МИНОБРНАУКИ РОССИИ

ФГБОУ ВО Череповецкий государственный университет

Институт информационных технологий

Кафедра: Математическое и программное

обеспечение ЭВМ

Дисциплина: Объектно-ориентированное

программирование

ЛАБОРАТОРНАЯ РАБОТА №9

**ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ОТНОШЕНИЯ**

**МНОЖЕСТВЕННОГО НАСЛЕДОВАНИЯ**

Выполнил:

студент гр.1ИСб-00-21оп

Лебедева А. В.

Проверил:

Шаханов Н. И.

Череповец, 2020 г

Цель работы: изучить принципы и механизмы множественного наследования, правила доступа к базовым классам; приобрести практические навыки работы с базовыми и производными классами при множественном наследовании.

**Задания**

1. Используя предыдущую программу, создайте новый производный класс с применением множественного наследования.

2. Проверьте работоспособность АТД и производных классов на тестовом наборе данных

Текст программы:

#include <iostream>

#include <string>

using namespace std;

class name;

class style;

class Optica;

class name

{

protected:

string tip; //тип

string appointment; // назначение

public:

name(string t, string a); // конструктор

//~form(); // деструктор

void print\_N() const; //селектор

};

class style

{

protected:

string color;

double veight;

public:

style(string c, double v);

void print\_S() const;

};

class Optica : public name, public style

{

private:

string linz;

double power;

public:

//Optica(); // констр. по умолчанию

//Optica(const Optica&); // копирующий конструктор

Optica(string t, string a, string l, int p, double v, string c); // конструктор с параметрами

// ~Optica(); // деструктор

void print() const;// селектор

};

name::name(string t, string a)

{

tip = t;

appointment = a;

}

void name::print\_N() const

{

cout << "Тип: " << tip << endl;

cout << "Назначение: " << appointment << endl;

}

style::style(string c, double v)

{

color = c;

veight = v;

}

void style::print\_S() const

{

cout << "Цвет: " << color << endl;

cout << "Вес: " << veight << endl;

}

Optica::Optica(string t, string a, string l, int p, double v, string c) : name(t, a), style (c,v)

{

linz = l;

power = p;

}

void Optica::print() const

{

cout << "Тип: " << tip << endl;

cout << "Назначение: " << appointment << endl;

cout << "Линза: " << linz << endl;

cout << "Мощност: " << power << endl;

cout << "Цвет: " << color << endl;

cout << "Вес: " << veight << endl << endl;

}

int main()

{

setlocale(LC\_ALL, "Rus");

name A("Фотоаппарат", "Дальнего действия");

A.print\_N();

cout << endl;

Optica a("Лупа", "Ближнего действия", "Выпуклая", 13, 0.2, "Жёлтый");

Optica b("Фотоаппарат", "Дальнего действия", "Выпуклая", 13, 0.2, "Красный");

a.print();

b.print();

cout << "Только name:" << endl;

a.print\_N();

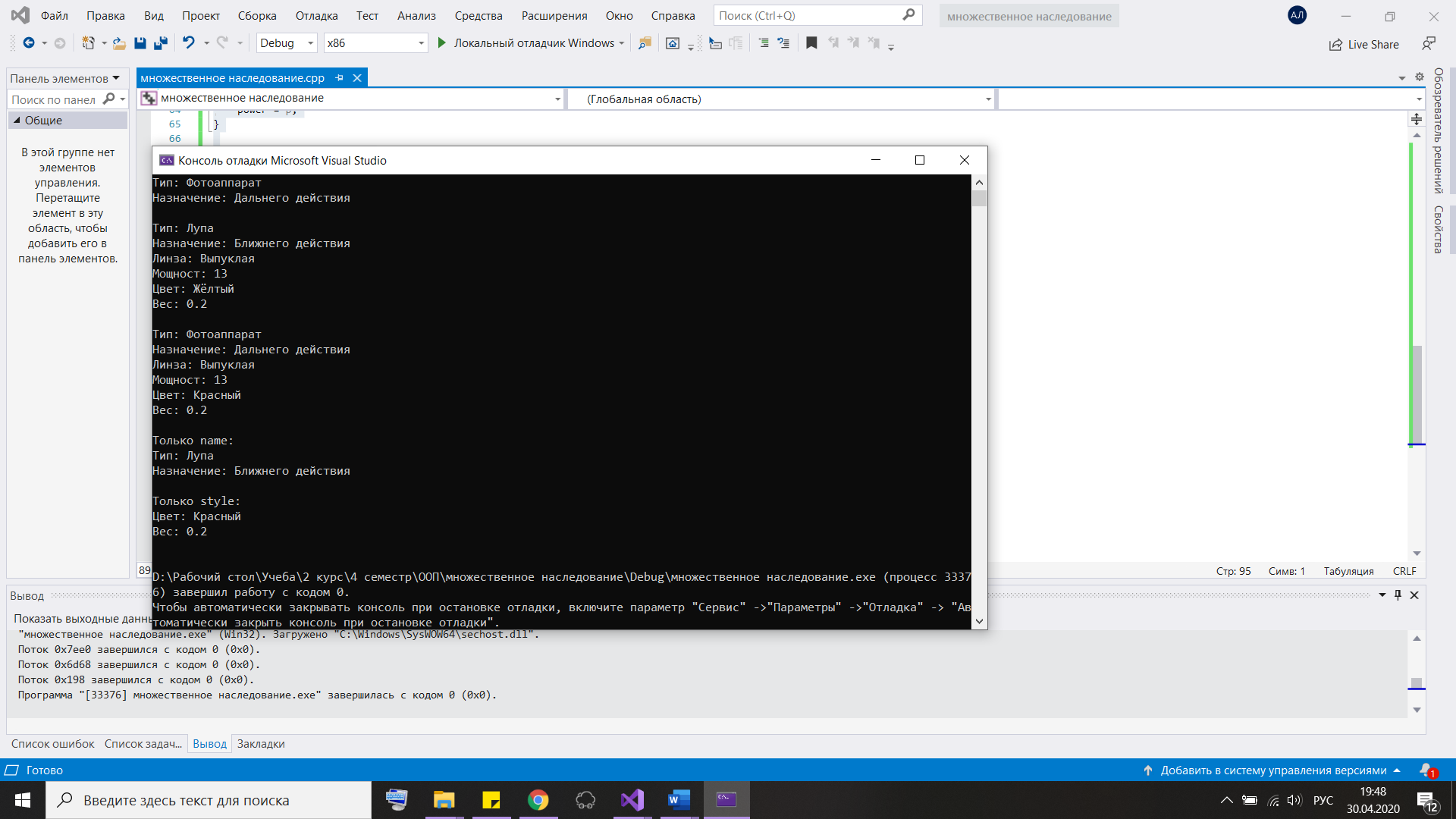
cout << "\nТолько style: " << endl;

b.print\_S();

cout << endl;

}

Вывод:



Контрольные вопросы

1. С какой целью и в каких случаях используется множественное наследование?

Множественное наследование (multiple inheritance) делает возможным получение производного класса от нескольких базо-вых классов. Синтаксис заголовка класса расширяется, чтобы можно было использовать список базовых классов с атрибутами доступа.

1. Опишите синтаксис заголовка производного класса при множественном наследовании.

class Optica : public name, public style{}

1. Дайте определение ориентированного ациклического графа.

Это граф, узлы которого являются классами, а ориентированные ребра направлены от производных классов к базовым.

1. Где инициируются виртуальные базовые классы?

Перед любыми не виртуальными базовыми классами и в том порядке, в котором они появляются в ПАГе наследования при про-смотре его снизу-вверх и слева направо.

1. Что будет, если из объявлений классов Cow и Buffalo убрать ключевое слово virtual (см. последний пример в теоретических положениях)?

Ключевое слово virtual в классе Cow и классе Buffalo предот-вращает многократное копирование полей данных weight, price, color из предков класса Beefalo.