

谈java的导入(import) PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/270/2021_2022__E8_B0_88java_E7_9A_84_c67_270955.htm 单类型导入 (single-type-import)，例如 `import java.io.File`；按需类型导入

(type-import-on-demand)，例如 `import java.io.*`；关于这两种导入类型大家各有所爱，众说纷纭。这里分析一下这两种导入类型的大致工作原理供大家参考。单类型导入比较好理解，仅仅导入一个public类或者接口。而对于按需类型导入，有人误解为导入一个包下的所有类，其实不然，看名字就知道，他只会按需导入，也就是说它并非导入整个包，而仅仅导入当前类需要使用的类。既然如此是不是就可以放心的使用按需类型导入呢？非也，非也。因为单类型导入和按需类型导入对类文件的定位算法是不一样的。java编译器会从启动目录 (bootstrap)，扩展目录 (extension) 和用户类路径下去定位需要导入的类，而这些目录仅仅是给出了类的顶层目录。编译器的类文件定位方法大致可以理解为如下公式：顶层路径名 \ 包名 \ 文件名。class = 绝对路径 对于单类型导入很简单，因为包名和文件名都已经确定，所以可以一次性查找定位。对于按需类型导入则比较复杂，编译器会把包名和文件名进行排列组合，然后对所有的可能性进行类文件查找定位。例如：`package com`；`import java.io.*`；`import java.util.*`；当你的类文件中用到了File类，那么可能出现File类的地方如下：
1、File \ File类属于无名包，就是说File类没有package语句，编译器会首先搜索无名包
2、com.File \ File类属于当前包
3、java.lang.File \ 编译器会自动导入java.lang包
4、java.io.File
5

、`java.util.File` 需要注意的地方就是，编译器找到`java.io.File`类之后并不会停止下一步的寻找，而要把所有的可能性都查找完以确定是否有类导入冲突。假设此时的顶层路径有三个，那么编译器就会进行 $3*5=15$ 次查找。了解以上原理之后，我们可以得出这样的结论：按需类型导入是绝对不会降低Java代码的执行效率的，但会影响到Java代码的编译速度。查看JDK的源代码就知道SUN的软件工程师一般不会使用按需类型导入。因为使用单类型导入至少有以下两点好处：1.提高编译速度。2.避免命名冲突。（例如：当你`import java.awt.*`；`import java.util.*`后，使用List的时候编译器将会出编译错误）当然，使用单类型导入会使用你的import语句看起来很长。

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