

Java学习路径 - 方法篇资源篇 PDF转换可能丢失图片或格式，
建议阅读原文

https://www.100test.com/kao_ti2020/138/2021_2022_Java_E5_AD_A6_E4_B9_A0_c97_138461.htm Java学习路径（四）方法篇 Java作为一门编程语言，最好的学习方法就是写代码。当你学习一个类以后，你就可以自己写个简单的例子程序来运行一下，看看有什么结果，然后再多调用几个类的方法，看看运行结果，这样非常直观的把类给学会了，而且记忆非常深刻。然后不应该满足把代码调通，你应该想想看如果我不这样写，换个方式，再试试行不行。记得哪个高人说过学习编程就是个破坏的过程，把书上的例子，自己学习Documentation编写的例子在运行通过以后，不断的尝试着用不同的方法实现，不断的尝试破坏代码的结构，看看它会有什么结果。通过这样的方式，你会很彻底的很精通的掌握Java。举个例子，我们都编过Hello World

```
public class HelloWorld { public static void main(String[] args) { System.out.println("Hello World"). } }
```

很多初学者不是很理解为什么main方法一定要这样来定义public static void main(String[] args)，能不能不这样写？包括我刚学习Java的时候也有这样的疑问。想知道答案吗？很简单，你把main改个名字运行一下，看看报什么错误，然后根据出错信息进行分析；把main的public取掉，在试试看，报什么错误；static去掉还能不能运行；不知道main方法是否一定要传一个String[]数组的，把String[]改掉，改成int[]，或者String试试看；不知道是否必须写args参数名称的，也可以把args改成别的名字，看看运行结果如何。我当初学习Java的时候就是这样做的，把Hello World程序反复改了七八次，不断运行，分

析运行结果，最后就彻底明白为什么了main方法是这样定义的了。此外，我对于static, public, private, Exception, try{ }catch {}finally{}等等等等一开始都不是很懂，都是把参考书上面的例子运行成功，然后就开始破坏它，不断的根据自己心里面的疑问来重新改写程序，看看能不能运行，运行出来是个什么样子，是否可以得到预期的结果。这样虽然比较费时间，不过一个例子程序这样反复破坏几次之后。我就对这个相关的知识彻底学通了。有时候甚至故意写一些错误的代码来运行，看看能否得到预期的运行错误。这样对于编程的掌握是及其深刻的。其中特别值得一提的是JDK有一个非常棒的调试功能，-verbose java verbose javac verbose 以及其它很多JDK工具都有这个选项 -verbose 可以显示在命令执行的过程中，JVM都依次加载哪里Class，通过这些宝贵的调试信息，可以帮助我们分析出JVM在执行的过程中都干了些什么。另外，自己在学习过程中，写的很多的这种破坏例程，应该有意识的分门别类的保存下来，在工作中积累的典型例程也应该定期整理，日积月累，自己就有了一个代码库了。遇到类似的问题，到代码库里面 Copy amp. Replace，就好了，极大提高了开发速度。最理想的情况是把一些通用的例程自己再抽象一层，形成一个通用的类库，封装好。那么可复用性就更强了。所以我觉得其实不是特别需要例程的，自己写的破坏例程就是最好的例子，如果你实在对自己写的代码不放心的话，我强烈推荐你看看JDK基础类库的Java源代码。在JDK安装目录下面会有一个src.zip，解开来就可以完整的看到整个JDK基础类库，也就是rt.jar的Java源代码，你可以参考一下Sun是怎么写Java程序的，规范是什么样子的。我自己在学

习Java的类库的时候，当有些地方理解的不是很清楚的时候，或者想更加清晰的理解运作的细节的时候，往往会打开相应的类的源代码，通过看源代码，所有的问题都会一扫而空。
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