

构造/析构函数在继承中注意事项

2015年11月29日 星期日 上午11:47

一、简介

构造函数、析构函数是类最基本的函数，负责初始化对象、释放对象。在派生类对基类继承的过程中，有一些需要注意的地方。

- A 构造函数没有虚函数，派生类会调用基类的构造函数。
- B 有纯虚函数的基类，不能直接调用基类构造函数。
- C 可以不显式声明、实现构造/析构函数，这样编译器会生成默认的
- D 如果显式声明，必须显式实现
- E 基类显式声明、实现，子类按照C中使用默认，这样是可以的
- F 基类不显式声明、实现，子类显式声明、实现，这样也是可以的
- G 派生类之间如果有继承关系，需要用虚继承

下面是详细介绍和实验

二、详细介绍

1、构造函数

1.1基类与派生类中构造函数

构造函数没有虚函数，不能写virtual，但是派生类可以定义自己的构造函数，派生类的构造函数中，除了完成显式写出的功能，还会默认调用基类的构造函数。

1.2构造函数的调用

基类的对象声明，使用子类的构造函数进行new

如果基类中有纯虚函数的话，直接调用基类的构造函数Base()进行声明是不行的，只能让基类声明一个指针，通过new调用子类的构造函数。

```
// Cannot instantiate, Base is abstract
Base base1;

// Should work
Example example;

// Should work
Base* base2 = new Example();
```

详见参考资料1

我在程序中，遇到的问题是声明了一个vector<Base>

2、析构函数

析构函数有虚函数，可以写virtual，如：

```
class A
{
public:
A();
virtual ~A();
void f1();
virtual void f2();
```

如果类中有虚函数，则析构函数也必须虚。不然会报warning: `class A' has virtual functions but non-virtual destructor。

3、构造/析构函数在基类与派生类中

如果你想重构构造函数/析构函数，需要在函数声明的时候进行声明然后实现

几种类型、现象及原因：

A 只实现不声明

会报错error: definition of implicitly-declared `A::A()', 可能是会和系统自动的构造/析构函数冲突。

B 只声明不实现，这样会在链接阶段报错和通常的函数一样

C 基类显式声明、实现构造/析构函数，但是不是以虚函数的形式，子类不显式声明、实现

会报warning，但是运行看起来没问题。

warning: `class goodcoder::IntType' has virtual functions but non-virtual destructor

可能是说，这个子类使用了虚函数，但是没有虚析构函数，这样可能存在这一些隐患，比如虚函数使得子类 and 基类实现上有所差异，但是析构的时候是没有差异的(因为没有使用需基函数)。

D 基类不显式声明、不显式实现构造/析构(相当于调用默认构造/析构函数)，子类声明、实现。

正常，可以编译运行，调用子类显式实现的构造/析构函数，子类的构造/析构函数调用基类的，基类是默认的。

E 基类和子类都显式声明、实现构造/析构函数，基类析构是虚函数(构造不能是虚函数)。

正常都是使用显式实现出的函数。

F 基类显式声明、实现，析构是虚函数(构造不能是虚函数)，子类不显式

声明、头文件。

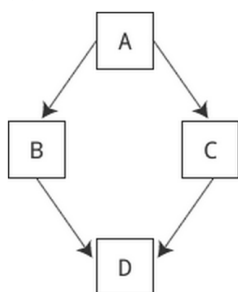
子类调用的是默认构造/析构函数，其中默认的构造/析构函数会调用基类的实现。

G 基类只声明不实现 构造/析构函数，基类析构是虚函数(构造不能是虚函数)。

子类构造函数编译过程中会报链接错误。因为子类的构造/析构函数，会调用基类的构造/析构函数，这样写的话，相当于基类只声明了"要显式的定义构造/析构函数"，但是并没有实现，导致子类找不到实现，链接错误。

4、派生类之间有层级关系的情况

各个派生类之间有层级的继承关系时，使用虚继承，使得基类的构造函数不会被重复调用



有层级的继承关系

详见参考资料，由于不常用，不再详述。

三、实验

实验地址：

work@cp01-qa2014-junheng2plus103.cp01.baidu.com:
/home/work/tianzhiliang/test/goodcoder/branch/code_test_class

goodcoder代码地址

<http://cooder.baidu.com/2321467>

四、参考资料

- 1、<http://stackoverflow.com/questions/23475470/pure-virtual-function-and-cannot-allocate-an-object>
- 2、虚基类 <http://blog.csdn.net/lovemysea/article/details/5298853>
- 3、虚基类 <http://c.biancheng.net/cpp/biancheng/view/238.html>