实验五 继承（二）—— 基类对象与派生类对象

张林鹏_2021032449

一、实验目的

1. 理解公有继承、私有继承方式下, 类及对象的访问权限; 掌握公有继承方式下派生类对象访问基类私有成员、保护成员的方法; 了解私有继承方式下派生类对象访问基类私有成员、保护成员的方法.
2. .理解继承的传递性; 掌握公有继承方式下, 间接派生类对象访问间接基类私有成员、保护成员的方法; 了解私有继承方式下间接派生类对象访问间接基类私有成员、保护成员的方法.
3. 理解基类与派生类对象、指针、引用的兼容性规则, 熟悉派生类对象、指针、引用作为函数参数的各种方式.

二、实验内容

程序1: exp_501.cpp

1. 编译运行输出结果为:

```
x=10 y=20
x=10 y=20 z=30
x=10 y=20
```

2. 程序中的`b.get_x()`, `b.get_y()`表明在公有继承方式下, 派生类对象可以直接访问基类的公有成员变量和成员函数.

3. 将程序中的派生类定义由`public`改为`private`继承方式, 重新调试程序将出现编译错误, 原因是派生类对象不能直接访问基类的成员变量和成员函数.

4. 在派生类的公有成员定义 (即`public:`) 下面加上`Base::get_x;Base::get_y;`, 再重新编译运行程序, 将得到正确结果. `Base::get_x;Base::get_y;`成为使用基类作用域解析运算符来指定派生类的成员函数调用基类中的同名函数.

程序2: exp_502.cpp

5. 程序编译运行输出结果为:

```
x=10 y=20 z=30
```

6. 将`class B:public A`, `class C:public B`的继承方式改为`private`, 重新编译程序将会出现编译错误, 原因是派生类不能直接访问基类的成员变量和成员函数.

7. 在类`B`中增加公有成员函数`float get_x() {return A::get_x();}`,

   在类`C`中增加公有成员函数`float get_x() {return B::get_x();}`, `float get_y() {return B::get_y();}`,

   重新编译运行程序, 输出结果为:

```
x=10 y=20 z=30
```

程序3: exp_503.cpp

- 你分析的程序输出结果是:

```
x=25.5  y=35.5
x=25.5  y=35.5  z=50.5
x=25.5  y=35.5
x=25.5  y=35.5  z=50.5
x=25.5  y=35.5
```

- 程序实际运行输出结果是:

```
x=25.5  y=35.5
x=25.5  y=35.5  z=50.5
x=25.5  y=35.5
x=25.5  y=35.5  z=50.5
x=25.5  y=35.5
```

程序4: exp_504.cpp

- 你分析的程序输出结果是:

```
x=15  y=25
x=15  y=25
x=15  y=25
```

- 程序实际运行输出结果是:

```
x=15  y=25
x=15  y=25
x=15  y=25
```

程序5: exp_505.cpp

8. 修改对应编号下的代码
   i. 处应为: `pb->getx()`.
   ii. 处应为: `pb->gety()`.
   iii. 处应为: `dynamic_cast<Derived*>(pb)->getz()`.

程序设计:

- person.h:

```
#ifndef PERSON_H
#define PERSON_H

#include <bits/stdc++.h>
using namespace std;

class person
{
private:
    string name;
    string gender;
    int age;

public:
    person(string name = "", string gender = "", int age = 0);

    void setName(string name);
    void setGender(string name);
    void setAge(int age);

    string getName();
    string getGender();
    int getAge();

    void showInfo();
};

person::person(string name, string gender, int age)
{
    this->name = name;
    this->gender = gender;
```

```
    this->age = age;
}

void person::setName(string name)
{
    this->name = name;
}

void person::setGender(string gender)
{
    this->gender = gender;
}

void person::setAge(int age)
{
    this->age = age;
}

string person::getName()
{
    return name;
}

string person::getGender()
{
    return gender;
}

int person::getAge()
{
    return age;
}

void person::showInfo()
{
    cout << "Name: " << name << " Gender: " << gender << " Age: " << age << endl;
}

#endif
```

- teacher.h:

```
#ifndef TEACHER_H
#define TEACHER_H

#include "person.h"

class teacher : public person
{
private:
    string collage;
```

```
    string major;
    string education;
    string degree;
    string title;
    int teachingYears;

public:
    teacher(
            string name = "",
            string gender = "",
            int age = 0,
            string collage = "",
            string major = "",
            string education = "",
            string degree = "",              string title = "",
            int teachingYears = 0
            );
    void setCollage(string collage);
    void setMajor(string major);
    void setEducation(string education);
    void setDegree(string degree);
    void setTitle(string title);
    void setTeachingYears(int teachingYears);

    string getCollage();
    string getMajor();
    string getEducation();
    string getDegree();
    string getTitle();
    int getTeachingYears();

    void showInfo();
};

teacher::teacher(
    string name,
    string gender,
    int age,
    string collage,
    string major,
    string education,
    string degree,
    string title,
    int teachingYears)
{
    this->setName(name);
    this->setGender(gender);
    this->setAge(age);
    this->collage = collage;
    this->major = major;
    this->education = education;
    this->degree = degree;
    this->title = title;
```

```
    this->teachingYears = teachingYears;
}

void teacher::setCollage(string collage)
{
    this->collage = collage;
}

void teacher::setMajor(string major)
{
    this->major = major;
}

void teacher::setEducation(string education)
{
    this->education = education;
}

void teacher::setDegree(string degree)
{
    this->degree = degree;
}

void teacher::setTitle(string title)
{
    this->title = title;
}

void teacher::setTeachingYears(int teachingYears)
{
    this->teachingYears = teachingYears;
}

string teacher::getCollage()
{
    return collage;
}

string teacher::getMajor()
{
    return major;
}

string teacher::getEducation()
{
    return education;
}

string teacher::getDegree()
{
    return degree;
}
```

```
string teacher::getTitle()
{
    return title;
}

int teacher::getTeachingYears()
{
    return teachingYears;
}

void teacher::showInfo()
{
    cout << "Name: " << getName() << " Gender: " << getGender() << " Age: " <<
getAge() << endl;
    cout << "Collage: " << collage << " Major: " << major << " Education: " <<
education << " Degree: " << degree << " Title: " << title << " TeachingYears: " <<
teachingYears << endl;
}

#endif
```

- exp_506.h:

```
#include "teacher.h"
#include "person.h"
using namespace std;

int main()
{
    person p1("张三", "男", 20);
    teacher t1("李四", "女", 30, "计算机学院", "计算机科学与技术", "本科", "硕士", "教
授", 10);
    p1.showInfo();
    t1.showInfo();

    return 0;
}
```