

C++ 多态性与虚函数概要

1. 多态性与虚函数

多态性：发生在一个或多个继承于某基类的子类与基类本身之间的一种函数重写关系，多态性一般要通过在类中声明一个虚函数(关键字 `virtual`)来实现。

使用说明：首先在基类中定义一个虚函数，函数体可以为空也可以不为空，再次先说不为空的，假如定义该虚函数是输出一个字符串“`abcdefg`”，当该基类派生出一个子类，此时子类也会自动把基类的虚函数继承下来（当然可以把子类的该函数加上 `virtual` 关键字也可以不加）。OK，现在我们可以重写改被继承下来的函数了，假如我们重写为输出一个字符串“`1234567`”。好了，到此为止我们就完成了多态性的准备工作，接下来还可以通过基类派生出更多的子类，并对对应的函数进行重写，这就要看具体需要来写了。

2. 静态关联和动态关联

下面进一步说明多态性是怎样实现的。假设上面的工作我们都准备好了，此时我们才可以写测试函数 `main()`，通常同学朋友们会这样做，直接就声明一个基类对象，然后通过这个对象就可以引用类内成员；然后再定义一个子类对象，再调用类内成员。这样做直接提供对象名，即通过对象名来调用虚函数，系统在编译时能够确定调用的是哪个类中的函数，这样的过程叫做静态关联，由于在运行前进行关联的，所以又称早期关联。不过同学朋友们别忘记了，还有一种做法就是：定义一个基类指针，并使它指向类对象，然后再使它指向子类对象，再一一调用各自的虚函数。显然，这种调用方式，编译系统是无法确定应该调用那一个类的虚函数，这样要确定的话，就只能放在运行阶段确定关联关系了，这种关联就叫做动态关联，也称滞后关联。

3. 什么时候应当声明虚函数

在声明虚函数之前要注意两点：

- (1) 只能用 `virtual` 声明类中的成员函数，使它成为虚函数，不能将类外的普通函数声明为虚函数。因为虚函数的作用是允许在派生类中对基类的虚函数进行重写。所以说，它只能用于类的继承层次结构中。
- (2) 一个成员函数被声明为虚函数后，在同一个类中不能再定义一个非 `virtual` 的但与该虚函数具有相同的参数和返回值的同名函数。

下面来说说把一个成员函数声明为虚函数要考虑什么：

- (1) 首先应当看成员函数所在的类是否作为基类，然后看成员函数在类的继承后有无可能被更改功能，如果有，一般都应将其声明为虚函数。
- (2) 如果你是通过声明一个基类指针或引用去访问成员函数的，那就应该声明为虚函数。
- (3) 有时，在定义虚函数时，可以不定义其函数体，即函数体是空的。它的作用只是定义了一个虚函数名，具体作用留给派生类去添加，这就是前面说的虚函数体可以为空也可以不为空，看具体情况。

*4. 虚析构函数

首先说明析构函数的作用是什么？析构函数的作用就是在对象撤销之前做必要的“清理现场”的工作。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。现在将说明一种情况，如果用 `new` 运算符建立了临时对象，若积累中有析构函数，并且定义了一个指针指向基类的指针变量，那在程序用带指针的参数调用 `delete` 运算符撤销对象时，会发生一个情况：系统会只执行基类的析构函数，而不执行派生类的析构函数。大家都知道执行了派生类的构造函数之后很有必要撤销掉该对象，然而上述的方式并没有做到撤销。那该如何解决这个问题呢？很简单，只需要把基类的

析构函数声明为虚函数就 OK 了，由该基类继承下来的派生类（所有）的析构函数也自动成为虚函数（即使派生类的析构函数和基类的析构函数名字不同）。当对象撤销时，系统会采用动态关联，调用相应的析构函数，对该对象进行清理工作。

5. 纯虚函数与抽象类

有时候将一成员函数声明为虚函数，并不是基类本身的要求，而是考虑到派生类的需要，这样在基类中预留一个函数名，具体功能留给派生类根据需要去定义。下面就说说可以怎样做。

假如我们已经定义好了一个基类，声明了一个成员函数为虚函数，可以这么写：`virtual type function () const {return 0;}` 其返回值是 0，其实很明显就是一个没有意义的函数，在基类中不使用该函数，为了简化，可以不写出函数体，只需给出函数原型即可，并在后面加上“=0”，例如 `virtual type function () const = 0;` 这样的函数成为纯虚函数。当然后面的 0 不表示返回值，只是一个表示而已。不过需要注意：纯虚函数只是一个函数名而已，不具备被调用的性质。唯有在派生类中被重写之后，才具有被调用的功能。如果在子类中没有重写这个纯虚函数，那么该虚函数在子类中仍然为纯虚函数。

6. 抽象类

大家都知道，如果声明了一个类，那么我们就可以好容易地定义一个对象，从而去引用类内的各个成员。但是在这里我要介绍有一种类是不能用来生成对象的，对！它就是抽象类，定义这样的类的作用是以它为基类去建立派生类。它只作为一种基本类型提供给用户，在这基础上用户能够根据自己的需要去定义出各个功能的派生类，进而再用这些派生类去建立对象。

Last:

哈哈!! 这次先说到这里。之前的 C 语言笔记只是为了应付考试用的，可是对于 C++ 的说明，我都会从实际应用去解释，让朋友们能够体会其中的作用。如果你要当作是考试来看的，那应该没问题!! 只要大家记住两个字就行：实践!! 要理解计算机语言光看不行的，要多动手，别为了考试而去看。（当然是对热爱编程的同学朋友说的）! 下次该写什么呢？嘿嘿!! 暂时还没想到。。(*^__^*)...嘻嘻。。。

Ps:临近考试!! 祝大家考试顺利!!

Made By Rock.Z

ROZ Studio

6.18.2009

