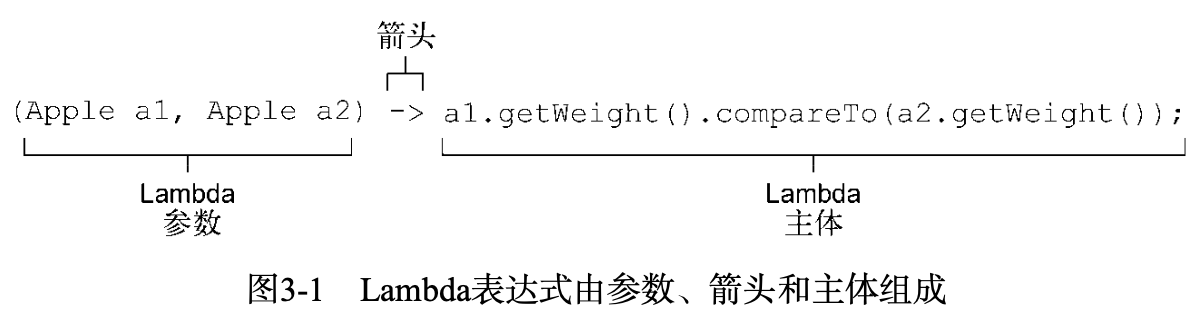
可以把Lambda表达式理解为简洁地表示可传递的匿名函数的一种方式：它没有名称，但它有参数列表、函数主体、返回类型，可能还有一个可抛出的异常列表。

* 匿名——它不像普通的方法那样有一个明确的名称。
* 函数——Lambda函数不像方法那样属于某个特定的类。但和方法一样，Lambda有参数列表、函数主体、返回类型，还可能有可以抛出的异常列表。
* 传递——Lambda表达式可以作为参数传递给方法或存储在变量中。
* 简洁——无需像匿名类那样写很多模板代码。

Lambda这个词从哪儿来的？来自于学术界开发出来的一套用来描述计算的λ演算法。Lambda解决了Java中传递代码的繁琐和冗长。现在不需要用匿名类写一堆笨重的代码。Lambda表达式鼓励采用行为参数化风格。比如，利用Lambda表达式，你可以更为简洁地自定义一个Comparator对象。



现在基本上值传递了比较两个苹果重量所真正需要的代码。看起来就像是只传递了compare方法的主体。甚至还可以进一步简化代码。

Lambda表达式有三个部分：

* 参数列表——它采用了Comparator中compare方法的参数，两个Apple。
* 箭头—— -> 把参数列表与Lambda主体分隔开。
* Lambda主体——比较两个Apple的重量。表达式就是Lambda的返回值了。

Java语言设计者选择这样的语法，是因为C#和Scala等语言中的类似功能广受欢迎。Lambda表达式的基本语法是：

(parameters) -> expression

或

(parameters) -> { statements；}

常见的一些Lambda的例子和使用案例：

