

Java程序基础入门必看 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/278/2021_2022_Java_E7_A8_8B_E5_BA_8F_c104_278562.htm (1) 类名首字母应该大写。字段、方法以及对象(句柄)的首字母应小写。对于所有标识符，其中包含的所有单词都应紧靠在一起，而且大写中间单词的首字母。例如：`ThisIsAClassName thisIsMethodOrFieldName` 若在定义中出现了常数初始化字符，则大写`static final`基本类型标识符中的所有字母。这样便可标志出它们属于编译期的常数。Java包(Package)属于一种特殊情况：它们全都是小写字母，即便中间的单词亦是如此。对于域名扩展名称，如`com`，`org`，`net`或者`edu`等，全部都应小写(这也是Java 1.1和Java 1.2的区别之一)。(2) 为了常规用途而创建一个类时，请采取“经典形式”，并包含对下述元素的定义：`equals()` `hashCode()` `toString()` `clone()`(`implement Cloneable`) `implement Serializable` (3) 对于自己创建的每一个类，都考虑置入一个`main()`，其中包含了用于测试那个类的代码。为使用一个项目中的类，我们没必要删除测试代码。若进行了任何形式的改动，可方便地返回测试。这些代码也可作为如何使用类的一个示例使用。(4) 应将方法设计成简要的、功能性单元，用它描述和实现一个不连续的类接口部分。理想情况下，方法应简明扼要。若长度很大，可考虑通过某种方式将其分割成较短的几个方法。这样做也便于类内代码的重复使用(有些时候，方法必须非常大，但它们仍应只做同样的一件事情)。(5) 设计一个类时，请设身处地为程序员考虑一下(类的使用方法是应该是非常明确的)。然后，再设身处地为管理代码

的人考虑一下(预计有可能进行哪些形式的修改,想想用什么方法可把它们变得更简单)。(6)使类尽可能短小精悍,而且只解决一个特定的问题。下面是对类设计的一些建议:

- 一个复杂的开关语句:考虑采用“多形”机制
- 数量众多的方法涉及到类型差别极大的操作:考虑用几个类来分别实现
- 许多成员变量在特征上有很大的差别:考虑使用几个类。

(7)让一切东西都尽可能地“私有”private。可使库的某一部分“公共化”(一个方法、类或者一个字段等等),就永远不能把它拿出。若强行拿出,就可能破坏其他人现有的代码,使他们不得不重新编写和设计。若只公布自己必须公布的,就可放心大胆地改变其他任何东西。在多线程环境中,隐私是特别重要的一个因素只有private字段才能在非同步使用的情况下受到保护。

(8)警惕“巨大对象综合症”。对一些习惯于顺序编程思维、且初涉OOP领域的新手,往往喜欢先写一个顺序执行的程序,再把它嵌入一个或两个巨大的对象里。根据编程原理,对象表达的应该是应用程序的概念,而非应用程序本身。

(9)若不得已进行一些不太雅观的编程,至少应该把那些代码置于一个类的内部。

(10)任何时候只要发现类与类之间结合得非常紧密,就需要考虑是否采用内部类,从而改善编码及维护工作(参见第14章14.1.2小节的“用内部类改进代码”)。

(11)尽可能细致地加上注释,并用javadoc注释文档语法生成自己的程序文档。

(12)避免使用“魔术数字”,这些数字很难与代码很好地配合。如以后需要修改它,无疑会成为一场噩梦,因为根本不知道“100”到底是指“数组大小”还是“其他全然不同的东西”。所以,我们应创建一个常数,并为其使用具有说服力的描述性名称,并在整个程序中

都采用常数标识符。这样可使程序更易理解以及更易维护。
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