

C++程序设计上机报告：继承和多文件

姓名	周新斌	学号	2118140201
班级	软工 212		
目的及要求	(1) 掌握继承的语法 (2) 能够编写包含继承关系的程序 (3) 掌握基类中的不同访问控制的数据和函数在派生类中的访问情况。 (4) 掌握派生来的对象创建时构造和析构函数调用次序 (5) 掌握做多文件编程的操作方法		
上机学时	2 学时		
设备要求	(1) 主要仪器设备：微型计算机 (2) 软件环境：WINDOWS 7 10 11 等操作系统；Dev C++、Visual C++等程序编译环境		
上机内容	建立一个建筑类的类体系。 (1) 其中基类为 building，包括保护类型的 bname、bfloors 和 bareas。 (2) 由 building 类公有派生出 house 类和 office 类；house 包括私有的数据成员 rooms 和 balcony；office 包括私有的 offices 和 meetingrooms。 (3) 基类 building、派生类 house 和 office 中都包括构造函数和 print 函数，print 函数用于输出各个类的数据成员 (4) 定义一个 main 函数对类中的所有的内容进行测试。 (5) 定义三个.h 文件分别用来声明三个类，定义三个.cpp 文件用来实现三个类中的函数，最后定义一个 cpp 文件用来编写 main 函数，在 main 函数的.cpp 文件中进行编译和运行。所有的文件都放在一个项目里。 (6) 所有的源代码按照文件顺序粘贴到报告中。		

源代码

main.cpp

```
#include "building.h"
#include "house.h"
#include "office.h"
#include <iostream>

/* run this program using the console pauser or add your own getch,
system("pause") or input loop */

int main(int argc, char** argv) {
    building b("building", 1, 1.1);
    b.print();
    puts("");
    house h("house", 2, 2.2, 222, true);
    h.print();
    puts("");
    office o("office", 3, 3.3, 333, 3333);
    o.print();
    return 0;
}
```

building.h

```
#ifndef BUILDING_H
#define BUILDING_H

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

class building
{
    protected:
        string bname;
        int bfloors;
        double bareas;
    public:
        // string 传引用 只读
        building(const std::string& name, int floor, double area);
        // const 函数保证数据不被修改, virtual 保证多态性
        virtual void print() const;
};
#endif
```

building.cpp

```
#include "building.h"
using namespace std;

building::building(const std::string& name, int floor, double area)
    : bname(name), bfloors(floor), bareas(area) {}

void building::print() const
{
    cout << "bname: " << bname << ' ' << "bfloors: " << bfloors
        << ' ' << "bareas: " << bareas << endl;
}
```

house.h

```
#ifndef HOUSE_H
#define HOUSE_H

#include "building.h"
using namespace std;

class house : public building
{
    int rooms;
    bool balcony;
public:
    house(const std::string& name, int floors, double areas, int
rooms, bool balcony);
    void print() const override;
};

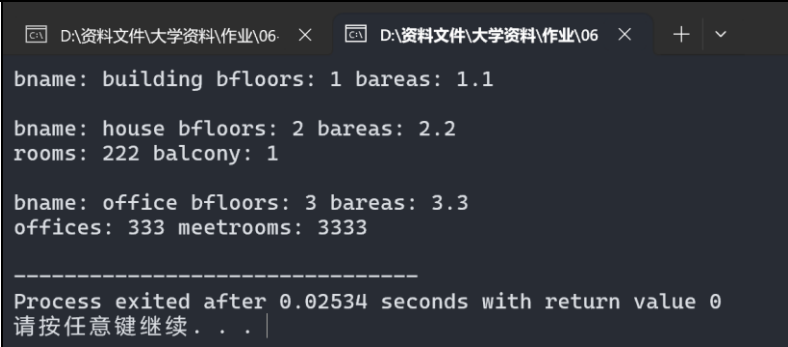
#endif
```

house.cpp

```
#include "house.h"
using namespace std;

house::house(const std::string& name, int floors, double areas, int rooms,
bool balcony)
    : building(name, floors, areas), rooms(rooms), balcony(balcony) {}

void house::print() const
{
    building::print();
    cout << "rooms: " << rooms << ' ' << "balcony: " << balcony <<
endl;
}
```

	<div>office.h</div> <pre>#ifndef OFFICE_H #define OFFICE_H #include "building.h" using namespace std; class office : public building { int offices; int meetingrooms; public: office(const std::string& name, int floor, double area, int office, int meetingroom); void print() const override; }; #endif</pre> <div>office.cpp</div> <pre>#include "office.h" using namespace std; office::office(const std::string& name, int floor, double area, int office, int meetingroom) : building(name, floor, area), offices(office), meetingrooms(meetingroom) {} void office::print() const { building::print(); cout << "offices: " << offices << ' ' << "meetrooms: " << meetingrooms << endl; }</pre>
程序的输出结果	 <pre>D:\资料文件\大学资料\作业\06 × D:\资料文件\大学资料\作业\06 × + v bname: building bfloors: 1 bareas: 1.1 bname: house bfloors: 2 bareas: 2.2 rooms: 222 balcony: 1 bname: office bfloors: 3 bareas: 3.3 offices: 333 meetrooms: 3333 ----- Process exited after 0.02534 seconds with return value 0 请按任意键继续. . . </pre>

<p>程序难点分析</p>	<p>(1) house 类和 office 类的对象分别占有多少个字节，怎么计算出来的？</p> <p>int = 4 Bytes</p> <p>double = 8 Bytes</p> <p>bool = 1 Bytes</p> <p>sizeof(building) = 32 Bytes</p> <p>所以：</p> <p>house = 32 + 4 + 1 = 37 Bytes</p> <p>(house 的 bool 内存对齐情况下 40Bytes)</p> <p>office = 32 + 4 + 4 = 40 Bytes</p> <p>(2) 定义 house 类的对象按照什么顺序调用构造和析构函数？</p> <p>构造顺序：building, house</p> <p>析构顺序：house, building</p> <p>(3) bname、bfloors 和 bareas 在 house 类和 office 类中是否能够访问，为什么？</p> <p>能够被访问，protected 保护类型的数据成员，能在基类以及所有子类访问，但不能被其他类访问。</p>
---------------	---