

概念题:

1. 在 C++ 中, protected 类成员访问控制的作用是什么?

缓解了封装与继承的矛盾。

2. 在 C++ 中, 继承方式的作用是什么? public 继承方式有什么特殊之处?

实现基类成员在派生类中对外的访问控制。

保持基类成员在派生类中对外的访问控制与基类对外访问控制一致。

3. 使用类聚合的方式和类组合的方式复用代码有什么不同? 什么情况下适合使用聚合? 什么情况下适合使用组合? 在编程时需要注意什么?

在聚合关系中, 被包含的对象与包含它的对象独立创建和消亡, 被包含的对象可以脱离包含它的对象独立存在; 在组合关系中, 被包含的对象随包含它的对象创建和消亡, 被包含的对象不能脱离包含它的对象独立存在。根据上述选择使用聚合或者组合。

注意:

在组合关系中, 成员对象是由包含它的对象创建和撤销。

在聚合关系中, 成员对象不由包含它的对象创建和撤销。

成员对象的创建和撤销的一般原则: 谁创建谁撤销!

4. 聚合/组合相比继承的代码复用有哪些优点? 能否仅仅通过前两者实现代码复用? 为什么?

- 继承与封装存在矛盾, 聚合/组合与封装则不存在这个矛盾。

- 在基于继承的代码复用中, 一个类向外界提供两种接口:

- public: 对象 (实例) 用户

- public+protected: 派生类用户

- 在基于聚合/组合的代码复用中, 一个类对外只需一个接口: public。

不能。

因为

- 继承更容易实现子类型:

- 在 C++ 中, public 继承的派生类往往可以看成是基类的子类型。
 - 在需要基类对象的地方可以用派生类对象去替代。
 - 发给基类对象的消息也能发给派生类对象。

- 具有聚合/组合关系的两个类不具有子类型关系!