**实验五继承与派生类**

**一、实验目的**

1．理解继承的含义，掌握派生类的定义方法和实现；

2．理解公有继承下基类成员对派生类成员和派生类对象的可见性，能正确地访问继承层次中的各种类成员；

3．解保护成员在继承中的作用，能够在适当的时候选择使用保护成员以便派生类成员可以访问基类的部分非公开的成员。

**二、实验要求**

1．独立完成程序的编辑、编译、调试及运行；

2．对于程序编译及运行中出现的错误，能够进行改正；

**三、实验内容**

1、创建人的类，数据成员包括姓名、性别、身份证号，成员函数包括构造、输出，已此为基类，创建本科生类，增加数据学号和专业，及相应输出。

代码：

#include<iostream>

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

using namespace std;

class people

{

public :

people(char s[10],char sex\_t,long id)

{

strcpy(name,s);sex=sex\_t;pid=id;

}

void print()

{

cout<<name<<" 性别:"<<sex<<"身份证:"<<pid<<" ";

}

private:

char name[10],sex;

long pid;

};

class ben:public people

{

public:

ben(char s[10],char sex\_t,long id,int number,char zhuanye[10]):people(s,sex\_t,id)

{

strcpy(z,zhuanye); num=number;

}

void print2()

{

cout<<"学号:"<<num<<"专业:"<<z<<endl;

}

private:

char z[10];int num;

};

int main()

{

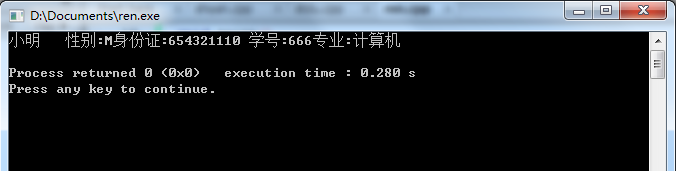
ben one("小明",'M',654321110,666,"计算机");

one.print();

one.print2();

}

截图：



分析：用到了单继承，重新写了构造函数，和一个新的打印函数；

2、生成学生类，其数据成员为姓名、学号、分数，成员函数包括构造、数据修改、输出等。要求用对象数组完成此功能。在此基础上，利用继承与派生，增加派生类研究生，扩充数据成员：研究方向。实例化并执行。

#include<iostream>

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

using namespace std;

class stdudents

{

public :

stdudents(char s1[10],int n,int s)

{

strcpy(name,s1);num=n;score=s;

}

void shuru(char a[10],int n,int s);

void print();

int score;

char name[10];int num;

};

void stdudents::shuru(char a[10],int n,int s)

{

strcpy(name,a);

num=n;

score=s;

}

void stdudents::print()

{

printf("姓名:%s,学号:%d 成绩为:%d\t",name,num,score);

}

class yan:public stdudents

{

public:

yan(char s1[10],int n,int s,char f[10]):stdudents(s1,n,s)

{

strcpy(fang,f);

}

void shuru(char a[10],int n,int s,char f[10])

{

strcpy(name,a);

num=n;

score=s;

strcpy(fang,f);

}

void print2()

{

printf("研究方向: %s\n",fang);

}

private:

char fang[10];

};

int main()

{

yan one("小刚",666,580,"计算机");

char a[10],f[10];int n,s;

cout<<"修改前:\n";

one.print();one.print2();

cout<<"录入修改信息:";

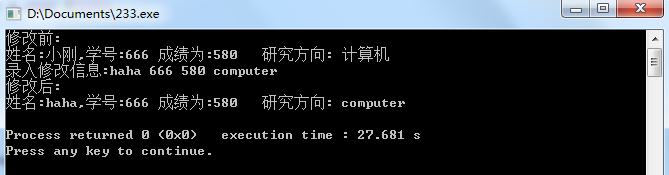
cin>>a>>n>>s>>f;

one.shuru(a,n,s,f);

cout<<"修改后:\n";

one.print();one.print2();

}



分析：加入修改函数可以修改研究生包括学生的信息