C++程序设计 课内实验

实验三 继承与派生实验

1 实验目的

（1）理解继承与派生的意义；

（2）观察构造函数和析构函数的执行过程；

（3）掌握继承与派生的基本概念；

（4）熟悉不同继承方式下对基类成员的访问控制方法。

（5）掌握类的继承与派生的使用方法。

2 实验内容

（1）设计一个用于人事管理的 People (人员)类。考虑到通用性，这里只抽象出所有类型人员都具有的属性：number (编号)、sex (性别)、birthday (出生日期)、id (身份证号)等等。具有的属性如下：姓名 char name[11]、 编号 char number[7]、性别 char sex[3]、 生日 birthday、身份证号 char id[20]。其中"出生日期”声明为一个“日期”类内嵌子对象。用成员函数实现对人员信息的录入和显示。要求包括：构造函数和析构函数、拷贝构造函数、内联成员函数、组合。在测试程序中声明 people 类的对象数组，录入数据并显示。Number:1; name:Zhangsan ;Sex:m ;Birthday:2000 Year 1 Month 1 Day; ID:10497202201.

（2）从 people(人员)类派生出 student(学生)类，添加属性:班号 char classNO[7]；从 people 类派生出 teacher (教师)类，添加属性：职务 char pship[11]、部门 char departt[21]。从 student 类中派生出 graduate (研究生)类，添加属性：专业 char subject[21]、导师 teacher adviser；从 graduate 类和 teacher 类派生出 TA (助教博士生)类，重载相应的成员函数，测试这些类。

3 主要仪器实验设备及相关参数

1）计算机；

2）Dev-C++编译器（仅为推荐编译器，非强制编译器）；

4 实验报告要求

1）写出 C++程序的撰写思路及核心的源代码；

2）程序的实现效果图，附录在实验报告中；

3）实验过程中是否遇到错误或困难？如有错误或困难，你是如何解决的？

注意事项

1）程序的实验效果图，原则上不需要彩色打印；实验报告可以打印，但是签名必须手写。

参考程序代码：

实验内容（1）

```
#include<iostream>

using namespace std;

class Date //日期类

{

private:

        int year;

        int month;

        int day;

public:

        Date(){} //默认构造

        Date(int y,int m,int d) //带参构造

{

        year=y;

        month=m;

        day=d;

}

void set() //设置数据函数

{

        cin>>year>>month>>day;

}
```

```
void display() //显示函数
{
    cout<<year<<" Year "<<month<<" Month "<<day<<" Day ";
}
};
class People //人员类
{
private:
    string name;
    int num;
    char sex;
    Date birthday;
    char ID[18];
public:
    People(){};//默认构造
    People(int n,int y,int m,int d,char id[18],char s='m'):birthday(y,m,d) {
    num=n;
    sex=s;
    strcpy(ID,id);
};//有默认值的带参构造
People(People& p) //拷贝构造
{ name= p.name;
```

```
num=p.num;

sex=p.sex;

birthday=p.birthday;

strcpy(ID,p.ID);

}

void input() //输入函数

{

cout<<"Enter data:"<<endl;

cout<<"Name:";

cin>>name;

cout<<"Number:";

cin>>num;

cout<< "Sex(m/f) :";

cin>>sex;

cout<<"Birthday(Year/Month/Day):";

birthday.set();

cout<<"ID:";

cin>>ID;

cout<<endl;

};

void output() //输出函数

{
```

```
cout<<"Number:"<<num<<endl;

cout<<"Name:"<<name<<endl;

cout<<"Sex:"<<sex<<endl;

cout<<"Birthday:";

birthday.display();

cout<<endl;

cout<<"ID:"<<ID<<endl;

};

~People() //析构函数

{

cout<<num<<" No. Person has been entered."<<endl;

}

};

int main()

{People p1;

p1.input();

p1.output();

return 0;

}
```

实验内容（2）

```
#include<iostream>

using namespace std;

class Date //日期类

{

private:

        int year;

        int month;

        int day;

public:

        Date(){} //默认构造

        Date(int y,int m,int d) //带参构造

{

        year=y;

        month=m;

        day=d;

}

void set() //设置数据函数

{

        cin>>year>>month>>day;

}

void display() //显示函数
```

```
{
        cout<<year<<" Year "<<month<<" Month "<<day<<" Day ";
}
};
class People //人员类
{
private:
        string name;
        int num;
        char sex;
        Date birthday;
        char ID[18];
public:
        People(){};//默认构造
        People(int n,int y,int m,int d,char id[18],char sex):birthday(y,m,d) {
        num=n;
        strcpy(ID,id);
};//有默认值的带参构造
People(People& p) //拷贝构造
{
name=p.name;
num=p.num;
```

```
sex=p.sex;

birthday=p.birthday;

strcpy(ID,p.ID);

}

void input() //输入函数

{

cout<<"Enter data:"<<endl;

cout<<"Name:";

cin>>name;

cout<<"Number:";

cin>>num;

cout<< "Sex(m/f) :";

cin>>sex;

cout<<"Birthday(Year/Month/Day):";

birthday.set();

cout<<"ID:";

cin>>ID;

cout<<endl;

};

void output() //输出函数

{

cout<<"Number:"<<num<<endl;
```

```
cout<<"Name:"<<name<<endl;

cout<<"Sex:"<<sex<<endl;

cout<<"Birthday:";

birthday.display();

cout<<endl;

cout<<"ID:"<<ID<<endl;

};

~People() //析构函数

{

cout<<num<<" No. Person has been entered."<<endl;

}

};

class student:public People

{

    char classno[7];

    public:student(){

        cout<<" "<<endl;

    }

void input()

{People::input();

cout<<"Enter Class Number"<<endl;

cin>>classno;
```

```
}

void getno(){

People::output();

cout<<"Class Number:"<<classno<<endl;

}

};

class teacher:public People

{

char pship[11],departt[21];

public:teacher(){

                cout<<" "<<endl;

        }

void input()

{

People::input();

cout<<"Enter Job Title"<<endl;

cin>> pship;

cout<<"Enter Department"<<endl;

cin>>departt;

}

void inputt()

{
```

```
cout<<"Enter Job Title"<<endl;

cin>> pship;

cout<<"Enter Department"<<endl;

cin>>departt;

}

void getno()

{

People::output();

cout<<"Job Title: "<<pship<<endl;

cout<<"Department: "<<departt<<endl;

}

void output()

{

cout<<"Job Title: "<<pship<<endl;

cout<<"Department: "<<departt<<endl;

}

};

class graduate:public student

{

char subject[21], adviser[21];

public:graduate(){

cout<<" "<<endl;
```

```
}

void input()

{student::input();

cout<<"Enter Major: "<<endl;

cin>>subject;

cout<<"Enter Adviser: "<<endl;

cin>>adviser;

}

void getno()

{student::getno();

cout<<"Major: "<<subject<<endl;

cout<<"Adviser: "<<adviser<<endl;

}

};

class TA:public graduate,teacher

{

public:TA(){

                cout<<" "<<endl;

        }

void input()

{

graduate::input();
```

```
teacher::inputt();
}
void getno()
{graduate::getno();
teacher::output();
}
};
int main()
{People p1;
student s;
teacher t;
graduate g;
TA T;
cout<<"Please enter the personnel data information in sequence." <<endl;
p1.input();
cout<<"Please enter student data information";
s.input();
cout<<"Please enter teacher data information";
t.input();
cout<<"Please enter graduate student data information";
g.input();
cout<<"Please enter TA data information";
```

```
T.input();

cout<<"Personnel data information: ";

p1.output();

cout<<"Student data information:";

s.getno();

cout<<"Teacher data information: ";

t.getno();

cout<<"graduate student data information: ";

g.getno();

cout<<"TA data information: ";

T.getno();

}
```