C++上机实验3

一、实验目的

1.了解继承在面向对象程序设计中的重要作用

2.进一步了解继承与派生的概念

3.掌握通过继承派生出一个新的类的方法

二、实验内容

1. 定义基类point，数据成员x,y(坐标)，构造函数，析构函数，输出x,y的函数get_point().
   定义公有派生类circle，数据成员：r(半径)，构造函数，析构函数，输出r的函数get_circle()。
   主函数定义circle对象c，输出x,y,r。

代码：

```
//problem 1
#include <iostream>
using namespace std;
// Base class:point
class point
{
private:
    float x;
    float y;
public:
    point();
```

```
    point(float a, float b);
    void get_point();
    ~point();
};
void point::get_point()
{
    cout << "(" << x << "," << y << ")"<< endl;
}
point::point()
{
    this->x = 0.0;
    this->y = 0.0;
}
point::point(const float a, const float b) : x(a) ,
y(b) {}
point::~point()
{
}

//Derived class circle
class circle : public point
{
private:
    float r;
public:
    circle(float a1,float b1,float c1);
    void get_circle();
    ~circle();
};
circle::circle(float a1 = 0.0 ,float b1 = 0.0 ,float
c1 = 0.0) : point(b1,c1)
{
    r = a1;
```

```
}
void circle::get_circle()
{
    cout << r << endl;
}
circle::~circle()
{
}
int main()
{
    float rr = 0.0;
    float xx = 0.0;
    float yy = 0.0;
    cout <<"Please input the point of circle:" <<
endl;
    cin >> xx >> yy;
    cout <<"Please input the r of circle:" << endl;
    cin >> rr;
    circle c(rr,xx,yy);
    cout << "The point of circle is:" << endl;
    c.get_point();
    cout <<"The r of circle is:" << endl;
    c.get_circle();
    return 0;
}
```

运行结果：

```
PS D:\Code\Cpp\Homework\week03> cd "d:\Code\Cpp\Homework\week03\" ; if ($?) { g++ point.cpp -o point } ; if ($?) { .\point }
Please input the point of circle:
1.1 2.2
Please input the r of circle:
1.2
The point of circle is:
(1.1,2.2)
The r of circle is:
1.2
```

2. 定义基类person，数据成员name,sex,age,构造函数，析构函数，输出name，sex，age的函数display()。
   定义公有派生类student，数据成员：num，构造函数，析构函数，输出name,sex,age,num的函数display()。
   主函数定义并使用student对象stu。

代码：

```
//problem 2
#include <iostream>
#include <string>
using namespace std;

//Base class : person
class person
{
private:
    string name;
    bool sex;
    int age;
public:
    person();
    person(const string name_,bool sex_,int age_);
    void display();
    ~person();
};
person::person()
{
}
person::person(const string name_,bool sex_,int
age_)
{
```

```
    name = name_;
    sex = sex_;
    age = age_;
}
void person::display()
{
    cout << "name: " << name << endl;
    if(sex == 1)
    {
        cout << "sex: " << "male" << endl;
    }
    else
    {
        cout << "sex: " << "femal" << endl;
    }
    cout << "age: " << age << endl;
}
person::~person()
{
}
// Derived class : student
class student : public person
{
private:
    string num;
public:
    student();
    student(const string name_,bool sex_,int
age_,const string num_);
    void display();
    ~student();
};
student::student()
```

```
{
}
student::student(const string name_,bool sex_,int
age_,const string num_) : person(name_,sex_,age_)
{
    num = num_;
}
void student::display()
{
    person::display();
    cout << "num: " << num << endl;
}
student::~student()
{
}
int main()
{
    student stu("shiwenwei",1,20,"20159100018");
    stu.display();
    return 0;
}
```

运行结果：

```
PS D:\Code\Cpp\Homework\week03> cd "d:\Code\Cpp\Homework\week03\" ; if ($?) { g++ person.cpp -o person } ; if ($?) { .\person }
name: shiwenwei
sex: male
age: 20
num: 20159100018
```

3. 分别声明Teacher(教师)类和Cadre(干部)类，采用多重继承方式由这两个类派生出新的类Teacher_Cadre类。要求：

在两个基类中都包含一部分相同名字的数据成员name,age,和成员函数display()。
在Teacher类中还包含数据成员title，在Cadre类中包含数据成员

post，在Teacher_Cadre中包含数据成员wages。
在派生类Teacher_Cadre的成员函数show中输出姓名、年龄、职称，职务与工资。
主函数定义Teacher_Cadre对象tc，输出其信息。

代码：

头文件：

```
//teachercadre.h
#ifndef TEACHERCADRE_H_
#define TEACHERCADRE_H_
#include <string>

//Teacher class
class Teacher
{
private:
    std::string name;
    int age;
    std::string title;
public:
    Teacher(/* args */);
    Teacher(const std::string name_,const int
age_,const std::string title_);
    void display();
    ~Teacher();
};
//Cadre class
class Cadre
{
private:
    std::string name;
```

```
    int age;
    std::string post;
public:
    Cadre();
    Cadre(const std::string name_,const int
age_,const std::string post_);
    void display();
    ~Cadre();
};

// Teacher_Cadre class
class Teacher_Cadre:public Teacher,public Cadre
{
private:
    float wages;
public:
    Teacher_Cadre();
    Teacher_Cadre(const std::string name_,
                  const int age_,
                  const std::string title_,
                  const std::string post_,
                  const float wages_);
    void show();
    ~Teacher_Cadre();
};
#endif
```

函数定义：

```
//teachercadre.cpp
#include <iostream>
#include "teachercadre.h"
```

```
//Teacher class
Teacher::Teacher(/* args */)
{
}
Teacher::Teacher(const std::string name_,const int
age_,const std::string title_)
{
    name = name_;
    age = age_;
    title = title_;
}
void Teacher::display()
{
    std::cout << "The name is: " << name <<
std::endl;
    std::cout << "The age is: " << age << std::endl;
    std::cout << "The title is: " << title <<
std::endl;
}
Teacher::~Teacher()
{
}
//Cadre class
Cadre::Cadre()
{
}
Cadre::Cadre(const std::string name_,const int
age_,const std::string post_)
{
    name = name_;
    age = age_;
    post = post_;
```

```
}
void Cadre::display()
{
    std::cout << "The post is: " << post <<
std::endl;
}
Cadre::~Cadre()
{
}
//Teacher_Cadre class
Teacher_Cadre::Teacher_Cadre()
{
}
Teacher_Cadre::Teacher_Cadre(const std::string
name_,
                    const int age_,
                    const std::string title_,
                    const std::string post_,
                    const float wages_) :
                    Teacher(name_,age_,title_),
                    Cadre(name_,age_,post_)
{
    wages = wages_;
}
void Teacher_Cadre::show()
{
    Teacher::display();//这里应该会重复输出吧
    Cadre::display();
    std::cout << "The wages is: " << wages <<
std::endl;
}
Teacher_Cadre::~Teacher_Cadre()
```

```
{
}
```

main函数：

```
//main.cpp
#include <iostream>
#include "teachercadre.h"
#include <string>
using namespace std;
int main()
{
    Teacher_Cadre tc("Wang",40,"Pro","aaa",10000.0);
    tc.show();
    return 0;
}
```

运行结果：

```
PS D:\Code\Cpp\Homework\week03> g++ .\teachercadre.cpp .\main.cpp -o main
PS D:\Code\Cpp\Homework\week03> .\main.exe
The name is: Wang
The age is: 40
The title is: Pro
The post is: aaa
The wages is: 10000
```

三、心得体会

通过本次的上机实验，我熟悉了类的继承与多重继承，编写了简单的程序进行实践。加深了对C++的理解。