**实验六 派生类和继承**

**一、实验目的**

1）掌握继承机制及派生类的定义及用法;

2）继承中常见问题的处理方法。

**二、实验内容**

**2.1派生类----单继承**

**1. 习题4.14，4.16，4.17**

**程序：**

**4.14**

**#include <iostream>**

**using namespace std;**

**class A{**

**private:**

**int a;**

**public:**

**A(){**

**a=0;**

**}**

**A(int i){**

**a=i;**

**}**

**void Print()**

**{cout<<a<<",";}**

**};**

**class B: public A{**

**private:**

**int b1,b2;**

**public:**

**B(){**

**b1=0;**

**b2=0;**

**}**

**B(int i){**

**b1=i;**

**b2=0;**

**}**

**B(int i,int j,int k):A(i),b1(j),b2(k)**

**{}**

**void Print(){**

**A::Print();**

**cout<<b1<<","<<b2<<endl;**

**}**

**};**

**int main(){**

**B ob1,ob2(1),ob3(3,6,9);**

**ob1.Print();**

**ob2.Print();**

**ob3.Print();**

**return 0;**

**}**

**4.16**

**#include <iostream>**

**using namespace std;**

**class A{**

**public:**

**A(int i,int j){x=i;y=j;}**

**int sum(){return x+y;}**

**private:**

**int x,y;**

**};**

**class B:public A{**

**public:**

**B(int i,int j,int k,int l);**

**int sum(){return w+h;}**

**private:**

**int w,h;**

**};**

**B::B(int i,int j,int k,int l):A(i,j){w=k;h=l;}**

**void f(A &s){**

**cout<<s.sum()<<endl;**

**}**

**int main(){**

**B ob(1,3,5,7);**

**f(ob);**

**return 0;**

**}**

**4.17**

**#include <iostream>**

**using namespace std;**

**class A{**

**int a,b;**

**public:**

**A(int i,int j){**

**a=i;**

**b=j;**

**}**

**void Move(int x,int y){**

**a+=x;**

**b+=y;**

**}**

**void Show(){**

**cout<<"("<<a<<","<<b<<")"<<endl;**

**}**

**};**

**class B:private A{**

**int x,y;**

**public:**

**B(int i,int j,int k,int l):A(i,j){**

**x=k;**

**y=l;**

**}**

**void Show(){**

**cout<<x<<","<<y<<endl;**

**}**

**void fun(){**

**Move(3,5);**

**}**

**void f1(){**

**A::Show();**

**}**

**};**

**int main(){**

**A e(1,2);**

**e.Show();**

**B d(3,4,5,6);**

**d.fun();**

**d.Show();**

**d.f1();**

**return 0;**

**}**

**执行结果：**

**4.14**

**0,0,0**

**0,1,0**

**3,6,9**

**4.16**

4

4.17

(1,2)

5,6

(6,9)

**2**．分析下面的程序，指出程序运行的结果

#include<iostream.h>

class CBase

{public:

void fn1();

};

void CBase::fn1()

{cout<<"调用基类类的函数fn1()\n";

}

class CDerived:public CBase

{

public:

void fn1();

};

void CDerived::fn1()

{cout<<"调用派生类的函数fn1()\n";

}

int main()

{

CDerived d1;

CBase \*pb=&d1;

CBase &pd=d1;

d1.fn1();

pb->fn1();

pd.fn1();

return 0;

}

**运行结果：**

**调用派生类的函数fn1()**

**调用基类类的函数fn1()**

**调用基类类的函数fn1()**

**3、**编写并调试程序：

1）p197 4.22

程序：

#include<iostream>

#include <string>

using namespace std;

class person{

public:

int id;

string name;

void setdata(){

cout<<"请输入ID和姓名"<<endl;

cin>>id>>name;

}

void print(){

cout<<"ID:"<<id<<"name:"<<name<<endl;

}

};

class stu:public person{

public:

int Class;

double score;

void studata(){

cout<<"请输入学生的班级和成绩";

cin>>Class>>score;

}

void stuprint(){

cout<<"class:"<<Class<<"score:"<<score<<endl;

}

};

class teacher:public person{

public:

string title;

string department;

void teacherdata(){

cout<<"请输入老师的职称和部门";

cin>>title>>department;

}

void teacherprint(){

cout<<"title:"<<title<<"department:"<<department<<endl;

}

};

int main(){

stu s1;

cout<<"学生："<<endl;

s1.setdata();

s1.studata();

s1.print();

s1.stuprint();

teacher t;

cout<<"老师："<<endl;

t.setdata();

t.teacherdata();

t.print();

t.teacherprint();

return 0;

}

运行结果：

学生：

请输入ID和姓名

1 zhang

请输入学生的班级和成绩3 100

ID:1name:zhang

class:3score:100

老师：

请输入ID和姓名

2 zhang

请输入老师的职称和部门dd ff

ID:2name:zhang

title:dddepartment:ff

**2**）定义一个图形类，其中有保护类型的成员数据：高度和宽度，一个公有的构造函数。由该图形类建立两个派生类：矩形类和等腰三角形类。在每个派生类中都包含一个函数area()，分别用来计算矩形和等腰三角形的面积。

**①．**程序设计如下：

#include<iostream.h>

class picture

{

protected:

double h,w;

public:

picture (double i,double j){h=i;w=j;}

};

class jpicture:public picture

{double mi;

public:

jpicture(double i,double j):picture(i,j){}

void area(){

cout<<"area:"<<h\*w<<endl;

}

};

class sanjiaoxing:public picture

{

double mi;

public:

sanjiaoxing(double i,double j):picture(i,j){}

void area(){

cout<<"area:"<<h\*w/2<<endl;

}

};

void main()

{

jpicture a(1,2);

sanjiaoxing b(3,4);

a.area();

b.area();

}

**②．程序运行结果：**

**area:2**

**area:6**

**2.2多继承和虚基类**

**1．习题 4.12， 4.13， 4.15， 4.18**

程序

4.12

#include <iostream>

using namespace std;

class B1{

public:

B1(int i){

b1=i;

cout<<"Constructor B1."<<endl;

}

void Print(){

cout<<b1<<endl;

}

private:

int b1;

};

class B2{

public:

B2(int i){

b2=i;

cout<<"Contructor B2."<<endl;

}

void Print(){

cout<<b2<<endl;

}

private:

int b2;

};

class A:public B2,public B1{

public:

A(int i,int j,int l);

void Print();

private:

int a;

};

A::A(int i,int j,int l):B1(i),B2(j){

a=l;

cout<<"Constructor A."<<endl;

}

void A::Print(){

B1::Print();

B2::Print();

cout<<a<<endl;

}

int main(){

A aa(3,2,1);

aa.Print();

return 0;

}

4.13

#include <iostream>

using namespace std;

class Main{

protected:

char \* mainfood;

public:

Main(char \* name){

mainfood=name;

}

};

class Sub{

protected:

char \* subfood;

public:

Sub(char \* name){

subfood=name;

}

};

class Menu:public Main ,public Sub{

public:

Menu(char \* m,char \* s):Main(m),Sub(s){}

void show();

};

void Menu::show(){

cout<<"主食="<<mainfood<<endl;

cout<<"副食="<<subfood<<endl;

}

int main(){

Menu m("bread","steak");

m.show();

return 0;

}

4.15

#include <iostream>

using namespace std;

class B1{

int b1;

public:

B1(int i){

b1=i;

cout<<"construtor B1."<<i<<endl;

}

void print(){

cout<<b1<<endl;

}

};

class B2{

int b2;

public:

B2(int i){

b2=i;

cout<<"constructor B2."<<i<<endl;

}

void print(){

cout<<b2<<endl;

}

};

class B3{

int b3;

public:

B3(int i){

b3=i;

cout<<"constructor B3."<<i<<endl;

}

int getb3(){return b3;}

};

class A:public B2,public B1{

int a;

B3 bb;

public:

A(int i,int j,int k,int l):B1(i),B2(j),bb(k){

a=l;

cout<<"constructor A."<<l<<endl;

}

void print(){

B1::print();

B2::print();

cout<<a<<","<<bb.getb3()<<endl;

}

};

int main(){

A aa(1,2,3,4);

aa.print();

return 0;

}

4.18

#include <iostream>

using namespace std;

class base1{

public:

base1(){

cout<<"class base1"<<endl;

}

};

class base2{

public:

base2(){

cout<<"class base2"<<endl;

}

};

class level1:public base2,virtual public base1{

public:

level1(){

cout<<"class level1"<<endl;

}

};

class level2:public base2,virtual public base1{

public:

level2(){

cout<<"class level2"<<endl;

}

};

class toplevel:public level1,virtual public level2{

public:

toplevel(){

cout<<"class toplevel"<<endl;

}

};

int main(){

toplevel obj;

return 0;

}

执行结果：

4.12

Contructor B2.

Constructor B1.

Constructor A.

3

2

1

4.13

主食=bread

副食=steak

4.15

constructor B2.2

construtor B1.1

constructor B3.3

constructor A.4

1

2

4,3

4.18

class base1

class base2

class level2

class base2

class level1

class toplevel

**2．分析下面的程序，指出程序运行的结果：**

#include<iostream.h>

class CBase1

{

int x;

public:

CBase1()

{x=0;cout<<"调用构造函数CBase1()!\n";}

CBase1(int a)

{x=1;cout<<"调用构造函数CBase1(int)!\n";}

~CBase1(){cout<<"调用析构函数~CBase1()!\n";}

};

class CBase2

{int y;

public:

CBase2()

{y=0;cout<<"调用构造函数CBase2()!\n";}

CBase2(int a)

{y=a;cout<<"调用构造函数CBase2(int)!\n";}

~CBase2(){cout<<"调用析构函数~CBase2()!\n";}

};

class A

{int x;

public:

A(){x=0;cout<<"调用构造函数A()!\n";}

A(int a){x=a;cout<<"调用构造函数A(int)!\n";}

~A(){cout<<"调用析构函数~A()!\n";}

};

class CDerived:public CBase1,virtual public CBase2

{

A a;

public:

CDerived()

{cout<<"调用构造函数CDerived()!\n";}

CDerived(int x,int y,int z):a(x),CBase1(y),CBase2(z)

{cout<<"调用构造函数CDerived(int,int,int)!\n";}

~CDerived(){cout<<"调用析构函数~CDerived()!\n";}

};

void main()

{

CDerived \*x=new CDerived;

CDerived y(2,3,4);

delete x;

cout<<"main()函数结束!\n";

}

**程序运行结果：**

调用构造函数CBase2()!

调用构造函数CBase1()!

调用构造函数A()!

调用构造函数CDerived()!

调用构造函数CBase2(int)!

调用构造函数CBase1(int)!

调用构造函数A(int)!

调用构造函数CDerived(int,int,int)!

调用析构函数~CDerived()!

调用析构函数~A()!

调用析构函数~CBase1()!

调用析构函数~CBase2()!

main()函数结束!

调用析构函数~CDerived()!

调用析构函数~A()!

调用析构函数~CBase1()!

调用析构函数~CBase2()!

**3．编写并调试程序**

**1）p196 4.20， 4.21**

**4.20**

**#include <iostream>**

**using namespace std;**

**class are\_cl{**

**protected:**

**double height;**

**double width;**

**public:**

**are\_cl(double r,double s){**

**height=r;**

**width=s;**

**}**

**virtual double area()=0;**

**};**

**class rectangle:public are\_cl{**

**public:**

**rectangle(double r,double s):are\_cl(r,s){}**

**double area(){**

**return height\*width;**

**}**

**};**

**class isosceles:public are\_cl{**

**public:**

**isosceles(double r,double s):are\_cl(r,s){}**

**double area(){**

**return height\*width/2;**

**}**

**};**

**int main(){**

**rectangle a(10.0,5.0);**

**isosceles b(4.0,6.0);**

**are\_cl \*p;**

**p=&a;**

**cout<<p->area()<<endl;**

**p=&b;**

**cout<<p->area()<<endl;**

**return 0;**

**}**

**4.21**

**#include <iostream>**

**#include <string>**

**using namespace std;**

**class Time{**

**public:**

**Time(int h,int m,int s){**

**hours=h;**

**minutes=m;**

**seconds=s;**

**}**

**void display(){**

**cout<<"出生时间："<<hours<<"时"<<minutes<<"分"<<seconds<<"秒"<<endl;**

**}**

**protected:**

**int hours,minutes,seconds;**

**};**

**class Date{**

**public:**

**Date(int m,int d,int y){**

**month=m;**

**day=d;**

**year=y;**

**}**

**void display(){**

**cout<<"出生年月"<<year<<"年"<<month<<"月"<<day<<"日"<<endl;**

**}**

**protected:**

**int month,day,year;**

**};**

**class Birthtime:public Time,public Date{**

**string Childname;**

**public:**

**Birthtime(string name,int h,int m,int s,int mm,int d,int y):Time(h,m,s),Date(mm,d,y){**

**Childname=name;**

**}**

**void print(){**

**cout<<"姓名："<<Childname<<endl;**

**Time::display();**

**Date::display();**

**}**

**};**

**int main(){**

**Birthtime a("陈子枫",6,59,59,5,5,2016);**

**a.print();**

**return 0;**

**}**

**2**）定义一个人员类cpeople，其属性有（保护类型）：姓名、性别、年龄；从中派生出学生类cstudent，添加属性：学号、入学时间和入学成绩；从cpeople类再派生出教师类cteacher，添加属性：职务、部门、工作时间；有cstudent类派生研究生类cgraduate，添加属性：研究方向和导师，由cgraduate和cteacher共同派生出在职研究生类cgradonwork，分别定义其中的构造函数和输出函数。主函数中定义各种类的对象，并完成测试。

**①．**程序设计如下：

**#include <iostream>**

**#include <string>**

**using namespace std;**

**class cpeople{**

**protected:**

**string name,sex;**

**int old;**

**public:**

**cpeople(string nn,string ss,int oo){**

**name=nn;**

**sex=ss;**

**old=oo;**

**}**

**void print(){**

**cout<<"name: "<<name<<"sex: "<<sex<<"old: "<<old<<endl;**

**}**

**};**

**class cstudent:public cpeople{**

**protected:**

**int ID;**

**string time;**

**int score;**

**public:**

**cstudent(int ii,string tt,int ss,string nn,string s,int oo):cpeople(nn,s,oo){**

**ID=ii;**

**time=tt;**

**score=ss;**

**}**

**cpeople::name;**

**cpeople::old;**

**cpeople::sex;**

**void print(){**

**cout<<"student:"<<endl;**

**cpeople::print();**

**cout<<"ID: "<<ID<<"starttime: "<<time<<"score: "<<score<<endl;**

**}**

**};**

**class cteacher:public cpeople{**

**protected:**

**string title,department,worktime;**

**public:**

**cteacher(string tit,string de,string work,string nn,string s,int oo):cpeople(nn,s,oo){**

**title=tit;**

**department=de;**

**worktime=work;**

**}**

**cpeople::name;**

**cpeople::old;**

**cpeople::sex;**

**void print(){**

**cout<<"teacher:"<<endl;**

**cpeople::print();**

**cout<<"title: "<<title<<"department: "<<department<<"worktime: "<<worktime<<endl;**

**}**

**};**

**class cgraduate:public cstudent{**

**protected:**

**string way,teacher;**

**public:**

**cgraduate(string ww,string tea,int ii,string tt,int ss,string nn,string s,int oo):cstudent(ii,tt,ss,nn,s,oo){**

**way=ww;**

**teacher=tea;**

**}**

**void print(){**

**cout<<"graduate:"<<endl;**

**cstudent::print();**

**cout<<"way: "<<way<<"teacher: "<<teacher<<endl;**

**}**

**};**

**class cgradonwork:public cgraduate,public cteacher{**

**public:**

**cgradonwork(string ww,string tea,int ii,string tt,int ss,string nn,string s,int oo,string tit,string de,string work,string tnn,string ts,int too):cgraduate(ww,**

**tea,ii,tt,ss,nn,s,oo),cteacher(tit,de,work,tnn,ts,too){}**

**void print(){**

**cout<<"gardonwork:"<<endl;**

**cgraduate::print();**

**cteacher::print();**

**}**

**};**

**int main(){**

**cpeople p("zhang","男",20);**

**p.print();**

**cstudent student(1507020329,"2015.09",100,"zhang","男",20);**

**student.print();**

**cteacher t("教授","数学系","2000.03","金","男",60);**

**t.print();**

**cgraduate g("计算机","雷",1507020329,"2015.09",100,"zhang","男",20);**

**g.print();**

**cgradonwork gg("计算机","雷",1507020329,"2015.09",100,"zhang","男",20,"教授","数学系","2000.03","金","男",60);**

**return 0;**

**}**

**②．**程序运行结果：

**name: zhangsex: 男old: 20**

**student:**

**name: zhangsex: 男old: 20**

**ID: 1507020329starttime: 2015.09score: 100**

**teacher:**

**name: 金sex: 男old: 60**

**title: 教授department: 数学系worktime: 2000.03**

**graduate:**

**student:**

**name: zhangsex: 男old: 20**

**ID: 1507020329starttime: 2015.09score: 100**

**way: 计算机teacher: 雷**

**三、实验结论：**