#### 为什么要关心Java 8？

核心新特性：Lambada（匿名函数）、流、默认方法。

Java 8对硬件也有影响：绝大多数现有的Java程序都只使用其中一个内核，其他三个都闲着，或只是用一小部分的处理能力来运行操作系统或杀毒程序。

在Java 8之前，专家们可能会告诉你，必须利用线程才能使用多个内核。问题是，线程用起来很难，也容易出现错误。从Java的演变路径来看，它一直致力于让并发编程更容易、出错更少。

* Java 1.0里有线程和锁，甚至有一个内存模型——这是当时的最佳做法。
* Java 5添加了工业级的构建模块，如线程池和并发集合。
* Java 7添加了分支/合并（fork/join）框架，使得并行变得更实用，但仍然很困难。
* Java 8对并行有了一个更简单的新思路，不过仍要遵循一些规则。

首先，快速了解一下这些想法：

* Stream API
* 向方法传递代码的技巧
* 接口中的默认方法

Java 8提供了一个新的API（称为“流”，Stream），它支持许多处理数据的并行操作，其思路和在数据库查询语言中的思路类似——用更高级的方式表达想要的东西，而由“实现”（在这里是Streams库）来选择最佳低级执行机制。这样就可以避免用synchronized编写代码。

在Java 8中加入Streams可以看作把另外两项扩充加入Java 8的直接原因：把代码传递给方法的简洁方式（方法引用、Lambda）和接口中的默认方法。