<https://mp.weixin.qq.com/s/xDGt42HSto8Cy7x7qe04Jw>

# Java8中的Stream

Java8中有两个非常有名的改进，一个是Lambda表达式，一个是Stream。如果我们了解过函数式编程的话，都知道Stream真正把函数式编程的风格引入到了java中。这篇文章由简入繁逐步介绍Stream。

# **一、Stream是什么**

从名字来看，Stream就是一个流，他的主要作用就是对集合数据进行查找过滤等操作。有点类似于SQL的数据库操作。一句话来解释就是一种高效且易用的数据处理方式。大数据领域也有一个Steam实时流计算框架，不过和这个可不一样。别搞混了。

举个例子吧，比如说有一个集合Student数据，我们要删选出学生名字为“张三”的学生，或者是找出所有年龄大于18岁的所有学生。此时我们就可以直接使用Stream来筛选。当然了这只是给出了其中一个例子。Stream还有很多其他的功能。

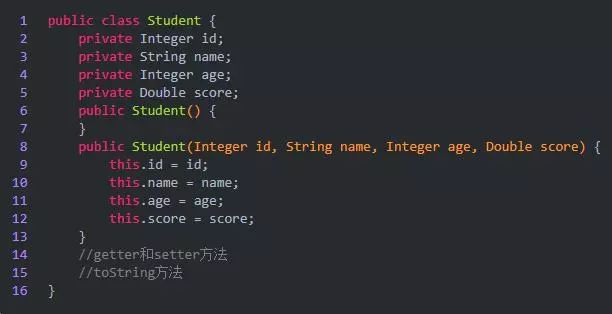
Stream和Collection的区别就是：Collection只是负责存储数据，不对数据做其他处理，主要是和内存打交道。但是Stream主要是负责计算数据的，主要是和CPU打交道。现在明白了吧。

# **二、Stream语法讲解**

Stream执行流程很简单，主要有三个，首先创建一个Stream，然后使用Stream操作数据，最后终止Stream。有点类似于Stream的生命周期。下面我们根据其流程来一个一个讲解。

## 1、前提准备

首先我们创建一个Student类，以后我们每次都是操作这个类



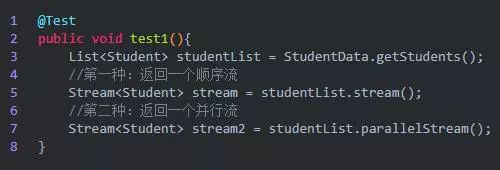
然后下面我们再创建一个StudentData类，用于获取其数据



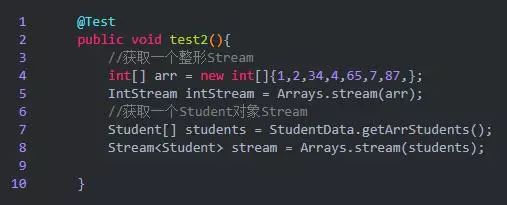
我们只需要把方法变成static类型的就可以了。

## 2、创建一个Stream

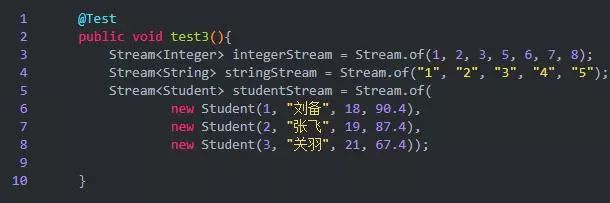
方式一：通过一个集合创建Stream



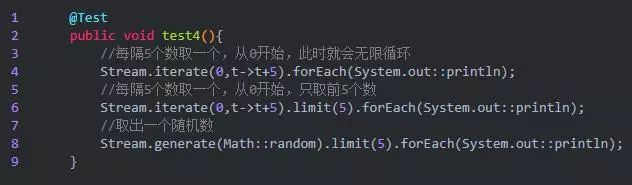
方式二：通过一个数组创建Stream



方式三：通过Stream.of

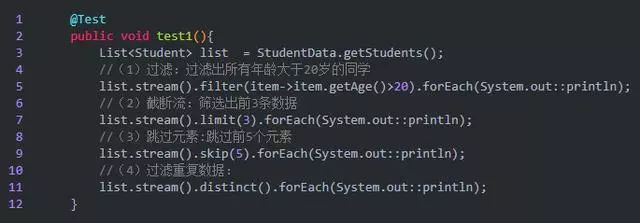


方式四：创建一个无限流

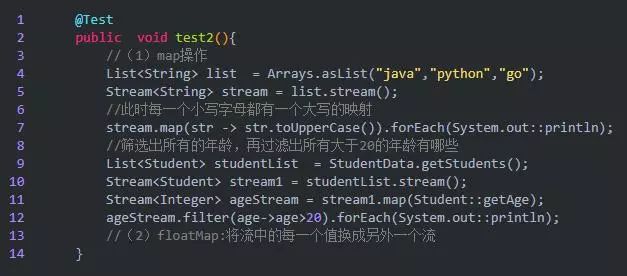


## 使用Stream操作数据

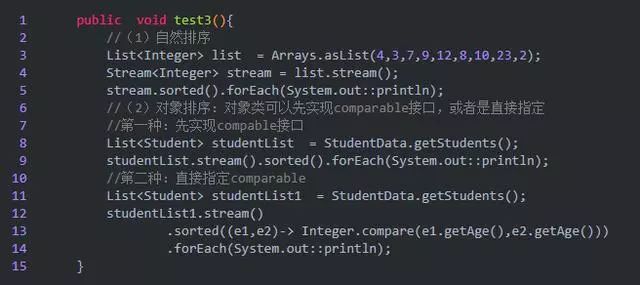
操作1：筛选和切片



操作2:映射

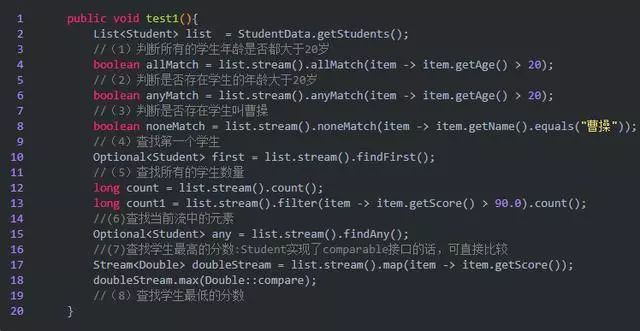


操作3:排序

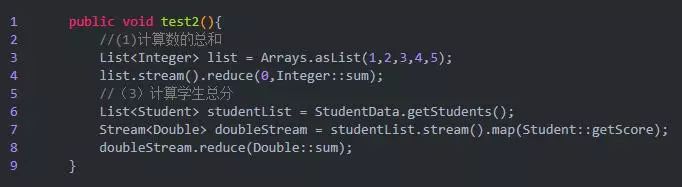


## 4、终止Stream

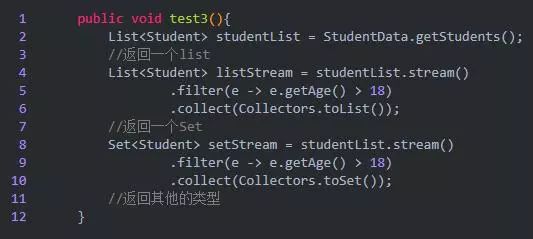
操作1：匹配和查找



操作2：归约



操作3：收集



stream基本的语法就是这样，你会发现Stream就像是一个工具一样，可以帮我们分析处理数据，极其的好用，但是目前还不知道其效率如何。根据网上一位大佬的内存时间分析，其实在数据量比较庞大的时候，Stream可以为我们节省大量的时间，数据量小的时候并不明显。