|  |  |
| --- | --- |
| 1. OPP概述 2. 继承 3. 动态绑定 | 面向对象程序设计  继承  基类  派生类  虚函数  Virtual double net\_price() const;  类派生列表  Class A : public B{};  Virtual double net\_price()cont override;  运行时绑定  Quote& item = new Bluk\_quote; |
| 1. OPP概述 2. 面向对象程序设计的核心思想是数据抽象、继承和动态绑定 3. 数据抽象：将类的接口与实现分离 4. 继承：相似的类型并对其相似关系建模 5. 动态绑定：在一定程度上忽略相似的类型的区别，而以统一的方式使用它们的对象 6. 继承 7. 继承：联系在一起的类构成一种层次关系，基类(根部)，派生类 8. 虚函数：派生类各自定义适合自身的版本，virtual开头，基类中声明也可以定义 9. 类派生列表：明确指出是哪个基类继承而来的，一个冒号 + 访问说明符 + 基类 10. 派生类如果对虚函数重新定义了就需要进行声明，可加virtual 与 override，virtual可不加，override为了显式注明改写基类的虚函数，并做检查作用 11. 动态绑定 12. 能用同一段代码分别处理Quote和Bulk\_quote对象。 13. 动态绑定有时又被称为运行时绑定 14. note：当我们使用基类的引用 或指针调用一个虚函数时将发生动态绑定 | |