Streams

Putting The Pipeline Together

```
// example: for given list of names get first two names which are
// four characters long, in an alphabetical order
// the old way:

var list = List.of("John", "George", "Paul", "Ringo");
List<String> filtered = new ArrayList<>();
for (String name: list)
  if (name.length() == 4) filtered.add(name);
Collections.sort(filtered);
var iter = filtered.iterator();
if (iter.hasNext()) System.out.println(iter.next());
if (iter.hasNext()) System.out.println(iter.next());
```

```
John
Paul
```

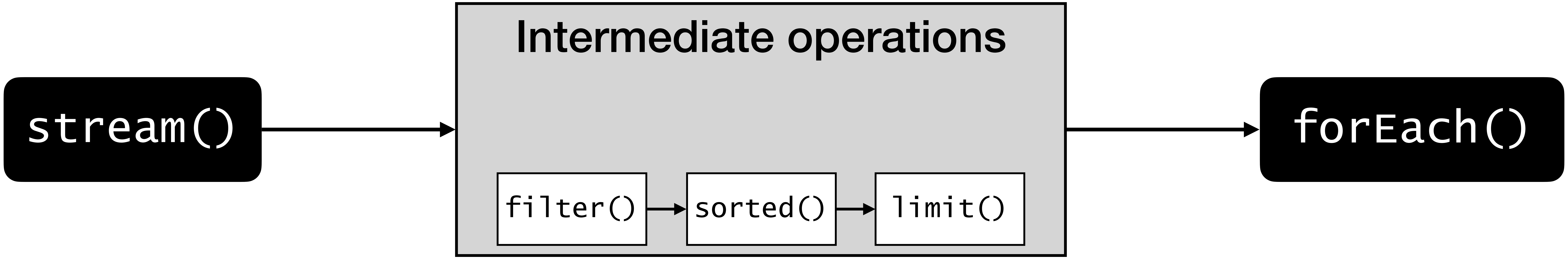
```
// same thing, but using streams


var list = List.of("John", "George", "Paul", "Ringo");
list.stream()
    .filter(n -> n.length() == 4)
    .sorted()
    .limit(2)
    .forEach(System.out::println);
```

source

intermediate operations

terminal operation

```
John
Paul
```



```
// you can chain more pipelines together


long count = Stream.of("John", "George", "Paul", "Ringo")   source
  .filter(s -> s.length() > 4)    intermediate operation
  .collect(Collectors.toList())   terminal operation
  .stream()    source
  .count();    terminal operation


System.out.println(count);

  => 2
```