

多重继承与虚继承

- 1- 多重继承的 构造可显示，可隐式调用：
- `C():A(),B(){} 这样就是显示的`
- `C():B(){} 这样就是隐式的调用A构造函数，显示调用B的构造函数了。`

2- 多态

- 运行时多态就是用父类的指针指向子类的对象，调用子类中的函数。在运行时决定调用哪一个函数。
- 运行时多态是与静态多态不同的，静态多态就是函数的重载和模板。

3- 虚继承： 解决菱形继承问题

- 虚继承的类就叫做虚基类
- 虚继承的类就叫做虚基类。 A就是是虚基类， B和C虚继承A， D中只有一个A对象。

```
class A{
};
class B:public virtual A{

};
class C:virtual public A{//virtual与public的顺序都可以的

};
class D:public B, public C{

};
```

4- 虚函数表： 运行时多态

C++中的多重继承和虚继承是一个非常重要的概念，也是看你是不是懂C++的一个重要的标志之一。这中间包括了运行时多态，虚函数表等等相关概念。

- 多重继承，顾名思义，是一个类继承了多个父类。例如
- `class C: public A, public B{};`这样就是一个多重继承。

多重继承的构造和析构函数。

- 多重继承的构造函数是先执行父类中的构造函数在执行自己的构造函数，多个父类按照继承顺序来。上面的例子就是先A再B最后C。析构正好相反。
- 可以显示的在初始化列表中进行构造，也可以不写隐式调用父类的构造函数。例如

`C():A(),B(){}` 这样就是显示的。`C():B(){}` 这样就是隐式的调用A构造函数，显示调用B的构造函数了。

运行时多态

- 运行时多态就是用父类的指针指向子类的对象，调用子类中的函数。在运行时决定调用哪一个函数。
- 运行时多态是与静态多态不同的，静态多态就是函数的重载和模板。

虚继承

- 虚继承是解决多重继承中的一个非常常见的问题：菱形继承问题。即基类是A，B和C分别继承A，D多重继承B和C。D在构造函数的时候就执行了两个A的构造函数。所以会出现错误的。这个时候就用到了虚继承。
- 虚继承的用法是这样的：**虚继承的类就叫做虚基类**。A就是是虚基类，B和C虚继承A，D中只有一个A对象。

```
class A{
};
class B:public virtual A{

};
class C:virtual public A{//virtual与public的顺序都可以的

};
class D:public B, public C{

};
```

- 最后注意，**虚基类总是先于非虚基类构造的**，与继承的顺序是没有关系的。

虚函数表

- 还要说的就是**虚函数表V-Table**的问题。**虚函数表是运行时多态的根本上的保障**。**虚函数是一个类如果它包含虚函数**，则该类就会在内存中建立这样一个虚函数表。在运行时，根据虚函数表决定调用哪一个函数。
- 一个类如果包含虚函数，则它的地址中有4位是虚函数表地址的指针。虚函数表示一块连续的内存。为每一个父类（有虚函数）都会有一个虚函数表。多重继承中，子类的函数放在第一个虚函数表内。详情请看这个链接，他这个讲的就非常全面。
- <http://blog.csdn.net/haoel/article/details/1948051/>