郑 州 轻 工 业 大 学

**实 验 报 告**

**课程名称： 面向对象程序设计**

**姓 名： 原彬贺**

**学 号： 542001020223**

**专业班级： 计算机科学与技术20-02**

**任课教师： 王岩**

**2021 年 10 月 27 日**

**实验报告正文**

**实验一**

**类与对象（一）**

**一、 实验目的**

1.掌握类的继承、派生的概念、声明和定义方式；

2.学会定义和使用类的继承关系；

3.掌握私有派生和共有派生的访问特性及使用方法；

4.了解虚基类在解决二义性问题中的作用。

**二、 问题的本质和抽象描述**

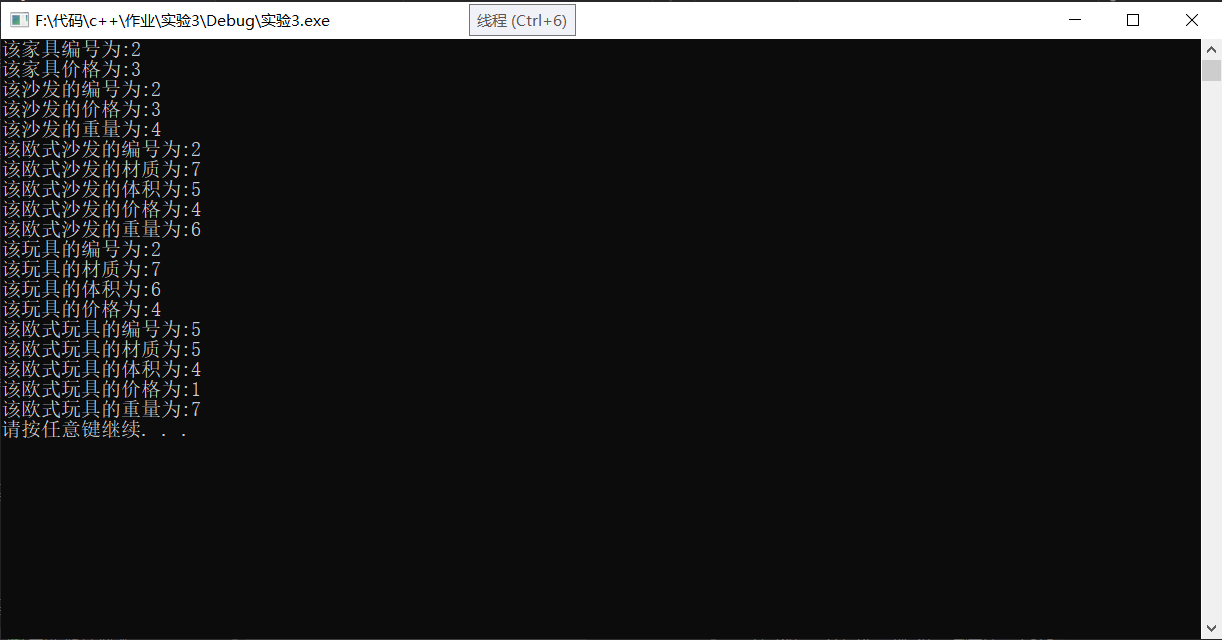
定义一个基类，再将其派生类通过共有继承成为虚基类，并在其基类中定义输入输出的虚函数，创建价格与其他数据关系的成员函数，并添加相关数据，在主函数中定义这些类的对象，并且访问。

**三、 测试**

**1.方案**

**先定义一个家具基类，包含价格与编号两个数据成员，并编写两个输入输出的虚函数，沙发类等其他类通过继承家具类，编写其构造函数，欧洲玩具类通过继承欧洲沙发类和玩具类而形成一个新的派生类，在主函数中对其分别进行调用。**

**2.结果**



**四、 总结与讨论**

在编写过程中，通过对构造函数的使用，如果其父类有共同的基类需要在构造函数时将参数传递到其共同的基类，在通过虚基类继承时，应当将继承方式设置为公有继承。在实验后，懂得了类的继承，派生的概念和声明和定义方式，收获很多，巩固了之前学到的知识。

**附：源代码**

#include<iostream>

#include<string>

#include<fstream>

using namespace std;

class Furniture

{

protected :

int Number, Prices;

public :

Furniture(int num,int p);

virtual void output();

virtual void input();

virtual double Count\_Prices();

};

Furniture::Furniture(int num, int p)

{

Number = num;

Prices = p;

}

void Furniture::output()

{

cout << "该家具编号为:" << Number << endl;

cout << "该家具价格为:" << Prices << endl;

}

void Furniture::input()

{

cin >> Number;

cin >> Prices;

}

double Furniture:: Count\_Prices()

{

return Prices;

}

class Sofa : virtual public Furniture

{

protected :

int weight;

public :

virtual double Count\_Prices();

virtual void output();

virtual void input();

Sofa(int num, int p ,int w) :Furniture(num, p)

{

weight = w;

}

};

void Sofa::output()

{

cout << "该沙发的编号为:" << Number << endl;

cout << "该沙发的价格为:" << Prices << endl;

cout << "该沙发的重量为:" << weight << endl;

}

void Sofa::input()

{

cin >> Number;

cin >> Prices;

cin >> weight;

}

double Sofa::Count\_Prices()

{

Prices = weight \* 10;

return Prices;

}

class EuroSofa :virtual public Sofa

{

protected :

int Volume ,Material;

public :

virtual double Count\_Prices();

virtual void output();

virtual void input();

EuroSofa(int num, int p, int w, int m, int v) :Furniture(num, p), Sofa(num, p, w)

{

Material = m;

Volume = v;

}

};

void EuroSofa::output()

{

cout << "该欧式沙发的编号为:" << Number << endl;

cout << "该欧式沙发的材质为:"<<Material <<endl;

cout << "该欧式沙发的体积为:" << Volume << endl;

cout << "该欧式沙发的价格为:" << Prices << endl;

cout << "该欧式沙发的重量为:" << weight << endl;

}

void EuroSofa::input()

{

cin >> Number;

cin >> Prices;

cin >> weight;

cin >> Volume;

cin >> Material;

}

double EuroSofa::Count\_Prices()

{

if (Material == 1)

Prices = Volume \* 100;

else if (Material == 2)

Prices = Volume \* 50;

else if (Material == 3)

Prices = Volume \* 20;

else

Prices = 5;

return Prices;

}

class Toy :virtual public Furniture

{

protected:

int Volume, Material;

virtual void output();

virtual void input();

public:

virtual double Count\_Prices();

Toy(int num, int p,int v,int m) :Furniture(num, p)

{

Volume = v;

Material = m;

}

};

void Toy::output()

{

cout << "该玩具的编号为:" << Number << endl;

cout << "该玩具的材质为:" << Material<< endl;

cout << "该玩具的体积为:" << Volume<< endl;

cout << "该玩具的价格为:" << Prices << endl;

}

void Toy::input()

{

cin >> Number;

cin >> Prices;

cin >> Volume;

cin >> Material;

}

double Toy::Count\_Prices()

{

if (Material == 1)

Prices = Volume \* 50;

else if (Material == 2)

Prices = Volume \* 30;

else if (Material == 3)

Prices = Volume \* 5;

else

Prices = 10;

return Prices;

}

class EuroToy : virtual public EuroSofa, virtual public Toy

{

protected:

int Volume, Material,Prices;

public:

virtual double Count\_Prices();

virtual void output();

virtual void input();

EuroToy(int num, int p,int w,int m,int v) :Furniture(num, p), Sofa(num, p, w),EuroSofa(num, p,w,m,v),Toy(num,p,v,m)

{

Volume = v;

Material = m;

Prices = p;

}

};

void EuroToy::output()

{

cout << "该欧式玩具的编号为:" << Number << endl;

cout << "该欧式玩具的材质为:" << Material << endl;

cout << "该欧式玩具的体积为:" << Volume << endl;

cout << "该欧式玩具的价格为:" << Prices << endl;

cout << "该欧式玩具的重量为:" << weight << endl;

}

void EuroToy::input()

{

cin >> Number;

cin >> Prices;

cin >> weight;

cin >> Volume;

cin >> Material;

}

double EuroToy::Count\_Prices()

{

if (Material == 1)

Prices = Volume \* 50;

else if (Material == 2)

Prices = Volume \* 30;

else if (Material == 3)

Prices = Volume \* 5;

else

Prices = 10;

return Prices;

}

int main()

{

Furniture a(2, 3),\*p;

Sofa b(2, 3, 4);

EuroSofa c(2, 4, 6, 7, 5);

Toy d(2, 4, 6, 7);

EuroToy e(5, 1, 7, 5, 4);

p = &a;

p->output();

p=&b;

p->output();

p = &c;

p->output();

p = &d;

p->output();

p = &e;

p->output();

system("pause");

}