

Stream流

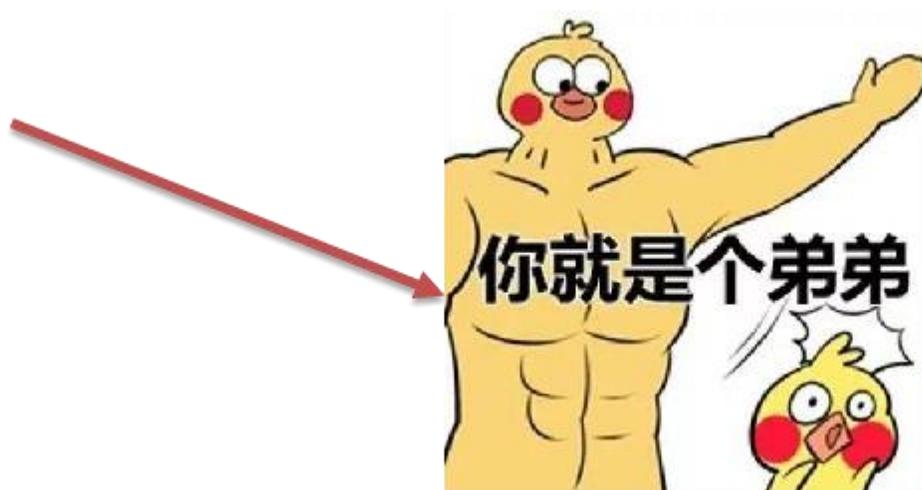
Stream流的介绍

一、什么是Stream流？

在Java 8中，得益于Lambda所带来的函数式编程，引入了一个全新的Stream流概念。

目的：用于简化集合和数组操作的API。

Stream流



集合、数组的API

Stream流的使用步骤

Stream流思想：
Stream流式思想的核心

- 1.先得到集合或者数组的Stream流（就是一根传送带）
- 2.把元素放上去
- 3.然后就用这个Stream流简化的API来方便的操作元素。



利用Stream流中的API进行各种操作



过滤

转换

中间方法

方法调用完毕之后,还可以调用其他方法

统计

打印

终结方法

最后一步,调用完毕后不能调用其他方法

01

如何获取一条Stream流

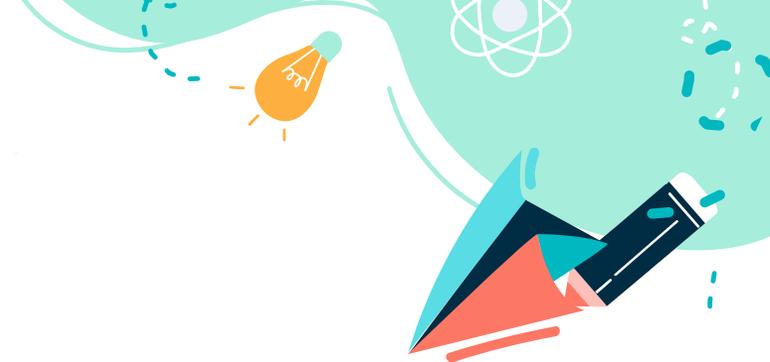
Stream流的获取



Stream操作集合或者数组的第一步是先得到Stream流，然后才能使用流的功能。

获取方式	方法名	说明
单列集合	<code>default Stream<E> stream()</code>	Collection中的默认方法
双列集合	无	无法直接使用stream流
数组	<code>public static <T> Stream<T> stream(T[] array)</code>	Arrays工具类中的静态方法
一堆零散数据	<code>public static<T> Stream<T> of(T... values)</code>	Stream接口中的静态方法

Stream流的中间方法



名称	说明
Stream<T> filter (Predicate<? super T> predicate)	过滤
Stream<T> limit (long maxSize)	获取前几个元素
Stream<T> skip (long n)	跳过前几个元素
Stream<T> distinct ()	元素去重, 依赖(hashCode和equals方法)
static <T> Stream<T> concat (Stream a, Stream b)	合并a和b两个流为一个流
Stream<R> map (Function<T, R> mapper)	转换流中的数据类型

03

Stream流的终结方法



Stream流的终结方法



名称	说明
void forEach (Consumer action)	遍历
long count ()	统计
toArray ()	收集流中的数据，放到数组中
collect (Collector collector)	收集流中的数据，放到集合中

04

课堂总结

When you copy & paste, choose "keep text only" option.

课堂总结

1. Stream流的作用

结合了Lambda表达式，简化集合、数组的操作

2. Stream的使用步骤

- 获取Stream流对象
- 使用中间方法处理数据
- 使用终结方法处理数据

3. 如何获取Stream流对象

- 单列集合：Collection中的默认方法stream
- 双列集合：不能直接获取
- 数组：Arrays工具类型中的静态方法stream
- 一堆零散的数据：Stream接口中的静态方法of

4. 常见方法

中间方法: filter, limit, skip, distinct, concat, map

终结方法: forEach, count, collect

LOGO HERE

**Thank you for
watching.**

OfficePLUS

Speaker name and title