

详细解述7个软件开发原则[1] PDF转换可能丢失图片或格式  
，建议阅读原文

[https://www.100test.com/kao\\_ti2020/234/2021\\_2022\\_\\_E8\\_AF\\_A6\\_E7\\_BB\\_86\\_E8\\_A7\\_A3\\_E8\\_c104\\_234250.htm](https://www.100test.com/kao_ti2020/234/2021_2022__E8_AF_A6_E7_BB_86_E8_A7_A3_E8_c104_234250.htm) 关于代码重复最著名的单词是Kent Beck的Once And Only Once，也就是说软件操作的任何一个片断--不管是一个算法，一个常量集合，用于阅读的文档或者其他东西--应当只出现一次。软件重复出现至少会导致以下问题：其中的一个版本会过期 代码的责任会四处散开，导致代码难以理解 当你修改代码时，需要重复修改很多地方，一不小心就会遗漏 你不能很好地进行性能优化 我以前的一位老板曾经跟我夸耀过他手下编程的能力：“他只要把一份模版代码拷过去，稍加修改，就可以完成一个新的模块”。我惊讶这位程序员思路清晰的同时也怀疑这样的程序除了他自己以外还有谁能维护，我想可能连他自己也无法做到。重复代码的产生由各种各样的原因，上面的例子就是一个，我经常看到程序员把几行或一整段代码从这里复制到那里，然后少加修改，就变成了一份新的代码。这里的原因是程序员可以通过极少的努力就完成代码重用，但是我们可以来看看DavidHooker提出的7个软件开发原则: 1.第一原则：存在的理由(Pattern: TheReason) 一个软件系统存在的理由就是：为它的用户提供价值。你所有的决定都取决于这一点。在指定一个系统需求，在写下一段系统功能，在决定硬件平台和开发过程之前，问你自己一个问题，“这样做会为系统增加价值吗？”，如果答案是”yes”，做。如果是”No”，不做。这个原则是其他原则的原则。 2．第二原则（能简单就简单，愚蠢！）KISS (Pattern: KeepItSimple) 软件设计不是

一个轻描淡写的过程。在做任何一个设计时，你必须考虑很多因素。所有设计应当尽可能简单，但是不要再比这简单了。这样产生的系统才是可以理解和容易维护的。这并不是说很多由意义的特性，因为这种简单性也要被抛弃。确实很多更优雅的设计往往更简单，但简单并不意味着“quick and dirty.”。事实上，简单是通过许多思考和一次又一次的反复修改才达到的。这些努力的汇报就是更容易维护，代码错误更少。（看看是否违反）

3. 第三原则：保持远见(Pattern: MaintainTheVision) 清晰的远见是一个软件项目成功的基础。没有这样的远见，项目开发最后就变成天天为一个不好的设计做补丁。Brooks说过：概念的完整性是系统设计中最重要的问题。Stroustrup 也说：有一个干净的内部结构识构建一个可理解、可辨识、可维护、可测试系统的基础。Booch则总结道：只有当你对系统的体系由一个清晰的感觉，才可能去发现通用的抽象和机制。开发这种通用性最终导致系统更简单，因此更小，更可靠 如果你不断地复制、粘贴、修改代码，最终你将陷入一个大泥潭(the Big Mud),你永远不可能对系统有一个清晰的认识。

4.第四原则：你制造的，别人会消费 (Pattern: WhatYouProduceTheyConsume) 软件系统不是在真空中使用的。其他人会使用、维护、文档你的系统。这依赖于对你系统的理解。所以，你设计、实现的东西应当能够让别人理解。要记住，你写的代码并非只给计算机看，你要时时记住，代码还要给人看。(Kent Beck) 如果到处泛滥似是而非的代码，别人如何能够辨别这些代码的相似和不同，如何去理解这些代码之间具有何种关系。

5. 第五原则：对将来开放( Pattern BuildForTodayDesignForTomorrow) 一个成功的软

件有很长的生命期。你必须能够使得软件能够适应这样和那样的变化。所以，一开始就不要软件设计到死角上去。请总是问一下自己“如果这样，那么。。？”这个问题，你要考虑到各种各样的可能性，而不光光是图省事。复制，粘贴一下即可。

6．第六原则：为重用做好计划 软件模式是重用计划的一种。不断重复的代码显然不是这样的计划。(See CommentsOnSix)

7．第七原则：思考！在采取任何动作之前首先做一个清晰、完整的考虑，这样才能产生更好的结果。如果你考虑了，但还是产生错误的结果，那么这种努力也是值得的。在你学习或研究类似的问题时，更容易理解和掌握。

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](http://www.100test.com)