<https://mp.weixin.qq.com/s/JmcA7ZQAg3_VGZX7XVR-tA>

## 为什么 Java 不支持运算符重载？

另一个类似的 Java 面试难题。为什么 C++ 支持运算符重载而 [Java](http://mp.weixin.qq.com/s?__biz=MzI3ODcxMzQzMw==&mid=2247493562&idx=1&sn=72feda976bc285ee1e22f0ef8e94fbc2&chksm=eb50628cdc27eb9ae5a3ee0becd5d73e91633c466fd5e695457d50018af2c3fc130206b84a56&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank) 不支持? 有人可能会说 + 运算符在 [Java](http://mp.weixin.qq.com/s?__biz=MzI3ODcxMzQzMw==&mid=2247493562&idx=1&sn=72feda976bc285ee1e22f0ef8e94fbc2&chksm=eb50628cdc27eb9ae5a3ee0becd5d73e91633c466fd5e695457d50018af2c3fc130206b84a56&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank) 中已被重载用于字符串连接，不要被这些论据所欺骗。

与 C++ 不同，Java 不支持运算符重载。Java 不能为程序员提供自由的标准算术运算符重载，例如+， -，\*和/等。如果你以前用过 C++，那么 [Java](http://mp.weixin.qq.com/s?__biz=MzI3ODcxMzQzMw==&mid=2247493562&idx=1&sn=72feda976bc285ee1e22f0ef8e94fbc2&chksm=eb50628cdc27eb9ae5a3ee0becd5d73e91633c466fd5e695457d50018af2c3fc130206b84a56&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank) 与 C++ 相比少了很多功能，例如 [Java](http://mp.weixin.qq.com/s?__biz=MzI3ODcxMzQzMw==&mid=2247493562&idx=1&sn=72feda976bc285ee1e22f0ef8e94fbc2&chksm=eb50628cdc27eb9ae5a3ee0becd5d73e91633c466fd5e695457d50018af2c3fc130206b84a56&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank) 不支持多重继承，Java中没有指针，[Java](http://mp.weixin.qq.com/s?__biz=MzI3ODcxMzQzMw==&mid=2247493562&idx=1&sn=72feda976bc285ee1e22f0ef8e94fbc2&chksm=eb50628cdc27eb9ae5a3ee0becd5d73e91633c466fd5e695457d50018af2c3fc130206b84a56&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)中没有地址引用传递。

另一个类似的问题是关于 Java 通过引用传递，这主要表现为 Java 是通过值还是引用传参。虽然我不知道背后的真正原因，但我认为以下说法有些道理，为什么 Java 不支持运算符重载。

**1) 简单性和清晰性。**

清晰性是 [Java](http://mp.weixin.qq.com/s?__biz=MzI3ODcxMzQzMw==&mid=2247493562&idx=1&sn=72feda976bc285ee1e22f0ef8e94fbc2&chksm=eb50628cdc27eb9ae5a3ee0becd5d73e91633c466fd5e695457d50018af2c3fc130206b84a56&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank) 设计者的目标之一。设计者不是只想复制语言，而是希望拥有一种清晰，真正面向对象的语言。添加运算符重载比没有它肯定会使设计更复杂，并且它可能导致更复杂的编译器, 或减慢 [JVM](http://mp.weixin.qq.com/s?__biz=MzI3ODcxMzQzMw==&mid=2247489332&idx=2&sn=65de5886e13b98116c8432d7d10ae4bc&chksm=eb539202dc241b14010f70edf89dc37c7629b5e2b7add50fd3f58070ecf2c14260196bf147d8&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)，因为它需要做额外的工作来识别运算符的实际含义，并减少优化的机会, 以保证 [Java](http://mp.weixin.qq.com/s?__biz=MzI3ODcxMzQzMw==&mid=2247493562&idx=1&sn=72feda976bc285ee1e22f0ef8e94fbc2&chksm=eb50628cdc27eb9ae5a3ee0becd5d73e91633c466fd5e695457d50018af2c3fc130206b84a56&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank) 中运算符的行为。

**2) 避免编程错误。**

Java 不允许用户定义的运算符重载，因为如果允许程序员进行运算符重载，将为同一运算符赋予多种含义，这将使任何开发人员的学习曲线变得陡峭，事情变得更加混乱。

据观察，当语言支持运算符重载时，编程错误会增加，从而增加了开发和交付时间。由于 [Java](http://mp.weixin.qq.com/s?__biz=MzI3ODcxMzQzMw==&mid=2247493562&idx=1&sn=72feda976bc285ee1e22f0ef8e94fbc2&chksm=eb50628cdc27eb9ae5a3ee0becd5d73e91633c466fd5e695457d50018af2c3fc130206b84a56&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)和 [JVM](http://mp.weixin.qq.com/s?__biz=MzI3ODcxMzQzMw==&mid=2247489332&idx=2&sn=65de5886e13b98116c8432d7d10ae4bc&chksm=eb539202dc241b14010f70edf89dc37c7629b5e2b7add50fd3f58070ecf2c14260196bf147d8&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank) 已经承担了大多数开发人员的责任，如在通过提供垃圾收集器进行内存管理时，因为这个功能增加污染代码的机会, 成为编程错误之源, 因此没有多大意义。

**3) JVM复杂性。**

从[JVM](http://mp.weixin.qq.com/s?__biz=MzI3ODcxMzQzMw==&mid=2247489332&idx=2&sn=65de5886e13b98116c8432d7d10ae4bc&chksm=eb539202dc241b14010f70edf89dc37c7629b5e2b7add50fd3f58070ecf2c14260196bf147d8&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)的角度来看，支持运算符重载使问题变得更加困难。通过更直观，更干净的方式使用方法重载也能实现同样的事情，因此不支持 Java 中的运算符重载是有意义的。与相对简单的 [JVM](http://mp.weixin.qq.com/s?__biz=MzI3ODcxMzQzMw==&mid=2247489332&idx=2&sn=65de5886e13b98116c8432d7d10ae4bc&chksm=eb539202dc241b14010f70edf89dc37c7629b5e2b7add50fd3f58070ecf2c14260196bf147d8&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank) 相比，复杂的 [JVM](http://mp.weixin.qq.com/s?__biz=MzI3ODcxMzQzMw==&mid=2247489332&idx=2&sn=65de5886e13b98116c8432d7d10ae4bc&chksm=eb539202dc241b14010f70edf89dc37c7629b5e2b7add50fd3f58070ecf2c14260196bf147d8&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank) 可能导致 [JVM](http://mp.weixin.qq.com/s?__biz=MzI3ODcxMzQzMw==&mid=2247489332&idx=2&sn=65de5886e13b98116c8432d7d10ae4bc&chksm=eb539202dc241b14010f70edf89dc37c7629b5e2b7add50fd3f58070ecf2c14260196bf147d8&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank) 更慢，并为保证在 [Java](http://mp.weixin.qq.com/s?__biz=MzI3ODcxMzQzMw==&mid=2247493562&idx=1&sn=72feda976bc285ee1e22f0ef8e94fbc2&chksm=eb50628cdc27eb9ae5a3ee0becd5d73e91633c466fd5e695457d50018af2c3fc130206b84a56&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank) 中运算符行为的确定性从而减少了优化代码的机会。

**4) 让开发工具处理更容易。**

这是在 [Java](http://mp.weixin.qq.com/s?__biz=MzI3ODcxMzQzMw==&mid=2247493562&idx=1&sn=72feda976bc285ee1e22f0ef8e94fbc2&chksm=eb50628cdc27eb9ae5a3ee0becd5d73e91633c466fd5e695457d50018af2c3fc130206b84a56&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank) 中不支持运算符重载的另一个好处。省略运算符重载后使语言更容易处理，如静态分析等，这反过来又更容易开发处理语言的工具，例如 IDE 或重构工具。Java 中的重构工具远胜于 C++。