МИНОБРНАУКИ РОССИИ

ФГБОУ ВО Череповецкий государственный университет

Институт информационных технологий

Кафедра: Математическое и программное

обеспечение ЭВМ

Дисциплина: Объектно-ориентированное

программирование

ЛАБОРАТОРНАЯ РАБОТА №8

**ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ОТНОШЕНИЯ**

**ПРОСТОГО НАСЛЕДОВАНИЯ**

Выполнил:

студент гр.1ИСб-00-21оп

Лебедева А. В.

Проверил:

Шаханов Н. И.

Череповец, 2020 г

Цель работы: изучить способы создания производного класса и особенности работы с ним, правила инициализации и доступа к элементам производного класса; приобрести практические навыки наследования.

**Задания**

1. Создайте производный класс для АТД, реализованного по заданию лабораторной работы 7, используя одиночное наследование.

2. Проверьте работоспособность АТД и производного класса на тестовом наборе данных

Вариант 17 – Оптические приборы

Текст программы:

#include <iostream>

#include <string>

using namespace std;

class form;

class Optica;

class form

{

protected:

string tip; //тип

string appointment; // назначение

public:

form(); // констр. по умолчанию

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

form(const form&);

//~form(); // деструктор

void printF() const; //селектор

};

class Optica : public form

{

private:

string linz;

double power;

double veight; // вес

public:

//Optica(); // констр. по умолчанию

Optica(const Optica&); // копирующий конструктор

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

// ~Optica(); // деструктор

void print() const;// селектор

Optica& operator = (const Optica& obj); // присваивание

bool& operator == (Optica& obj); // сравнение

};

form::form()

{

tip = "tip";

appointment = "appointment";

}

form::form(string t, string a)

{

tip = t;

appointment = a;

}

void form::printF() const

{

cout << "Тип: " << tip << endl;

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

}

form::form(const form& obj)

{

tip = "";

appointment = "";

tip = obj.tip;

appointment = obj.appointment;

}

Optica::Optica(string t, string a, string l, int p, float v) : form(t, a)

{

linz = l;

power = p;

veight = v;

}

Optica::Optica(const Optica& obj)

{

linz = obj.linz;

power = obj.power;

veight = obj.veight;

}

void Optica::print() const

{

cout << "Линза: " << linz << endl;

cout << "Мощност: " << power << endl;

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

}

Optica& Optica :: operator=(const Optica& obj)

{

if (this != &obj)

{

tip = ""; appointment = "";

linz = "";

tip = obj.tip;

appointment = obj.appointment;

linz = obj.linz;

power = obj.power;

veight = obj.veight;

}

return (\*this);

}

bool& Optica :: operator== (Optica& obj)

{

bool flag = true;

int a = 0;

if (tip == obj.tip) a++;

if (appointment == obj.appointment) a++;

if (linz == obj.linz) a++;

if (power == obj.power) a++;

if (veight == obj.veight) a++;

if (a == 5) flag = true; else flag = false;

return flag;

}

int main()

{

setlocale(LC\_ALL, "Rus");

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

A.printF();

cout << endl;

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

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

a.printF();

a.print();

b.print();

cout << "Cравнение: " << endl;

if (a == b) cout << "Обьекты равны" << endl << endl;

else cout << "Обьекты не равны" << endl << endl;

cout << "Присваивание: " << endl;

a = b;

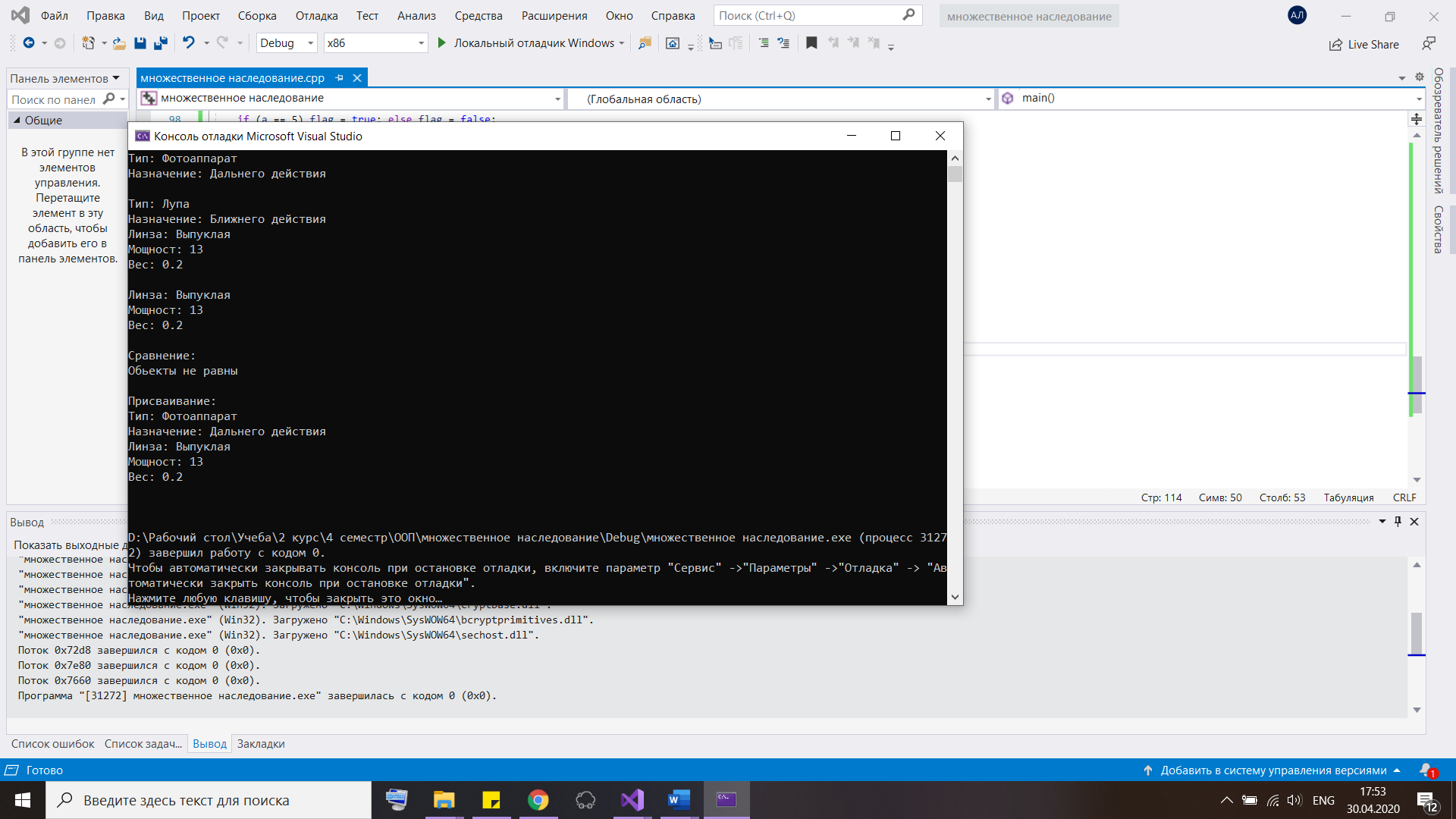
a.printF();

a.print();

cout << endl;

}

Вывод:



# Контрольные вопросы

1. Дайте определение наследования.

Наследование (inheritance) – это механизм получения нового класса на основе существующего класса.

1. Какие модификаторы прав доступа к производному классу вы знаете и в чем их особенности?

Модификаторы прав доступа к членам класса (public, protected и private) могут применяться в объявлении класса в любом порядке и сколько угодно раз. Открытый член доступен во всей области видимости, где виден класс. Закрытый член доступен другим функциям-членам своего класса. Защищенный член доступен не только другим функциям-членам своего класса, но и функциям-членам класса, унаследованного непосредственно от данного класса. Если все члены открыты, то ключевое слово class можно заменить ключевым словом struct.

1. Как выполняется конструктор при наследовании?

При наследовании и инициализации членов класса конструкторы выполняются в следующем порядке:

1. Базовые классы инициализируются в порядке объявления.

2. Члены инициализируются в порядке объявления.

1. Дайте определение чисто виртуальной функции.

Чисто виртуальная функция – это виртуальная функция-член, тело которой не определено. Она объявляется внутри класса.

1. Какие операторы используются для разыменования указателя на член класса?

Для разыменования указателя на член класса используются два оператора: .\* и →\*.