Additional Terminal (Reduction) Operations

Reduction operations combine the contents of a stream, to return a value, or they can return a collection.

On this slide, I want to show you some additional terminal operations, and their return types and signatures.

Return Type	Terminal Operations

R
R
	collect(Collector<? superT,A,R> collector)
	collect(Supplier<R> supplier,BiConsumer<R,? superT> accumulator,BiConsumer<R,R> combiner)

Optional<T>
T
<U> U
	reduce(BinaryOperator<T> accumulator)
	reduce(T identity,BinaryOperator<T> accumulator)
	reduce(U identity,BiFunction<U,? superT,U> accumulator,BinaryOperator<U> combiner)

Object[]
A[]
	toArray()
	toArray(IntFunction<A[]> generator)

List<T>	toList()

Revised View of Transformative Operations

There is one interface I haven't talked about yet, and that's the Collector.

This is not a functional interface, but there are helper methods on another class, named Collectors that provide these special types.

Return Type	Terminal Operations
R	collect(Collector collector)
R	collect(Supplier supplier,BiConsumer accumulator,BiConsumer combiner)
Optional	reduce(BinaryOperator accumulator)
T	reduce(T identity,BinaryOperator accumulator)
<U> U	reduce(U identity,BiFunction accumulator,BinaryOperator combiner)
Object[]	toArray()
A[]	toArray(IntFunction generator)
List	toList()