

如何解决Java语言的十大问题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\\_ti2020/644/2021\\_2022\\_\\_E5\\_A6\\_82\\_E4\\_BD\\_95\\_E8\\_A7\\_A3\\_E5\\_c104\\_644772.htm](https://www.100test.com/kao_ti2020/644/2021_2022__E5_A6_82_E4_BD_95_E8_A7_A3_E5_c104_644772.htm) 别误解，其实在我的职业生涯中，我已经编写了无数的Java代码；而且，我仍然认为Java一门伟大的（程序）语言。相对于C和Smalltalk，Java已经有了很大的改进；但现在，即使是Java，也开始感觉到了其15年的积重。事实上，在我的经历中，我总是不得不面对Java的设计和规格上的一些错误、缺陷和不足，这些东西，让我的Java程序员生活少有乐趣可言。现在全世界的Java程序员有数百万之众，Java写就的代码更达数亿行，要是我说Java在不久的将来死去，这还有些远。不管怎样，随着一些兼容JVM的语言出现（我最钟意Scala）后，这些问题变得越发不能容忍了，我开始想，是时候慢慢离开Java了（但并不脱离JVM）。具体说来，我认为Java语言的10大问题是：1、缺少闭包（closure）：我想这个不需要解释了。函数式编程已经存在几十年了，但最近几年，它们获得了越来越多的关注，最主要的原因，是它可以自然地编写并行程序。我部分的同意Joshua Bloch强调在Java中引入闭包的问题需要再想一想（BGGA提议的方式真的很糟），至少闭包的缺失，使得在Java中做任何真正的函数式编程都是不可能的。2、缺少一等函数：这个问题与前一个有些关联，但我认为它更糟糕。在Java里，要达到类似效果的唯一方式，是使用著名的、丑陋悲惨的单方法匿名内部类，但这看上去的确是一个拙劣的方法。甚至在C#中，也通过代理机制，提供了一个更好的实现。3、原生类型（Primitive types）：如果在Java中一切皆对象

，那是多么完美啊，但他们偏偏不这样设计。因而，这一点导致了一些问题，比如，不能把一个int放到集合（Collection）里，这个在Java5中通过自动装箱特性得到了解决（下面会提到）。它也造成了传值与传引用上的困扰，原生类型数据是通过值传给方法的（复制一份拷贝，然后传给函数），而真正的对象是通过传递（译注：其实是复制对象地址再传递，因此应该也是传值方式，只是由于函数内部可通过这个对象地址访问对象，因此效果上类似传引用）。4、自动装箱（Autoboxing）和自动拆箱（autounboxing）：这个特性是为了解决因原生类型的存在所导致的问题，在Java5引入的。它允许静默地转换原生类型到相应的对象，但这常常导致其它的问题。比如Integer可以为null，但int不能，因此这时JVM只能抛出一个难以调试的空指针异常（NullPointerException）。此外，它还可能导致其它奇怪的行为，就像下面的例子，我们就很难理解，变量test为什么是false：

```
Integer a = new Integer(1024). Integer b = new Integer(1024). boolean test = a == b
```

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)