实验四 继承（一）—— 派生类定义及访问权限

张林鹏_2021032449

一、实验目的

1. 理解继承的概念、特性及在C++语言中的实现方法;
2. 掌握C++语言派生类的定义, 熟悉不同的继承方式 (派生方式);
3. 掌握派生类构造函数的定义及在定义、释放派生类对象时构造函数、析构函数的执行顺序;
4. 掌握不同继承方式下, 基类的成员在派生类中的访问特性;
5. 初步熟悉派生类的应用.

二、实验内容

程序1: exp_401.cpp

1. 运行程序输出结果为:

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

2. 程序中`a.print( );`调用的是基类成员中的`print( )`成员函数, `b.print( );`调用的是Derived类成员中的`print( )`成员函数, `b.Base::print( );`调用的是类成员中的`print( )`成员函数.

3. 构造函数`Derived(float a=0,float b=0,float c=0):Base(a,b)`中的`Base(a,b)`的作用是:

   调用基类Base的构造函数,将a和b分别传递给基类构造函数中定义的x和y.

4. 将派生类定义中的`public`改为`private`, 重新编译程序, 程序中调用派生类函数语句会出现编译错误, 其原因是访问权限被限制, 无法访问私有成员.

程序2: exp_402.cpp

- 你认为的输出结果是:

```
基类构造函数被调用!
基类构造函数被调用!
派生类构造函数被调用!
x=10  y=20
x=10  y=20  z=30
派生类析构函数被调用!
基类析构函数被调用!
基类析构函数被调用!
```

- 程序运行输出结果是:

```
基类构造函数被调用!
基类构造函数被调用!
派生类构造函数被调用!
x=10  y=20
x=10  y=20  z=30
派生类析构函数被调用!
基类析构函数被调用!
基类析构函数被调用!
```

程序3: exp_403.cpp

- 你认为的输出结果是:

```
A 类的构造函数被调用!
Data 类的构造函数被调用!
B 类的构造函数被调用!
B 类的析构函数被调用!
Data 类的析构函数被调用!
A 类的析构函数被调用!
```

- 程序运行输出结果是:

```
A 类的构造函数被调用!
Data 类的构造函数被调用!
B 类的构造函数被调用!
B 类的析构函数被调用!
Data 类的析构函数被调用!
A 类的析构函数被调用!
```

程序4: exp_404.cpp

5. 程序中:
    i. 处应改为: `float a=0,float b=0,float c=0`

ii. 处应改为: `Base(a,b)`

iii. 处应改为: `float a=0,float b=0,float c=0`

iv. 处应改为: `setBase(a,b);`

程序设计实验:

- point.h:

```
#ifndef _POINT_H
#define _POINT_H

#include <bits/stdc++.h>
#define M_PI acos(-1)

class point
{
public:
    double x, y;
    point(double x, double y) : x(x), y(y) {}
    point() {}
    friend class distance;
};

class circle : public point
{
public:
    double r;
    circle(double x, double y, double r) : point(x, y), r(r) {}
    circle() {}

    void inputCircle();
    void outputCircle();
};

void circle::inputCircle()
{
    std::cout << "Input x(cm): ";
    std::cin >> x;
    std::cout << "Input y(cm): ";     std::cin >> y;
    std::cout << "Input r(cm): ";
    std::cin >> r;
}

void circle::outputCircle()
{
    std::cout << "Circumference: " << 2 * M_PI * r << "cm" << std::endl;
    std::cout << "Area: " << M_PI * r * r << "cm^2" << std::endl; }

#endif
```

- exp_405.cpp:

```cpp
#include "point.h"

int main()
{
    circle c;
    c.inputCircle();
    c.outputCircle();

    return 0;
}
```