

Java的final变量需要注意地方 PDF转换可能丢失图片或格式，  
建议阅读原文

[https://www.100test.com/kao\\_ti2020/144/2021\\_2022\\_Java\\_E7\\_9A\\_84fin\\_c104\\_144475.htm](https://www.100test.com/kao_ti2020/144/2021_2022_Java_E7_9A_84fin_c104_144475.htm) 对与java中的final变量，java编译器是进行了优化的。每个使用了final类型变量的地方都不会通过连接而进行访问。比如说Test类中使用了Data类中一个final的int数字fNumber=77，这时候，java编译器会将77这个常数编译到Test类的指令码或者常量池中。这样，每次Test类用到fNumber的时候，不会通过引用连接到Data类中进行读取，而是直接使用自己保存在类文件中的副本。用程序说话

```
: Test.java: public class Test{ public static void main(String[] args){
System.out.println(Data.fNumber). }}Data.java:public class Data{
public static final int fNumber=77. }执行命令和结果：Microsoft
Windows XP [版本 5.1.2600](C) 版权所有 1985-2001 Microsoft
Corp.C:\Documents and Settings\zangmeng>cd ..C:\Documents
and Settings>cd ..C:\>javac Test.javaC:\>java Test77C:\>这时候，
我们更改Data.java的内容：public class Data{ public static final int
fNumber=777. }然后执行如下命令：C:\>javac Data.javaC:\>java
Test77C:\>这里我们看到，虽然Data.java中的fNumber已经更改
为777，而且已经重新编译了，但是因为编译器把fNumber的
副本保存Test类中，所以在重新编译Test类的前，Test类一直
把fNumber认为是77而不是777。下面我们变异Test.java，再执
行，看看结果。C:\>javac Test.javaC:\>java Test777C:\>这时候
，我们看到，重新编译的Test类将新的777数值封装到了自己
类中。整个过程如下：Microsoft Windows XP [版本
5.1.2600](C) 版权所有 1985-2001 Microsoft Corp.C:\Documents
```

```
and Settings\zangmeng>cd ..C:\Documents and Settings>cd
..C:\>javac Test.javaC:\>java Test77//在这里改变了Data.java的内
容C:\>javac Data.javaC:\>java Test77C:\>javac Test.javaC:\>java
Test777C:\> 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
```