

三步教你改善Java代码质量Java认证考试 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/620/2021\\_2022\\_\\_E4\\_B8\\_89\\_E6\\_AD\\_A5\\_E6\\_95\\_99\\_E4\\_c104\\_620976.htm](https://www.100test.com/kao_ti2020/620/2021_2022__E4_B8_89_E6_AD_A5_E6_95_99_E4_c104_620976.htm) 本文讨论了如何以递进的方式使用Apache Ant来改善我们的代码质量。并按着三步走的原则给出了具体的实现步骤。 一、充分利用单元测试、代码覆盖 单元测试、代码覆盖是最容易被接受和实现的方式。事实上，大多数开发人员都知道单元测试对他们很重要。在我们开始讨论这些东西之前，先看一下Google研究院主管Peter Norvig的一段话：“如果你认为你们不需要对自己的代码进行单元测试，那么就写在纸上写一所有的原因，并且仔细研究这张纸，然后扔了它，继续测试自己的程序吧”。看来Google也是非常推崇进行单元测试的。那么谁又来测试那么测试者呢？也就是说，我们怎么能验证对程序做了足够的测试呢？这是一个非常有价值的问题，因为那些未通过测试的程序才是我们更应该关注的地方。这个问题的一个解决方案就是使用代码覆盖工具，这种工具将告诉我们我们到底测试了多少代码（也就是被测试代码的百分比），然后使用一般的综合处理来合并覆盖核对结果。如果覆盖核对失败，那么我们建立应用程序的过程也就失败了。对于本文所讨论的递增代码策略选择了代码覆盖工具Cobertura，这是由于它非常容易使用，而且拥有良好的定义格式，以及四个Ant任务接口。这些任务之一就是cobertura-check，当代码不能完成我们要求的覆盖率时，它就会失败。如下面的代码显示如果覆盖率未达到80%，Ant在建立工程时就会失败：除了使用硬编码来指定这个覆盖率外，我们还可能以使用一个更容易建立

的结果作为当前核对的覆盖率。我们可以通过使用两个核心Ant任务连接一对Cobertura任务来完成这个任务。并不不用担心各种覆盖率的具体的值。我们的目标是完成可测量的代码改善，而不是设置一个绝对的代码覆盖率。在建立用于测试和运行我们的代码的targets后，我们可以将用于核对我们的建立脚本的增量覆盖率。第一步是使用一个cobertura报任务建立一个XML格式的覆盖报表。代码如下：下面是由一个cobertura报表任务产生的 coverage.xml

```
gt. gt. ./src/java .....
```

现在要确保将这个文件保存在某个硬盘上，因为我们在后面会需要这个文件。二、从报表中取出覆盖率 一开始，我们可以试着使用Ant的XmlProperty任务来直接获得这个覆盖率，并给一个Ant属性。但是这个方法有以下两个问题：1. 在coverage.xml中的覆盖率是一个小数，但当核对任务时需要一个整数百分比。2. 在实际的项目中，coverage.xml的文件尺寸非常大，如果在Java中尝试使用XmlProperty任务时可能会出现Java OutOfMemoryError错误。而我们只想从coverage.xml文件中获得以下的内容：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)