

第 21 章 决定什么是重要的

良好软件设计中最重要元素之一是区分什么是重要的和什么是不重要的。围绕重要的事情设计软件系统。对于不重要的事情，尽量减少它们对系统其余部分的影响。重要的事情应该得到强调以使其更明显，不重要的事情应该尽可能隐藏起来。

前面章节中的许多想法本质上都是关于区分什么是重要的和什么是不重要的。例如，这就是我们在设计抽象时所做的。模块的接口反映了对模块的用户来说重要的事情，而对模块的用户来说不重要的事情应该隐藏到实现中。在选择变量名称时，目标是挑选几个单词来最好地传达变量的信息，并在名称中使用这些单词，这些是变量最重要的方面。如果性能对于一个模块真的很重要，那么该模块的设计就应该围绕实现性能目标而构建。在[第 20.4 节](#)的例子中，这意味着找到一个性能关键路径上的方法调用和特殊情况检查尽可能少的设计，同时仍然保持整洁、简单和易理解。

21.1 如何决定什么是重要的？

有时候重要的事情来自系统的外部约束，比如[第 20.4 节](#)中的性能。更多的时候，设计者必须自己确定什么是重要的。即使有外部约束，设计者也必须弄清楚在实现这些约束时最重要的是什么。

要决定什么是重要的，可以寻找杠杆点，也就是一个解决方案可以解决许多其他问题的地方，或者知道一些信息后，理解其他事情就容易多了的地方。例如，在[第 6.2 节](#)关于如何存储文本的讨论中，用于插入和删除字符范围的通用接口可以解决许多问题，而专门的方法如 `backspace` 只能解决单个的问题。通用的接口提供了更大的杠杆。在文本类接口的层级上，重要的是文本需要被删除，而不是接口是否在响应退格键时被调用。不变量是另一个杠杆点的例子：一旦为变量或结构确定了一个不变量，就可以预测该变量或结构在许多不同情况下的行为。

如果有很多选择，确定什么是最重要的就容易多了。例如，在选择变量名时，先列出与该变量相关的单词，然后选择几个最能传达信息的单词，使用这些单词来形成变量名。这是一个“设计两次”原则的例子。

有时候什么是最重要的可能不是很明显，对于没有太多经验的年轻开发人员尤其如此。在这种情况下，我建议提出一个假设：“我认为这是最重要的事情”。然后，坚持这个假设，基于该假设构建系统，看看它会如何工作。如果您的假设是正确的，请思考为什么它最终是正确的，以及是否有你以后可以使用的线索。如果您的假设是错误的，也是可以接受的：思考为什么它最终是错误的，以及是否有你以后可以用来避免这个选择的线索。无论哪种方式，您都能从经验中学习，并且您将逐渐做出越来越好的选择。

21.2 最小化重要的事情

可能的话，尽量让重要的事情少一些，这可以产生更简单的系统。例如，尝试最小化构造对象时必须指定的参数数量，或者提供反映最常见用法的默认值。对于确实重要的事情，尽量减少它们出现的地方。隐藏在模块内部的信息对模块外部代码来说不重要。如果异常可以在系统较低层级完全处理，那么它对系统其余部分来说就不重要。如果一个配置参数

可以根据系统行为自动计算出来（而不是让管理员手动选择），那么它对管理员来说就不重要。

21.3 如何强调重要的事情

一旦您识别了什么是重要的事情，您应该在设计中强调它们。一种强调方法是突出显示：重要的事情应该出现在更可能被看到的地方，比如接口文档、名称或是被重度使用的方法的参数。另一种强调方法是重复：关键的思想应反复出现。第三种强调方法是中心化：最重要的事情应该位于系统的中心，它们决定了周围事物的结构。一个例子是操作系统中的设备驱动程序的接口，这就是中心思想，因为成百上千个驱动程序将依赖于它。

当时，反过来也是成立的：如果一个想法更可能被看到，或者它反复出现，或者它以显著的方式影响系统的结构，那么这个想法就是重要的。

类似的，不重要的事情不应该被强调。应该尽可能将其隐藏起来，它们不应该经常被遇见，它们也不应该影响系统的结构。

21.4 常见的错误

在决定什么是重要的时候，有两种常见的错误。第一种错误是认为重要的东西太多。当这种情况发生时，不重要的事情会弄乱设计，增加复杂性和提高认知负荷。一个例子是方法的部分参数与大多数调用者无关。另一个例子是[第 4.7 节](#)讨论的 Java I/O 接口，它迫使开发人员了解缓冲和非缓冲 I/O 之间的区别，即使这个区别几乎从来都不是重要的（开发人员几乎总是想要缓冲，并且不想浪费时间来明确要求它）。将太多的东西视为重要的往往会导致浅的类。

第二种错误是未能认识到某些事情是重要的。这种错误会导致重要的信息被隐藏，或者重要的功能不可用，导致开发人员必须不断重新创建它。这种错误的后果使开发人员的生产力降低，并导致“未知的未知”问题。

21.5 打开思路

关注重要事情的想法也适用于软件设计之外的其它领域。在技术写作中，使文档易于阅读的最佳方法是先确定几个关键概念，并围绕它们组织文档的其余部分。讨论系统的细节时，将它们与总体概念联系起来也会有所帮助。

关注重要事情也是一种伟大的生活哲学：确定对你最重要的事情，尽可能将你的精力花在这些事情上。不要把时间浪费在您认为不重要或无益的事情上。

“好的品味”这个词描述的是能够区分什么是重要的和什么是不重要的能力。拥有良好的品味是成为一名优秀软件设计师的重要部分。