

C++语言基础

迂者 - 贺利坚

<http://blog.csdn.net/sxhelijian/>

<http://edu.csdn.net>





本节主题：

虚析构函数

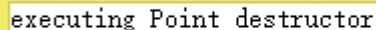
问题的由来

- 当派生类的对象从内存中撤销时
 - 一般先调用派生类的析构函数
 - 然后再调用基类的析构函数。
- 问题由此而来.....

```
class Point{
public:
    Point( ) {}
    ~Point() {
        cout<<"executing Point destructor"<<endl;
    }
};
```

```
class Circle:public Point
{
public:
    Circle( ) {}
    ~Circle( )
    {
        cout<<"executing Circle destructor"<<endl;
    }
private:
    int radius;
};
```

```
int main( )
{
    Point *p=new Circle;
    delete p;
    return 0;
}
```



D:\CPP\code... - [X]

executing Point destructor

- 如果用new运算符建立了派生类对象，并且由一个基类的指针变量指向该对象。
- 用delete运算符撤销对象时，系统只执行基类的析构函数，而不执行派生类的析构函数——派生类对象析构中要求的工作将被忽略！

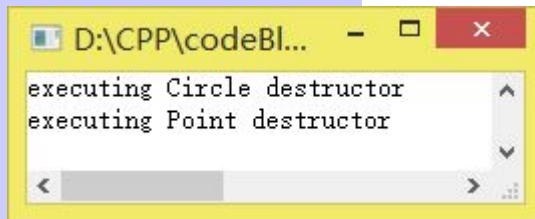
使用虚析构函数

- 当派生类的对象从内存中撤销时
 - 一般先调用派生类的析构函数
 - 然后再调用基类的析构函数。
- 这样做问题没有了.....

```
class Point{
public:
    Point() {}
    virtual ~Point() {
        cout<<"executing Point destructor"<<endl;
    }
};
```

如果将基类的析构函数声明为虚函数时，由该基类所派生的所有派生类的析构函数也都自动成为虚函数，即使派生类的析构函数与基类的析构函数名字不相同。

```
class Circle:public Point
{
public:
    Circle() {}
    ~Circle()
    {
        cout<<"executing Circle destructor"<<endl;
    }
private:
    int radius;
};
```

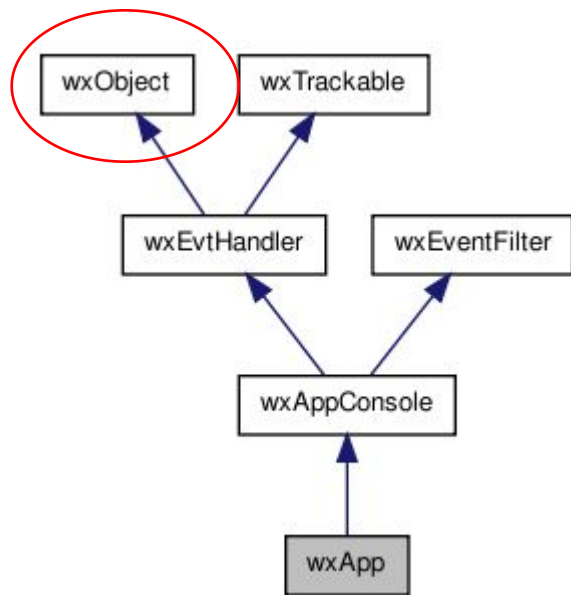


```
int main( )
{
    Point *p=new Circle;
    delete p;
    return 0;
}
```

虚析构造函数用法

- ❏ 当基类的析构造函数为虚函数时，无论指针指的是同一类族中的哪一个类对象，系统会采用动态关联，调用相应的析构造函数，对该对象进行清理工作
 - 📁 先调用了派生类的析构造函数，再调用了基类的析构造函数，符合人们的愿望
- ❏ 最好把基类的析构造函数声明为虚函数
 - 📁 这将使所有派生类的析构造函数自动成为虚函数
- ❏ 虚析构造函数在面向对象程序设计中是很重要的技巧
 - 📁 专业人员一般都习惯声明虚析构造函数，即使基类并不需要析构造函数，也显式地定义一个函数体为空的虚析构造函数，以保证撤销动态分配空间时能正确的处理
- ❏ 构造函数不能声明为虚函数。

实例：wxWidgets 中的顶层基类 wxObject 的声明



Public Member Functions

```
wxObject ()  
Default ctor; initializes to NULL the inter:  
  
wxObject (const wxObject &other)  
Copy ctor.  
  
virtual ~wxObject ()  
Destructor.  
  
virtual wxClassInfo * GetClassInfo () const  
This virtual function is redefined for every  
  
wxObjectRefData * GetRefData () const  
Returns the wxObject::m_refData pointer,  
  
bool IsKindOf (const wxClassInfo *info) const  
Determines whether this class is a subclass  
  
bool IsSameAs (const wxObject &obi) const
```

THANKS

本课程由 迂者-贺利坚 提供

CSDN网站：www.csdn.net
企业服务：<http://ems.csdn.net/>
人才服务：<http://job.csdn.net/>
CTO俱乐部：<http://cto.csdn.net/>
高校俱乐部：<http://student.csdn.net/>
程序员杂志：<http://programmer.csdn.net/>

CODE平台：<https://code.csdn.net/>
项目外包：<http://www.csto.com/>
CSDN博客：<http://blog.csdn.net/>
CSDN论坛：<http://bbs.csdn.net/>
CSDN下载：<http://download.csdn.net/>

