Министерство образования и науки РФ

Севастопольский государственный университет

Кафедра информатики и управления в технических системах

ОТЧЕТ

по лабораторной работе №4

ИССЛЕДОВАНИЕ МЕХАНИЗМА МНОЖЕСТВЕННОГО НАСЛЕДОВАНИЯ

по дисциплине «Объектно-ориентированное программирование»

Выполнил:

Студент группы ИВТ/б 22-о

Черняев Н.Г.

Проверил:

Тимофеев И.С.

г. Севастополь 2018

**Цель работы:**

Исследование основных средств описания класса, наследующего свойства нескольких базовых классов.

**Вариант №10**

Описать интерфейс ввода-вывода. Описать иерархию классов, заданную по варианту. Для каждого класса описать конструкторы и деструктор (по необходимости), функции ввода и вывода значений полей. В каждом классе должны присутствовать минимум одно уникальное поле и один уникальный метод. Проиллюстрировать корректную работу механизма множественного наследования — для этого создать объекты базовых классов и заполнить их поля данными, вывести на печать. Создать объект класса-наследника, его поля заполнить значениями соответствующих полей базовых классов. Вывести на печать данные полученного объекта.

**Базовые классы: Ткань (название, цвет, цена), Стиль (название). Класс наследник: Костюм.**

**Текст программы:**

#include "stdafx.h"

#include <iostream>

#include <string>

using namespace std;

class Cloth {

protected:

string title;

string color;

int price;

public:

Cloth() {

title = "Basic";

color = "Black";

price = 0;

};

Cloth(string \_title, string \_color, int \_price) {

title = \_title;

color = \_color;

price = \_price;

};

void InputCloth() {

cout << "Enter Cloth title" << endl;

cin >> title;

cout << "Enter Cloth color" << endl;

cin >> color;

cout << "Enter Cloth price" << endl;

cin >> price;

};

void OutputCloth() {

cout << "Cloth title: " << title << endl;

cout << "Cloth color: " << color << endl;

cout << "Cloth price: " << price << endl;

};

};

class Style {

protected:

string title;

public:

Style() { title = "Normal"; };

Style(string \_title) { title = \_title; };

void InputStyle() {

cout << "Enter Style title" << endl;

cin >> title;

};

void OutputStyle() { cout << "Style title: " << title << endl; };

};

class Costume : public Cloth, public Style {

protected:

int size;

public:

Costume() { size = 0; };

Costume(string \_title1, string \_color, int \_price, string \_title2, int \_size) : Cloth(\_title1, \_color, \_price), Style(\_title2) { size = \_size; };

void OutputCostume() {

cout << "Cloth of Costume:" << endl;

OutputCloth();

cout << "Style of Costume:" << endl;

OutputStyle();

cout << "Size of Cotume:" << endl;

cout << "Costume size: " << size << endl;

};

void InputCostume() {

cout << "Enter Cloth of Costume" << endl;

InputCloth();

cout << "Enter Style of Costume" << endl;

InputStyle();

cout << "Enter Costume size" << endl;

cin >> size;

};

};

int main()

{

Cloth Cloth1;

Style Style1;

Costume Costume1;

cout << "Enter Cloth1" << endl;

Cloth1.InputCloth();

Cloth1.OutputCloth();

cout << "Enter Style1" << endl;

Style1.InputStyle();

Style1.OutputStyle();

cout << "Enter Costume1" << endl;

Costume1.InputCostume();

Costume1.OutputCostume();

return 0;

}

**Тестовые примеры:**

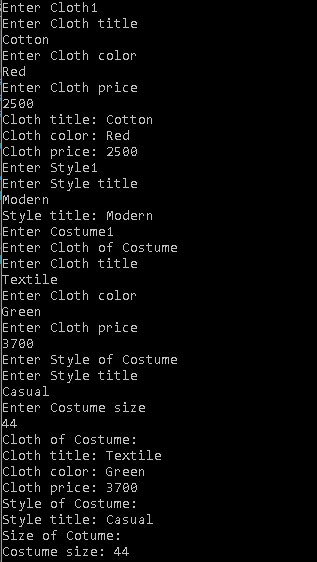


рис.1 – пример работы программы

**Вывод:**

В данной лабораторной работе были исследованы основные средства описания класса, наследующего свойства нескольких базовых классов, а также получены практические навыки использования множественного наследования.