

2011年计算机二级考试JAVA知识点整理 (81) PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/646/2021_2022_2011_E5_B9_B4_E8_AE_A1_c97_646648.htm

4.1 代码复用的规则 代码复用是绝大多数程序员所期望的，也是OO的目标之一。总结我多年的编码经验，为了使代码能够最大程度上复用，应该特别注意以下几个方面。

对接口编程 "对接口编程"是面向对象设计(OOD)的第一个基本原则。它的含义是：使用接口和同类型的组件通讯，即，对于所有完成相同功能的组件，应该抽象出一个接口，它们都实现该接口。具体到JAVA中，可以是接口(interface)，或者是抽象类(abstract class)，所有完成相同功能的组件都实现该接口，或者从该抽象类继承。我们的客户代码只应该和该接口通讯，这样，当我们需要用其它组件完成任务时，只需要替换该接口的实现，而我们代码的其它部分不需要改变! 当现有的组件不能满足要求时，我们可以创建新的组件，实现该接口，或者，直接对现有的组件进行扩展，由子类去完成扩展的功能。优先使用对象组合，而不是类继承

优先使用对象组合，而不是类继承是面向对象设计的第二个原则。并不是说继承不重要，而是因为每个学习OOP的人都知道OO的基本特性之一就是继承，以至于继承已经被滥用了，而对象组合技术往往被忽视了。下面分析继承和组合的优缺点：类继承允许你根据其他类的实现来定义一个类的实现。这种通过生成子类的复用通常被称为白箱复用(white-box reuse)。术语"白箱"是相对可视性而言：在继承方式中，父类的内部细节对子类可见。对象组合是类继承之外的另一种复用选择。新的更复杂的功能可以通过组合对象来

获得。对象组合要求对象具有良好定义的接口。这种复用风格被称为黑箱复用(black-box reuse)，因为被组合的对象的内部细节是不可见的。对象只以"黑箱"的形式出现。继承和组合各有优缺点。类继承是在编译时刻静态定义的，且可直接使用，类继承可以较方便地改变父类的实现。但是类继承也有一些不足之处。首先，因为继承在编译时刻就定义了，所以无法在运行时刻改变从父类继承的实现。更糟的是，父类通常至少定义了子类的部分行为，父类的任何改变都可能影响子类的行为。如果继承下来的实现不适合解决新的问题，则父类必须重写或被其他更适合的类替换。这种依赖关系限制了灵活性并最终限制了复用性。对象组合是通过获得对其他对象的引用而在运行时刻动态定义的。由于组合要求对象具有良好定义的接口，而且，对象只能通过接口访问，所以我们并不破坏封装性。只要类型一致，运行时刻还可以用一个对象来替代另一个对象。更进一步，因为对象的实现是基于接口写的，所以实现上存在较少的依赖关系。优先使用对象组合有助于你保持每个类被封装，并且只集中完成单个任务。这样类和类继承层次会保持较小规模，并且不太可能增长为不可控制的庞然大物(这正是滥用继承的后果)。另一方面，基于对象组合的设计会有更多的对象(但只有较少的类)，且系统的行为将依赖于对象间的关系而不是被定义在某个类中。注意：理想情况下，我们不用为获得复用而去创建新的组件，只需要使用对象组合技术，通过组装已有的组件就能获得需要的功能。但是事实很少如此，因为可用的组件集合并不丰富。使用继承的复用使得创建新的组件要比组装已有的组件来得容易。这样，继承和对象组合常一起使用。然而，

正如前面所说，千万不要滥用继承而忽视了对对象组合技术。
100Test 下载频道开通，各类考试题目直接下载。详细请访问
www.100test.com