



Java 核心技术(高阶)

第八章 Java Stream

第二节 Stream的创建

华东师范大学 陈良育



Stream的工作流程

- 流的工作流程

- 创建一个流

- 指定将流转换为其他流的中间操作，可包括多个步骤(惰性操作)

- 应用终止操作，从而产生结果。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。



Stream创建(1)

- Collection接口的stream方法

```
Stream<String> as = new ArrayList<String>().stream();  
Stream<String> hs = new HashSet<String>().stream();
```

– 还有其他的子类，如LinkedList, LinkedSet, TreeSet, Vector等

- Arrays.stream可以将数组转为Stream

```
Stream<String> b1 = Arrays.stream("a,b,c,d,e".split(","), 3, 5);
```



Stream创建(2)

- 利用Stream类进行转化

- of 方法, 直接将数组转化

```
Stream<Integer> c1 = Stream.of(new Integer[5]);  
Stream<String> c2 = Stream.of("a,b,c".split(","));  
Stream<String> c3 = Stream.of("a", "b", "c");
```

- empty方法, 产生一个空流

```
Stream<String> d1 = Stream.empty();
```




Stream创建(3)

- 利用Stream类进行转化

- generate 方法，接收一个Lambda表达式

```
Stream<String> e1 = Stream.generate(() -> "hello");  
Stream<Double> e2 = Stream.generate(Math::random);
```

- iterate 方法，接收一个种子，和一个Lambda表达式

```
Stream<BigInteger> e3 = Stream.iterate(BigInteger.ZERO, n -> n.add(BigInteger.ONE));
```



Stream创建(4)

- 基本类型流(只有三种)
 - IntStream, LongStream, DoubleStream

```
IntStream s1 = IntStream.of(1,2,3,4,5);
```

```
s1 = Arrays.stream(new int[] {1,2,3});
```

```
s1 = IntStream.generate(() -> (int) (Math.random() * 100));
```

```
s1 = IntStream.range(1,5); //1,2,3,4 step 1
```

```
s1 = IntStream.rangeClosed(1,5); //1,2,3,4,5
```

```
IntStream s2 = IntStream.of(1,2,3,4,5);
```

```
Stream<Integer> s3 = s2.boxed();
```

```
IntStream s5 = s3.mapToInt(Integer::intValue);
```



Stream创建(5)

- 并行流

- 使得所有的中间转换操作都将被并行化
- `Collections.parallelStream()` 将任何集合转为并行流
- `Stream.parallel()` 方法, 产生一个并行流

```
IntStream s1 = IntStream.range(1,100000000);  
long evenNum = s1.parallel().filter(n->n%2==0).count();  
System.out.println(evenNum);
```

- 注意: 需要保证传给并行流的操作不存在竞争



Stream创建(6)

- 其他类/方法产生Stream流

- Files.lines 方法

```
Stream<String> contents = Files.lines(Paths.get("C:/abc.txt"));
```

- Pattern的splitAsStream 方法

```
Stream<String> words = Pattern.compile(",").splitAsStream("a,b,c");
```


总结



- 流的内容来源
 - Collection的子类
 - 数组
 - Lambda表达式产生
 - 文件
 - 其他的方法。。。



谢谢!