



VIP C++ 直播答疑课

潘祖江
清华大学计算机系

类和对象

面向对象基本特点-知识点综述

- 面向对象方法中的抽象，是指对具体问题（对象）进行概括，抽出一类对象的公共性质并加以描述的过程。
- 封装就是将抽象得到的数据和行为（或功能）相结合，形成一个的有机整体，也就是将数据与操作数据的函数代码进行有机地结合，形成“类”，其中的数据和函数都是类的成员。
- C++语言中提供了类的继承机制，允许程序员在保持原有类特性的基础上，进行更具具体、更详细的说明。
- 多态性是指一段程序能够处理多种类型对象的能力。在C++语言中，这种多态性可以通过强制多态、重载多态、类型参数化多态、包含多态4种形式来实现。

在C++语言中类和结构体类型有何异同？

- C++增加了class类型后，仍保留了结构体类型struct，而且把它的功能进行了扩展。结构体类型中同样可以包括成员函数。C++中用关键字class声明的类类型可以改为用关键字struct。C++语言完全兼容C，但是C++不是简单地继承C的结构体，而是使它也具有类的特点，以便于用于面向对象程序设计。用struct声明的结构体类型实际上也就是类。
- 用struct声明的类和class声明的类是有区别的。用struct声明的类，如果对其成员不作private或public的声明，系统将其默认为public(公用的)。如果想分别指定私有成员和公用成员，则应用private或public作显式声明。而用class定义的类，如果不作private或public声明，系统将其成员默认为private(私有的)，在需要时也可以自己用显式声明改变访问属性。建议尽量使用class来建立类，写出完全体现C++风格的程序。

知识点综述-类和对象

在面向对象程序设计中，程序模块是由类构成的。类是对逻辑上相关的函数与数据的封装，它是对问题的抽象描述。

声明类的语法形式如下：

```
class 类名称
{
    public:
        外部接口
    protected:
        保护型成员
    private:
        私有成员
}
```

知识点综述-类和对象 *CON'D*

与普通函数不同的是，实现成员函数时要指明类的名称，具体形式为：

返回值类型 类名::函数成员名(参数表)

{ 函数体 }

声明一个对象，和声明一个一般变量相同，采用以下方式：

类名 对象名；

声明了类及其对象，我们就可以访问对象的公有成员。这种访问采用的是“.”操作符，其一般形式是：

对象名.公有成员函数名(参数表)

示例

编写设计全部成员为`public`模式的类

编写设计`private`数据访问模式的类

编写结构体类——以结构体形式定义类

构造函数和析构函数-知识点综述

- 构造函数的作用就是在对象被创建时将对象初始化为一个特定的状态。构造函数的函数名与类名相同，而且没有返回值。构造函数在对象被创建的时候将被自动调用。如果类中没有写构造函数，编译器会自动生成一个默认形式的构造函数——没有参数，也不做任何事情。
- 拷贝构造函数是一种特殊的构造函数，其形参是本类的对象的引用，其作用是使用一个已经存在的对象，去初始化同类的一个新对象。如果没有定义类的拷贝构造函数，系统就会在必要时自动生成一个默认的拷贝构造函数，把初始值对象的每个数据成员的值都复制到新建立的对象中。

CON'D

- 声明和实现拷贝构造函数的一般方法：

```
class 类名
{public:
    类名(形参表);                //构造函数
    类名(类名 &对象名);          //拷贝构造函数
    ...
};
类名::类名(类名 & 对象名);      //拷贝构造函数的实现
{    函数体}
```

- 析构函数用来完成对象被删除前的一些清理工作。析构函数是在对象的生存期即将结束的时刻被自动调用的，可以是虚函数。它的名字是类名的前面加一个“~”符号，不返回任何值，没有函数类型，也没有函数参数。

什么时候程序会自动调用析构函数？- 疑难解析

当对象的生存期结束时，会自动执行析构函数。具体情况有：

①函数执行结束时，*auto*局部对象应该被释放，此时自动执行析构函数。

②在程序结束（*main*函数结束或执行*exit*函数）时，调用*static*局部对象的析构函数和全局对象的析构函数。

④如果用*new*运算动态地建立了一个对象，当用*delete*运算释放该对象时，调用该对象的析构函数。

什么时候程序会自动调用拷贝构造函数？-疑难解析

- ① 当用类的一个对象去初始化该类的另一个对象时。
- ② 如果函数的形参是类的对象，调用函数时，进行形参和实参结合时。
- ③ 如果函数的返回值是类的对象，函数执行完成返回调用者时。

示例

编写有理数类——使用初始化列表的构造函数

完善有理数类——拷贝构造函数的调用时机

完善有理数类——析构函数

类的组合-知识点综述

- 类的组合描述的就是一个类内嵌其它类的对象作为成员的情况，它们之间的关系是一种部分与整体的关系。

当创建类的对象时，如果这个类具有对象成员，那么各个对象成员将首先被自动创建。

- 在创建一个组合类的对象时，不仅会调用它自身的构造函数，而且还将调用其对象成员的构造函数。这时构造函数的调用顺序是：① 调用对象成员的构造函数，调用顺序按照对象在组合类的声明中出现的次序② 执行本类构造函数的函数体。析构函数的调用执行顺序与构造函数刚好相反。

组合类构造函数定义的语法形式为：

类名::类名(形参表):内嵌对象1(形参表), 内嵌对象2(形参表), ...

{ 类的初始化 }

- 如果需要在类声明之前引用该类名，则需要进行前向引用声明。

示例

计算三角形周长——一类组合的应用

C++语言存储与访问属性

作用域与可见性-知识点综述

- 作用域是一个标识符在程序正文中有效的区域。C++中标识符的作用域有函数原型作用域、块作用域、类作用域和文件作用域。
- 从标识符引用的角度，来看变量的有效范围，即标识符的可见性。程序运行到某一点，能够引用到的标识符，就是该处可见的标识符。
- 作用域可见性的一般规则是：标识符要声明在前，引用在后。在同一作用域中，除了重载的函数名以外，不能声明同名的标识符。在没有互相包含关系的不同的作用域中声明的同名标识符，互不影响。如果在两个或多个具有包含关系的作用域中声明了同名标识符，则外层标识符在内层不可见。

为什么应避免使用不必要的全局变量？-疑难解析

全局变量存储在由程序预留的一个固定内存区内。在整个执行过程中，它们都会占用内存，而不只是在需要时才占用内存；在可以使用局部变量的地方使用全局变量会降低函数的通用性，由于这样的函数依赖于在其外部定义的变量；太多的全局变量会带来未知的、预料不到的副作用。

示例

有理分数的输出——一类作用域变量

静态生存期与动态生存期-知识点综述

- 对象都有诞生和结束的时刻。对象从诞生到结束的这段时间就是它的生存期。在生存期内，对象将保持其状态（即数据成员的值），直到它们被更新为止。
- 如果对象的生存期与程序的运行期相同，称它具有静态生存期。在文件作用域中声明的对象都是具有静态生存期的。如果要在函数内部的块作用域中声明具有静态生存期的对象，则要使用关键字***static***。
- 在块作用域中声明的具有动态生存期的对象，习惯上称为局部生存期对象。动态生存期对象诞生于声明点，结束于该标识符作用域结束处。

示例

求数据序列的平均值——*static*局部变量的持续效果

求数据序列的平均值——*static*全局变量的应用

静态数据成员和函数成员-知识点综述

- 静态成员用于解决同一个类的不同对象之间数据和函数共享问题。
- 静态成员是“类属性”的概念。如果某个属性为整个类所共有，不属于任何一个具体对象，则采用***static***关键字来声明为静态成员。
- 静态数据成员具有静态生存期。在类的声明中仅仅对静态数据成员进行引用性声明，必须在文件作用域的某个地方使用类名限定进行定义性说明，这时也可以进行初始化。
- 所谓静态成员函数就是使用***static***关键字声明的函数成员，同静态数据成员一样，静态成员函数也属于整个类，由同一个类的所有对象共享。
- 公有的静态成员，可以通过类名或对象名来访问。
- 静态成员函数可以直接访问该类的静态数据和函数成员。而访问非静态数据成员，需要通过参数传递方式得到对象名，然后通过对象名来访问。



示例

实现数据共享-公有静态数据成员

实现数据共享-私有型静态数据成员

友元函数和友元类-知识点综述

- 友元关系提供了不同类或对象的成员函数之间、类的成员函数与一般函数之间进行数据共享的机制。
- 在一个类中，可以利用关键字 *friend* 将其它函数或类声明为友元。如果友元是一般函数或类的成员函数，称为友元函数；如果友元是一个类，则称为友元类，友元类的所有成员函数都自动成为友元函数。
- 在友元函数中可以通过对象名访问类的私有和保护成员。

疑难解析

1、友元关系与访问属性是否相关？

友元仍然不是类的成员，*private*、*protected*和*public*与友元关系的声明无关，因此友元关系声明可以放在类声明中的任何地方。

2、使用友元关系应注意什么？

第一，友元关系是不能传递的，*B*类是*A*类的友元，*C*类是*B*类的友元，*C*类和*A*类之间，如果没有声明，就没有任何友元关系。第二，友元关系是单向的，如果只声明*B*类是*A*类的友元，*A*类不会自动成为*B*类的友元。第三，友元关系是不被继承的，如果类*B*是类*A*的友元，类*B*的派生类并不会自动成为类*A*的友元。

示例

判断圆柱体和立方体的颜色是否相同 - 多个类共享友元函数

常量数据与共享数据的保护-知识点综述

- 如果在声明引用时用`const`修饰，被声明的就是常引用。不能通过常引更新用所引用的对象。如果用常引用做形参，便不会意外地发生对实参的更改。常引用声明形式如下：

`const` 类型说明符 &引用名

- 常对象的数据成员值在对象的整个生存期间内不能被改变。因此常对象必须进行初始化，而且不能被更新。声明常对象的语法形式为：

类名 `const` 对象名

- 使用`const`关键字修饰的函数为常成员函数，常成员函数声明的格式如下：

类型说明符 函数名（参数表）`const`;

- 就像一般数据一样，类的成员数据也可以是常量和常引用，使用`const`说明的数据成员为常数据成员。如果在一个类中说明了常数据成员，那么任何函数中都不能对该成员赋值。构造函数对该数据成员进行初始化，就只能通过初始化列表。

