МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра ИИТ

Лабораторная работа №5

За 1 семестр

По дисциплине Проектирование программ в интеллектуальных системах

Тема: «Иерархии классов. Наследование»

Выполнил:

Студент 2 курса

Группы ИИ-19

Летченя С. А.

Проверил:

Монтик Н.С

Брест 2021

Цели:

1. Изучение правил наследования классов.

2. Реализация одиночного наследования классов.

3. Изучение управления методами и свойствами производных классов через объекты производных классов и через указатели на объекты производных классов.

4. Изучение правил описания наследования и диаграмм классов в языке UML.

**Ход работы:**

#include <iostream>

#include <cstring>

using namespace std;

class Human {

protected:

string name;

string lastname;

int age;

public:

Human() {

name = "Sergei";

lastname = "Letchenia";

age = 19;

}

Human(string n, string s, int a) {

name = n;

lastname = s;

age = a;

}

void To\_Set(string n, string s, int a) {

name = n;

lastname = s;

age = a;

}

void TO\_Show() {

cout << "Имя -- " << name << endl;

cout << "Фамилия -- " << lastname << endl;

cout << "Возраст -- " << age << endl;

}

};

class Studen : public Human {

private:

int grup\_n;

int curs;

string faculty;

public:

Studen() {

grup\_n = 19;

curs = 2;

faculty = "Fic";

}

Studen(int g, int c, string f, string n, string s, int a) :Human(n, s, a) {

grup\_n = c;

curs = g;

faculty = f;

}

void To\_Set(int g, int c, string f, string n, string s, int a) {

Human::To\_Set(n, s, a);

grup\_n = c;

curs = g;

faculty = f;

}

void TO\_Show() {

Human::TO\_Show();

cout << "Номер группы -- " << grup\_n << endl;

cout << "Курс -- " << curs << endl;

cout << "Факультет -- " << faculty << endl;

}

int To\_Get\_age() {

return age;

}

string To\_Get\_name() {

return name;

}

};

int main() {

system("color 70");

setlocale(LC\_ALL, "rus");

Studen\* obj\_1 = new Studen;

obj\_1->TO\_Show();

cout << endl;

cout << "Имя -- " << obj\_1->To\_Get\_name() << endl;

cout << "Возраст -- " << obj\_1->To\_Get\_age() << endl;

cout << endl;

obj\_1->To\_Set(3, 21, "rty", "Gleb", "Glebov", 25);

obj\_1->TO\_Show();

cout << endl;

Studen\* obj\_2 = new Studen(5, 19, "jod", "Vova", "Ivanov", 28);

obj\_2->TO\_Show();

cout << endl;

cout << "Имя -- " << obj\_2->To\_Get\_name() << endl;

cout << "Возраст -- " << obj\_2->To\_Get\_age() << endl;

cout << endl;

obj\_2->To\_Set(4, 22, "rti", "Pety", "Petrov", 25);

obj\_2->TO\_Show();

cout << endl;

Studen obj\_3;

obj\_3.TO\_Show();

cout << endl;

cout << "Имя -- " << obj\_3.To\_Get\_name() << endl;

cout << "Возраст -- " << obj\_3.To\_Get\_age() << endl;

cout << endl;

obj\_3.To\_Set(3, 20, "rty", "Ivan", "Ivanov", 25);

obj\_3.TO\_Show();

cout << endl;

Studen obj\_4(5, 19, "jod", "Vova", "Ivanov", 28);

obj\_4.TO\_Show();

cout << endl;

cout << "Имя -- " << obj\_4.To\_Get\_name() << endl;

cout << "Возраст -- " << obj\_4.To\_Get\_age() << endl;

cout << endl;

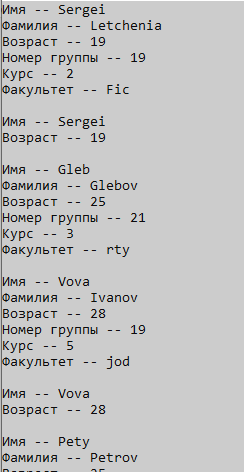
obj\_4.To\_Set(4, 22, "rti", "Pety", "Petrov", 25);

obj\_4.TO\_Show();

cout << endl;

return 0;

}



|  |
| --- |
| Human |
| - string name  - int age  -string surname |
| + Human()  + Human ( int, string , string)  + ~ Human()  + void TO\_Show ()  + void To\_Set ( int, string , string) |

|  |
| --- |
| Student |
| -int grup\_n  -int curs  -string faculty |
| + Student ()  + Student (int g,int c,string f,stringn,strings,int a):Human(n,s,a)  + ~ Student()  +void TO\_Show ()  + int To\_get \_age()  +string To\_get\_name()  + void To\_Show((int g,int c,string f,stringn,strings,int a) |

Вывод: изучил правил наследования классов.