Streams

Stream Spliterator

How Spliterator Works

- you start with some kind of collection with n elements
  - imagine a box containing number of toys
- now you want to split this collection in two (or more collections)
  - imagine moving some toys from the original box to the new box
- in order to do this you have to create an object of type `Spliterator`
- once you have `Spliterator` object, you can apply some common methods...

Common Spliterator Methods

Method	What it does
`Spliterator<T> trySplit()`	Returns `Spliterator` containing about half of the data, which is then removed from the original `Spliterator` (if data is no longer splittable it returns `null`)
`void forEachRemaining(` `  Consumer<T> c)`	Processes remaining elements in `Spliterator`
`boolean tryAdvance(` `  Consumer<T> c)`	Processes single element from Spliterator (if exits), returns if the element is processed

```
// example #1
List<String> list = Arrays.asList("One", "Two", "Three", "Four", "Five");
Stream<String> stream = list.stream();

Spliterator<String> originalSpliterator = stream.spliterator();
Spliterator<String> newSpliterator = originalSpliterator.trySplit();

newSpliterator.forEachRemaining(System.out::println);
System.out.println("---");
originalSpliterator.forEachRemaining(System.out::println);
```

```
One
Two
---
Three
Four
Five
```

NB. Once you apply `forEachRemaining()` method on a Spliterator, all elements are processed and Spliterator is now empty, so if you run this command again:

`originalSpliterator.forEachRemaining(System.out::println);` it will return nothing

newSpliterator originalSpliterator

```
// example #2
List<String> list = Arrays.asList("One", "Two", "Three", "Four", "Five");
Stream<String> stream = list.stream();
Spliterator<String> originalSpliterator = stream.spliterator();
Spliterator<String> newSpliterator = originalSpliterator.trySplit();
newSpliterator.tryAdvance(System.out::println);
System.out.println("---");
newSpliterator.forEachRemaining(System.out::println);
System.out.println("---");
originalSpliterator.tryAdvance(System.out::println);
System.out.println("---");
originalSpliterator.forEachRemaining(System.out::println);
```

```
One
---
Two
---
Three
---
Four
Five
```