# Stream、异常体系





# 今天同学们需要学会什么

#### 不可变集合

Stream流

认识异常体系

不可变集合对象

有些业务场景下需集合自己提供的API非常 要有不可变集合对 繁琐, JDK 8开始,得益 象, Java如何得到 于Lambda, 提供了操作 集合、数组更好用的技

术:Stream流

程序一旦出现了bug 则会终止,如何尽力 避免程序出现异常, 出现异常如何进行处 理让程序更稳健



- **)** 创建不可变集合
- > Stream流
- **异常处理**



## 什么是不可变集合?

- 不可变集合,就是不可被修改的集合。
- 集合的数据项在创建的时候提供,并且在整个生命周期中都不可改变。否则报错。

"迪丽热巴" "古力娜扎	"卡尔扎巴"	"马儿扎哈"	• • •
--------------	--------	--------	-------

老王





# 为什么要创建不可变集合?

- 如果某个数据不能被修改,把它防御性地拷贝到不可变集合中是个很好的实践。
- 或者当集合对象被不可信的库调用时,不可变形式是安全的。



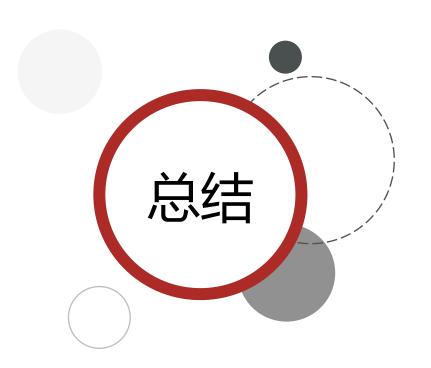
#### 如何创建不可变集合?

● 在List、Set、Map接口中,都存在of方法,可以创建一个不可变的集合。

方法名称	说明
static <e> List<e> of(Eelements)</e></e>	创建一个具有指定元素的List集合对象
static <e> Set<e> of(Eelements)</e></e>	创建一个具有指定元素的Set集合对象
static <k ,="" v=""> Map<k ,="" v=""> of(Eelements)</k></k>	创建一个具有指定元素的Map集合对象

● 这个集合不能添加,不能删除,不能修改。





- 1. 不可变集合的特点?
  - 定义完成后不可以修改,或者添加、删除
- 2. 如何创建不可变集合?
  - List、Set、Map接口中,都存在of方法可以创建不可变集合。



- **)** 创建不可变集合
- Stream流
  - ◆ Stream流的概述
  - ◆ Stream流的获取
  - ◆ Stream流的常用方法
  - ◆ Stream流的综合应用
  - ◆ 收集Stream流
- > 异常处理



### 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。



# **a** 案例

# 体验Stream流的作用

需求:按照下面的要求完成集合的创建和遍历

● 创建一个集合,存储多个字符串元素

```
List<String> list = new ArrayList<>();
list.add("张无忌");
list.add("周芷若");
list.add("赵敏");
list.add("张强");
list.add("张玉丰");
```

- 把集合中所有以"张"开头的元素存储到一个新的集合
- 把"张"开头的集合中的长度为3的元素存储到一个新的集合
- 遍历上一步得到的集合中的元素输出。

#### 就问你服不服





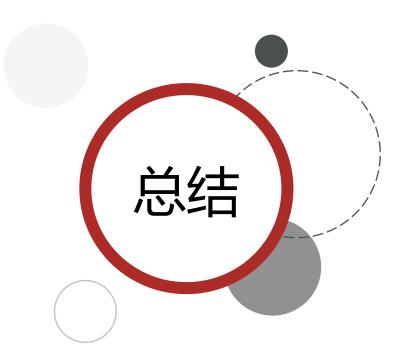
#### Stream流的思想

张三丰 张无忌 张翠山 张良 王二麻子 谢广坤

> 过滤操作 留下以张开头的

过滤操作 留下长度为3的 将剩余的数 据全部输出





- 1、Stream流的作用是什么,结合了什么技术?
  - 简化集合、数组操作的API。结合了Lambda表达式。
- 2、说说Stream流的思想和使用步骤。
  - 先得到集合或者数组的Stream流(就是一根传送带)。
  - 把元素放上去。
  - 然后就用这个Stream流简化的API来方便的操作元素。



- **)** 创建不可变集合
- > Stream流
  - ◆ Stream流的概述
  - ◆ Stream流的获取
  - ◆ Stream流的常用方法
  - ◆ Stream流的综合应用
  - ◆ 收集Stream流
- > 异常处理



#### 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。

### 集合获取Stream流的方式

● 可以使用Collection接口中的默认方法stream()生成流

名称	说明
default Stream <e> stream()</e>	获取当前集合对象的Stream流

# 数组获取Stream流的方式

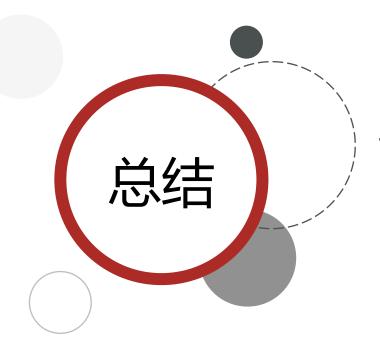
名称	说明
<pre>public static <t> Stream<t> stream(T[] array)</t></t></pre>	获取当前数组的Stream流
<pre>public static<t> Stream<t> of(T values)</t></t></pre>	获取当前数组/可变数据的Stream流



## Stream流的三类方法

- 获取Stream流
  - 创建一条流水线,并把数据放到流水线上准备进行操作
- 中间方法
  - 流水线上的操作。一次操作完毕之后,还可以继续进行其他操作。
- 终结方法
  - 一个Stream流只能有一个终结方法,是流水线上的最后一个操作





- 1、集合获取Stream流的方式?
  - 集合获取Stream的方式是通过调用stream()方法实现的。
- 2、数组获取Stream流的方式?

名称	说明
<pre>public static <t> Stream<t> stream(T[] array)</t></t></pre>	获取当前数组的Stream流
<pre>public static<t> Stream<t> of(T values)</t></t></pre>	获取当前数组/可变数据的Stream流



- **)** 创建不可变集合
- Stream流
  - ◆ Stream流的概述
  - ◆ Stream流的获取
  - ◆ Stream流的常用方法
  - ◆ Stream流的综合应用
  - ◆ 收集Stream流
- > 异常处理



# Stream流的常用API(中间操作方法)

名称	说明
<pre>Stream<t> filter(Predicate<? super T> predicate)</t></pre>	用于对流中的数据进行 <mark>过滤。</mark>
<pre>Stream<t> limit (long maxSize)</t></pre>	获取前几个元素
Stream <t> skip (long n)</t>	跳过前几个元素
<pre>Stream<t> distinct ()</t></pre>	去除流中重复的元素。依赖(hashCode和equals方法)
<pre>static <t> Stream<t> concat (Stream a, Stream b)</t></t></pre>	合并a和b两个流为一个流

#### 注意:

- 中间方法也称为非终结方法,调用完成后返回新的Stream流可以继续使用,支持链式编程。
- 在Stream流中无法直接修改集合、数组中的数据。



### Stream流的常见终结操作方法

名称	说明
void forEach (Consumer action)	对此流的每个元素执行遍历操作
long count ()	返回此流中的元素数

注意:终结操作方法,调用完成后流就无法继续使用了,原因是不会返回Stream了。





- 1、终结和非终结方法的含义是什么?
  - 终结方法后流不可以继续使用,非终结方法会返回新的流,支持链式编程。



- **)** 创建不可变集合
- > Stream流
  - ◆ Stream流的概述
  - ◆ Stream流的获取
  - ◆ Stream流的常用API
  - ◆ Stream流的综合应用
  - ◆ 收集Stream流
- > 异常处理



# **富**案例

# 案例标题



需求:某个公司的开发部门,分为开发一部和二部,现在需要进行年中数据结算。

#### 分析:

① : 员工信息至少包含了(名称、性别、工资、奖金、处罚记录)

② : 开发一部有4个员工、开发二部有5名员工

③ :分别筛选出2个部门的最高工资的员工信息,封装成优秀员工对象Topperformer

④ :分别统计出2个部门的平均月收入,要求去掉最高和最低工资。

⑤ :统计2个开发部门整体的平均工资,去掉最低和最高工资的平均值。



- **)** 创建不可变集合
- > Stream流
  - ◆ Stream流的概述
  - ◆ Stream流的获取
  - ◆ Stream流的常用API
  - ◆ Stream流的综合应用
  - ◆ 收集Stream流
- > 异常处理



# Stream流的收集操作

● **收集Stream流的含义**:就是把Stream流操作后的结果数据转回到集合或者数组中去。

● Stream流:方便操作集合/数组的**手段。** 

● 集合/数组:才是开发中的**目的。** 



# Stream流的收集方法

名称	说明
R collect (Collector collector)	开始收集Stream流,指定收集器

#### Collectors工具类提供了具体的收集方式

名称	说明
<pre>public static <t> Collector toList ()</t></pre>	把元素收集到List集合中
<pre>public static <t> Collector toSet ()</t></pre>	把元素收集到Set集合中
<pre>public static Collector toMap (Function keyMapper , Function valueMapper)</pre>	把元素收集到Map集合中





1、收集Stream流的作用 ?

- Stream流是操作集合/数组的手段
- 操作的结果数据最终要恢复到集合或者数组中去。







传智教育旗下高端IT教育品牌