параметры назначает пользователь.
5. Далее создаем класс Faculty, в котором будет заключена об администрации, преподавателе и студенте. Создаём переменные типа private Student student, Administration administration, Teacher teacher.
6. Создаём переменную title – название факультета, метод Faculty(string name, string student\_name, string student\_surname, Administration admin, Teacher person): student(student\_name, student\_surname), в котором мы передаём классу Faculty значения Student – информацию о студенте, Administration – информация об администрации, Teacher – информацию о преподавателе.
7. В каждом из классов создаем метод show\_data, который будет выводит информацию об администрации, преподавателе, студенте и факультете, под которым они прикреплены.
8. В функции main запрашиваем у пользователя всю необходимую информацию (рисунок 1).

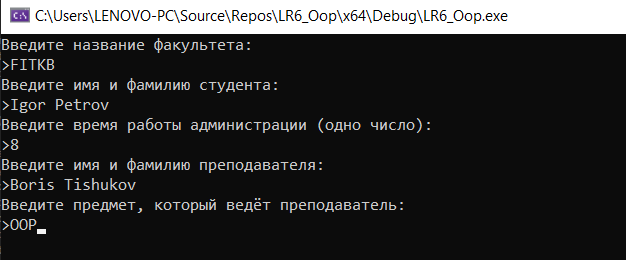


Рисунок 1 – Контекстное меню

Результат проделанной работы представлен на рисунке 2.

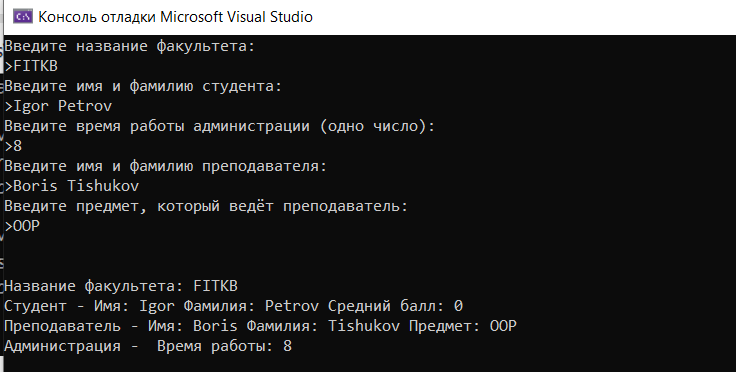


Рисунок 2 – Результат проделанной работы

Вывод

В ходе выполнения лабораторной работы мы освоили организацию взаимодействия между классами в языке программирования C++.

Приложение А

Листинг программного кода языка C++

#include <iostream>

using namespace std;

class Administration

{

private:

int timework;

public:

Administration()

{

timework = 0;

}

Administration(int time)

{

timework = time;

}

void show\_data()

{

cout << " Время работы: " << timework << endl;

}

};

class Teacher

{

private:

string name;

string surname;

string obj;

public:

Teacher()

{

name = ""; surname = ""; obj = "";

}

Teacher(string name1, string name2, string object)

{

name = name1; surname = name2; obj = object;

}

void show\_data()

{

cout << "Имя: " << name << " Фамилия: " << surname << " Предмет: " << obj << endl;

}

};

class Student

{

public:

string name;

string surname;

float average\_score;

Student()

{

name = ""; surname = ""; average\_score = 0;

}

Student(string name1, string name2)

{

name = name1; surname = name2; average\_score = 0;

}

void show\_data()

{

cout << "Имя: " << name << " Фамилия: " << surname << " Средний балл: " << average\_score << endl;

}

};

class Faculty

{

private:

Student student;

Administration administration;

Teacher teacher;

public:

string title;

Faculty(string name, string student\_name, string student\_surname, Administration admin, Teacher person): student(student\_name, student\_surname)

{

title = name;

administration = admin;

teacher = person;

}

void show\_data()

{

cout << "Название факультета: " << title << endl

<< "Студент - ";

student.show\_data();

cout << "Преподаватель - ";

teacher.show\_data();

cout << "Администрация - ";

administration.show\_data();

}

};

int main()

{

setlocale(LC\_ALL, "RUS");

string title;

string student\_name, student\_surname;

string teacher\_name, teacher\_surname, obj;

int time;

cout << "Введите название факультета:\n>";

cin >> title;

cout << "Введите имя и фамилию студента:\n>";

cin >> student\_name >> student\_surname;

cout << "Введите время работы администрации (одно число):\n>";

cin >> time;

cout << "Введите имя и фамилию преподавателя:\n>";

cin >> teacher\_name >> teacher\_surname;

cout << "Введите предмет, который ведёт преподаватель:\n>";

cin >> obj;

cout << endl << endl;

Administration admin(time);

Teacher person(teacher\_name, teacher\_surname, obj);

Faculty faculty(title, student\_name, student\_surname, admin, person);

faculty.show\_data();

return 0;

}