|  |  |
| --- | --- |
| 1. 多重继承 2. 多重继承的派生类从每个基类中继承状态 3. 派生类的构造函数初始化所有基类 4. 继承的构造函数与多重继承 5. 析构函数与多重继承 6. 多重继承的派生类的拷贝与移动操作 | Class b:public a,public c{}  :a(name),c(sont){}  :c(sont){}  Using a::a;  Using c::c; |
| 1. **多重继承** 2. 指从多个直接基类中产生派生类的能力 3. 在给定的派生列表中，同一个基类只能出现一次 4. 多重继承的派生类从每个基类中继承状态 5. 派生类的对象包含有每个基类的子对象 6. 派生类的构造函数初始化所有基类 7. 构造一个派生类的对象将同时构造并初始化它的所有基类子对象，也只能初始化直接基类 8. 不直接写的话就是隐式执行基类的默认构造函数 9. 顺序是：a的基类，基类a，基类c，派生类b 10. 继承的构造函数与多重继承 11. 允许派生类从它的一个或几个基类中继承构造函数，若从多个基类继承了相同的构造函数产生错误（得不是默认构造函数版本才会产生错误）。使用using 声明来继承 12. 若从两个基类中继承了相同的构造函数则需要自定义它自己的版本并实现父类版本   就是父类若是父类名(string n){}的构造函数，子类继承的话也是子类名(string n):  父类(n)的样式：D1（string b）:B(n)   1. 析构函数与多重继承   1）与构造函数顺序相反   1. 多重继承的派生类的拷贝与移动操作 2. 与构造函数的顺序一样，只是部分实现形式不一样。 | |