Cheatography

Java 8 Cheat Sheet

by gnowakow via cheatography.com/36666/cs/12335/

Lambda

Optional type declaration	The compiler can infer the type
Optional parenthesis around parameter	For single parameter, no need for parens. Required for multiple parameters
Optional curly braces	If it contains a single statement
Optional return keyword	Compiler automatically returns the value.

Facilitates functional programming.

Method references

Pass a method as a function parameter

```
public class Java8Tester {
public static void main(String
args[]){
List names = new ArrayList();
names.add("George");
names.add("Nowakowski");
names.forEach(System.out::println);
}
}
```

Optional Class

Optional.ofNullable()	Returns a non-empty Optional if a value present in the given object. Otherwise returns an empty Optiona
Optional.empty()	empty Optional object
Optional.isPresent()	Returns true if the given Optional object is non-empty
Optiona.ifPresent()	Performs the given action if the Optional object is non-empty

Optional Class (cont)

Optional .orElse( )	Returns the value if present. Otherwise returns the given default value

Optional is a container object which is used to represent a value is present or absent.

Advantages:
1. Null checks are not required.
2. No more NPE at run-time
3. Encourages clean and neat APIs
4. No more Boiler plate code

Streams

Sequence of elements	A stream provides a set of elements of specific type in a sequential manner.
Source	Collections, Arrays, or I/O resources
Aggregate operations	filter, map, limit, reduce, find, match
Pipelining	Most stream operations return a stream, so their result can be pipelined.
Automatic iterations	Stream operations do the iterations internally
stream()	Returns a sequential stream considereing collection as its source
parallelStream()	Returns a parallel stream
map	maps each element to its corresponding result
filter	used to elminiate elements based ona criteria.
limit	used to reduce the size of the stream
sorted	

Streams (cont)

Collectors	Used to combine the result of processing on the elements of a stream. Can return a list or a string

Stream represents a sequence of objects from a source, which supports aggregate functions. The concept of stream that lets the developer to process data declaratively, rather than having to write loops.

Method References

Help to point to methods by their names. A method reference is described using :: symbol. A method reference can be used to point to the following types of methods:

- Static methods
- Instance methods
- Constructors using new operator (TreeSet::new)

Nashorn JavaScript

Usage:

```
ScriptEngine engine =
new
ScriptEngineManager().getEngineByNa
me("nashorn");
engine.eval("print('Hello World!
');");
```

Or, you can read it in using a FileReader() then call functions:

```
engine.eval(new
FileReader("testFun.js")
Invocable invocable = (Invocable)
engine;
Object result =
invocable.invokeFunction("fun1",
"Peter Parker");
```