Workshop on Java 8 New Features

Pravin Jain

Zen Softech Private Limited

November 18, 2020 - November 21, 2020
Day - 4
Advanced Usage of Stream

Outline I

Advanced Usage of Stream

- Understanding the reduce method of `Stream<T>`
- Understanding the collect method and the Collector
- Using the Collectors class to create common Collector

New Data and Time API (`java.time` package)

- Various elements from the `java.time` package

General Instructions

There is an accompanying source code for this workshop. There are various exercises mentioned as TODO comments in the source code.

Outline

Understanding the reduce method of `Stream<T>`

- `T reduce(T identity, BinaryOperator<T> accumulator)`

Understanding the reduce method of `Stream<T>`

- `T reduce(T identity, BinaryOperator<T> accumulator)`
- `Optional<T> reduce(BinaryOperator<T> accumulator)`

Understanding the reduce method of `Stream<T>`

- `T reduce(T identity, BinaryOperator<T> accumulator)`
- `Optional<T> reduce(BinaryOperator<T> accumulator)`
- `<U> U reduce(U identity, BiFunction<U,T,U> accumulator, BinaryOperator<U> combiner)`

Outline

Understanding the collect method of `Stream<T>`

- `<R> R collect(Supplier<R> supplier, BiConsumer<R,T> accumulator, BiConsumer<R,R> combiner)`

Understanding the collect method of `Stream<T>`

- `<R> R collect(Supplier<R> supplier, BiConsumer<R,T> accumulator, BiConsumer<R,R> combiner)`
- `<R> R collect(Collector<T,A,R> collector)`

Understanding the `Collector<T,A,R>` interface

has static method to create instance

- `static <T,A,R> Collector<T,A,R> of(Supplier<A> supplier, BiConsumer<A,T> accummulator, BinaryOperator<A> combiner, Function<A,R> finisher, Collector.Characteristics... characteristics)`
- `static <T,R> Collector<T,R,R> of(Supplier<A> supplier, BiConsumer<A,T> accummulator, BinaryOperator<A> combiner, Collector.Characteristics... characteristics)`

Understanding the `Collector<T,A,R>` interface - continued

- `Supplier<A> supplier()`
- `BiConsumer<A,T> accumulator()`
- `BinaryOperator<A> combiner()`
- `Function<A,R> finisher()`
- `Set<Collector.Characteristics> characteristics()`

Outline

Advanced Usage of Stream

- Understanding the reduce method of `Stream<T>`
- Understanding the collect method and the Collector
- Using the Collectors class to create common Collector

New Data and Time API (`java.time` package)

- Various elements from the `java.time` package

Creating `Collector` using `Collectors` class

static methods of the `Collectors` class

- `static <T> Collector<T,?,Double> averagingInt(ToIntFunction<T> mapper)`
- `static <T> Collector<T,?,Double> averagingLong(ToLongFunction<T> mapper)`
- `static <T> Collector<T,?,Double> averagingDouble(ToDoubleFunction<T> mapper)`
- `static <T> Collector<T,?,Integer> summingInt(ToIntFunction<T> mapper)`
- `static <T> Collector<T,?,Long> summingLong(ToLongFunction<T> mapper)`
- `static <T> Collector<T,?,Double> summingDouble(ToDoubleFunction<T> mapper)`

Creating `Collector` - continued

- `static <T> Collector<T,?,IntSummaryStatistics> summarizingInt(ToIntFunction<T> mapper)`
- `static <T> Collector<T,?,LongSummaryStatistics> summarizingLong(ToLongFunction<T> m)`
- `static <T> Collector<T,?,DoubleSummaryStatistics> summarizingDouble(ToDoubleFunction<T> m)`
- `static <T> Collector<T,?,Long> counting()`
- `static <T> Collector<T,?,Optional<T>> minBy(Comparator<T> c)`
- `static <T> Collector<T,?,Optional<T>> maxBy(Comparator<T> c)`

Creating `Collector` - continued

- `static <T,A,R,RR> Collector<T,A,RR> collectingAndThen(Collector<T,A,R> downstream, Function<R,RR> finisher)`
- `static <T,A,R> Collector<T,?,R> mapping(Function<T,U> Collector<U,A,R> downstream)`
- `static <T,U,A,R> Collector<T,?,R> flatMapping(Function<T,Stream<U>> Collector<U,A,R> downstream)` since Java 9
- `static <T,A,R> Collector<T,?,R> filtering(Predicate<T> predicate, Collector<T,A,R> downstream)` since Java 9

Creating `Collector` - continued

- `static <T> Collector<T,?,T> reducing(T identity, BinaryOperator<T> op)`
- `static <T> Collector<T,?,Optional<T>> reducing(BinaryOperator<T> op)`
- `static <T,U> Collector<T,?,U> reducing(U identity, Function<T,U> mapper, BinaryOperator<U> op)`
- `static <T,R1,R2,R> Collector<T,?,R> teeing(Collector<T,?,R1> downstream1, Collector<T,?,R2> downstream2, BiFunction<R1,R2,R> merger)` since Java 12

Creating `Collector` - continued

- `static <T,K> Collector<T,?,Map<K,List<T>>> groupingBy(Function<T,K> classifier)`
- `static <T,K,A,D> Collector<T,?,Map<K,D>> groupingBy(Function<T,K> classifier, Collector<T,A,D> downstream)`
- `static <T,K,A,D,M extends Map<K,D>> Collector<T,?,M> groupingBy(Function<T,K> classifier, Supplier<M> mapFactory, Collector<K,A,D> downstream)`

Creating `Collector` - continued

- `static <T,K> Collector<T,?,ConcurrentMap<K,List<T>>> groupingByConcurrent(Function<T,K> classifier)`
- `static <T,K,A,D> Collector<T,?,ConcurrentMap<K,D>> groupingByConcurrent(Function<T,K> classifier, Collector<T,A,D> downstream)`
- `static <T,K,A,D,M extends ConcurrentMap<K,D>> Collector<T,?,M> groupingByConcurrent(Function<T,K> classifier, Supplier<M> mapFactory, Collector<K,A,D> downstream)`

Creating `Collector` - continued

- `static <T,K> Collector<T,?,Map<Boolean,List<T>> partitioningBy(Predicate<T> predicate)`
- `static <T,K,A,D> Collector<T,?,Map<Boolean,D> partitioningBy(Predicate<T> predicate, Collector<T,A,D> downstream)`

Creating `Collector` - continued

- `static <T,K> Collector<T,?,Map<Boolean,List<T>> partitioningBy(Predicate<T> predicate)`
- `static <T,K,A,D> Collector<T,?,Map<Boolean,D> partitioningBy(Predicate<T> predicate, Collector<T,A,D> downstream)`
- `static Collector<CharSequence,?,String> joining()`
- `static Collector<CharSequence,?,String> joining(CharSequence delim)`
- `static Collector<CharSequence,?,String> joining(CharSequence delim, CharSequence pref, CharSequence suff)`

Creating `Collector` - continued

- `static <T,C extends Collection<T>> Collector<T,?,C> toCollection(Supplier<C> collectionFactory)`
- `static <T> Collector<T,?,Set<T>> toSet()`
- `static <T> Collector<T,?,Set<T>> toUnmodifiableSet()` since Java 10
- `static <T> Collector<T,?,List<T>> toList()`
- `static <T> Collector<T,?,List<T>> toUnmodifiableList()` since Java 10

Creating `Collector` - continued

- `static <T,K,U> Collector<T,?,Map<K,U>> toMap(Function<T,K> keyMapper, Function<T,U> valueMapper)`
- `static <T,K,U> Collector<T,?,Map<K,U>> toMap(Function<T,K> keyMapper, Function<T,U> valueMapper, BinaryOperator<U> mergeFunction)`
- `static <T,K,U,M extends Map<K,U>> Collector<T,?,M> toMap(Function<T,K> keyMapper, Function<T,U> valueMapper, BinaryOperator<U> mergeFunction, Supplier<M> mapFactory)`

Creating `Collector` - continued

- `static <T,K,U> Collector<T,?,ConcurrentMap<K,U>> toConcurrentMap(Function<T,K> keyMapper, Function<T,U> valueMapper)`
- `static <T,K,U> Collector<T,?,ConcurrentMap<K,U>> toConcurrentMap(Function<T,K> keyMapper, Function<T,U> valueMapper, BinaryOperator<U> mergeFunction)`
- `static <T,K,U,M extends ConcurrentMap<K,U>> Collector<T,?,M> toConcurrentMap(Function<T,K> keyMapper, Function<T,U> valueMapper, BinaryOperator<U> mergeFunction, Supplier<M> mapFactory)`

Creating `Collector` - continued

- `static <T,K,U> Collector<T,?,Map<K,U>> toUnmodifiableMap(Function<T,K> keyMapper, Function<T,U> valueMapper)` since Java 10
- `static <T,K,U> Collector<T,?,Map<K,U>> toUnmodifiableMap(Function<T,K> keyMapper, Function<T,U> valueMapper, BinaryOperator<U> mergeFunction)` since Java 10

Outline

New Data and Time API (`java.time` package)

- `Clock` and `Instant`

New Data and Time API (`java.time` package)

- `Clock` and `Instant`
- `Duration` and `Period`

New Data and Time API (`java.time` package)

- `Clock` and `Instant`
- `Duration` and `Period`
- `LocalDate`, `LocalTime` and `LocalDateTime`

New Data and Time API (`java.time` package)

- `Clock` and `Instant`
- `Duration` and `Period`
- `LocalDate`, `LocalTime` and `LocalDateTime`
- `ZonedDateTime`, `OffsetTime`, `OffsetDateTime`

New Data and Time API (`java.time` package)

- `Clock` and `Instant`
- `Duration` and `Period`
- `LocalDate`, `LocalTime` and `LocalDateTime`
- `ZonedDateTime`, `OffsetTime`, `OffsetDateTime`Partial dates
- `MonthDay`, `YearMonth`, `Year`

New Data and Time API (`java.time` package)

- `Clock` and `Instant`
- `Duration` and `Period`
- `LocalDate`, `LocalTime` and `LocalDateTime`
- `ZonedDateTime`, `OffsetTime`, `OffsetDateTime`Partial dates
- `MonthDay`, `YearMonth`, `Year`

enums

- `DayOfWeek`
- `Month`

Exercise

Update the `Account.Transaction` class to use `LocalDate` class instead of the `long` type for the transaction date.