Java 核心技术(高阶)

第八章 Java Stream

第一节 Stream的概述

华东师范大学 陈良育

例子(1)

- 给定一个字符串数组，统计长度大于5的元素个数

```
String[] planets = new String[] {
            "Mercury", "Venus", "Earth",
            "Mars", "Jupiter", "Saturn",
            "Uranus", "Neptune" };

//查找星球名字大于等于5个字母
long count = 0;
for(String p : planets) {
    if(p.length() > 5) {
        count ++;
    }
}
```

例子(2)

- 采用流(Stream)的办法，统计长度大于5的元素个数

```
List<String> pList = Arrays.asList(planets);

//采用流方法
count = pList.stream()
        .filter(p->p.length()>5).count();

//采用并行流方法
count = pList.parallelStream()
        .filter(p->p.length()>5).count();
```

Stream(1)

- Stream流
  - a sequence of elements from source that supports aggregate operations
  - sequence of elements: 一个流对外提供一个接口，可以访问到一串特定的数据。流不存储元素，但是可以根据需要进行计算转化
  - source：数据来源，如数据结构，数组，文件等
  - aggregate operation：聚合操作，流支持像SQL操作或者其他函数式语言的操作，如filter/map/reduce/find/match/sorted等。

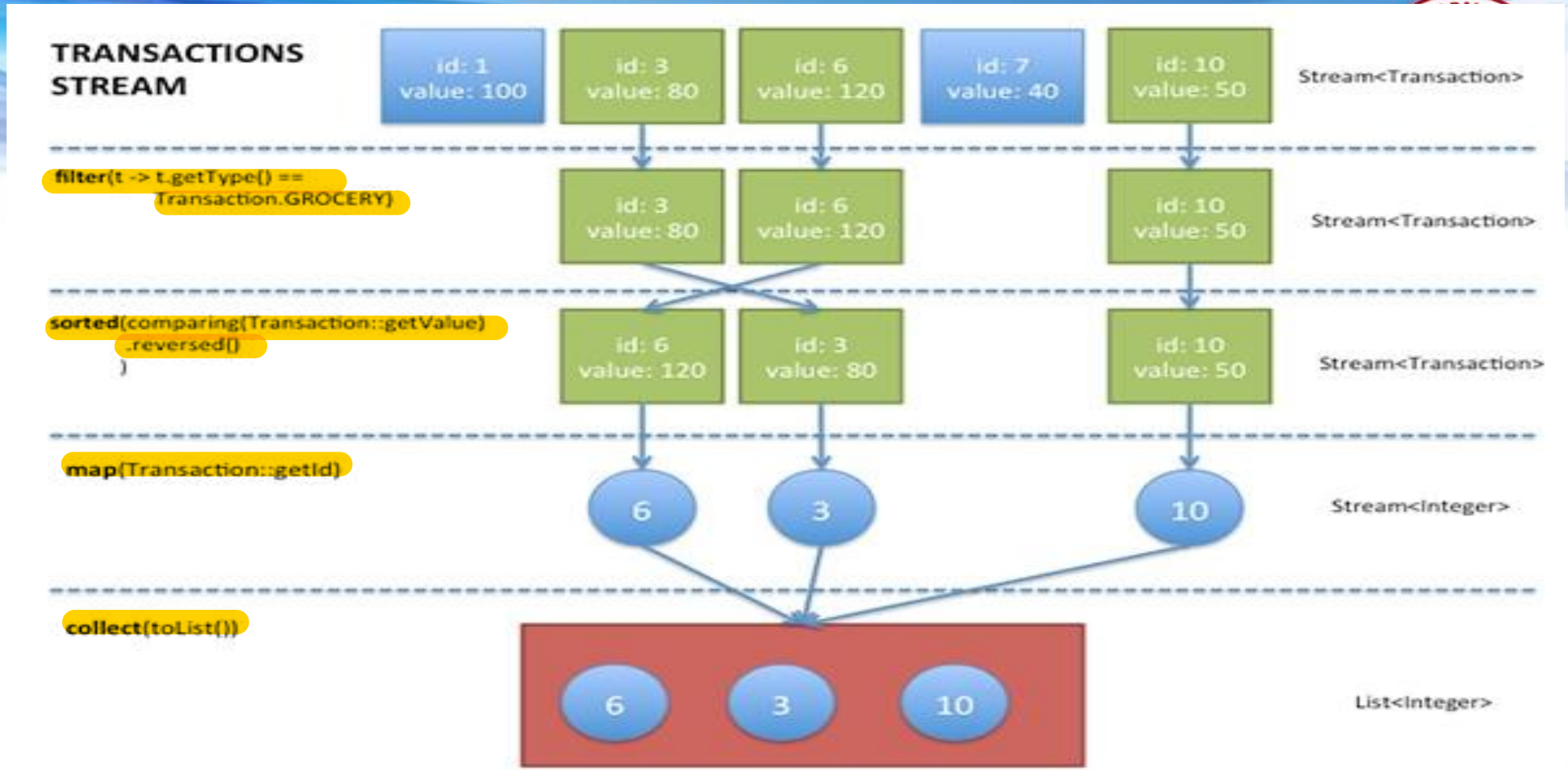
Stream(2)

- Stream流
  - pipelining: 很多流操作也是返回一个流
  - Internal Iteration: 流操作进行迭代，用户感知不到循环遍历

```
List<String> pList = Arrays.asList(planets);

//采用流方法
count = pList.stream()
        .filter(p->p.length()>5).count();

//采用并行流方法
count = pList.parallelStream()
        .filter(p->p.length()>5).count();
```



- https://www.oracle.com/technical-resources/articles/java/ma14-java-se-8-streams.html

Stream(4)

- Stream语法
  - 类似SQL语句，遵循“做什么而非怎么做”原则

```
transactions.stream().filter(t->t.getType().equals("grocery"))
    .sorted(Comparator.comparing(Transaction::getValue).reversed())
    .map(Transaction::getId)
    .collect(Collectors.toList())
    .forEach(System.out::println);
```

Stream(5)

- 流的工作流程
  - 流的创建
  - 流的转换，将流转换为其他流的中间操作，可包括多个步骤(惰性操作)
  - 流的计算结果。这个操作会强制执行之前的惰性操作。这个步骤以后，流就再也不用了。

总结

- 了解流的作用
- 了解流的特点
- 了解流的计算流程

谢 谢！