

Lambdas

<code>java.util.function</code>	Package
<code>@FunctionalInterface</code>	Optional. Force a compiler error if the interface has more than one abstract method.
<code>(String text) -> System.out.println(text)</code> <code>(text) -> System.out.println(text)</code> <code>System.out::println</code>	Anonymous function.
Default/static method in interface	

Stream API - Generatoren

Iterate / Range	Automatische Erzeugung von IntStream durch Iteration
Builder	Aufbau eines Streams mit Elementen ohne Collection
Of	Sammlung von Objekten als Argument
Generate	Supplier Lambda zur Generierung von Einträgen
Files (NIO), ZipFile, JarFile, Patterns (Regex)	

Stream API - Transformation

<code>map(lambda)</code>	Objekte von Eingabetyp in Ausgabotyp umwandeln
<code>flatMap(lambda)</code>	Collection-Elemente in Stream-Elemente begradigen (<code>Array[][] -> Array[]</code>)

Stream API - Combining / Shortening

<code>.filter(p -> p.name.startsWith("M"))</code>	
<code>.reduce((x1, x2) -> x1 + x2)</code>	<code>y = ((x1 + x2) + x3) + ...</code>
<code>.collect(Collectors.joining(", "));</code>	<code>e1 + ", " + e2 + ...</code>
<code>.limit(5)</code>	limit number of results
<code>.skip(5)</code>	skip elements

Stream API - Aggregation

Those APIs consider the whole set	
<code>.sorted()</code>	
<code>.min()/max()</code>	
<code>.groupingBy()</code>	Gruppieren gleicher Elemente
<code>.distinct()</code>	make list unique based on lambda
<code>.count()</code>	

Stream API - Ergebnisse

<code>forEach / forEachOrdered</code>	
<code>toArray(String[]::new)</code>	
<code>collect(Collectors.toList());</code>	
<code>collect(Collectors.toCollection(ArrayList::new));</code>	

Stream API

```
List<Person> persons1, persons2;
Stream.of(persons1, persons2) // stream of lists
    .flatMap(ps -> ps.stream()) // stream of
    obj
    .map(p -> p.firstName + " " + p.name) //
    stream of name
    .forEach(System.out::println);
IntStream.iterate(0, i -> i + 1)
    .limit(10)
    .forEach(System.out::print);
Stream.builder()
    .add("Wir ").add(" bauen ").add(" -
    einen Satz.").build()
    .forEachOrdered(...);
Stream.generate(() -> Double.valueOf(Math.
    random() * 100).intValue())
    .limit(5).forEach(...)
```

Java I/O, NIO

```
Files.lines(Paths.get(this.class.getResource("file.txt")))
    .count();
```

Date/Time API

LocalTime

LocalDate

LocalDateTime

ZonedDateTime.of() / .now(ZonedDateTime.of("CET"));

Instant Maschinenzeit, genauer, ohne Zeitzone

TemporalAdjuster

Duration

DateTimeFormatter.ISO_OFFSET_DATE_TIME

// 2011-12-03T10:15:30+01:00

DateTimeFormatter.ofLocalizedDate(FormatStyle.LONG)

// June 20, 2014; 10:15:58 GMT+1

Optional

Optional<String> result = doSomething();

if (result.isPresent()) result.get();

String result = doSomething().orElse("unknown");

doSomething().ifPresent(lambda);

Repeatable Annotation

```
@Repeatable(TableAnnos.class)
@Retention(RetentionPolicy.RUNTIME)
public @interface Table {
    String value();
}

@Retention(RetentionPolicy.RUNTIME)
public @interface TableAnnos {
    Table[] value();
}

@Table("Customer")
@Table("Employee")
public class Person {
}
```



By marb

cheatography.com/marb/

Not published yet.

Last updated 9th December, 2016.

Page 2 of 2.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>