Java 8 - API

Lukáš Křečan
GoodData

Questions?

Agenda

- Language changes
  - Method references
  - Default methods on interfaces
- API changes
  - Streams
    - Parallelism
  - CompletableFutures

- Coding

More declarative, less imperative

Tell what to do,
not how

Method references

- Clever matching of method signature

- Reference to a static method
- Reference to an instance method of a particular object
- Reference to an instance method of an arbitrary object of a particular type
- Reference to a constructor

Default methods

- Lot’s of new methods - Map, ...

- Single inheritance
- Refuses to compile if not clear

Streams

- Create
- Intermediate
- Terminal

Streams - creation

- From a Collection or Array
- By hand
- From BufferedReader.lines();
- File paths from methods in Files
- Random numbers from Random.ints()
- Others BitSet.stream(), Pattern.splitAsStream(java.lang.CharSequence), and JarFile.stream().
- May be infinite

Streams - intermediate ops

- Stateless
  - map / flatMap
  - filter
  - peek
  - ...
- Stateful
  - sorted
  - limit
  - skip
  - ...
- Lazy
  - Create a new stream

Streams - terminal operation

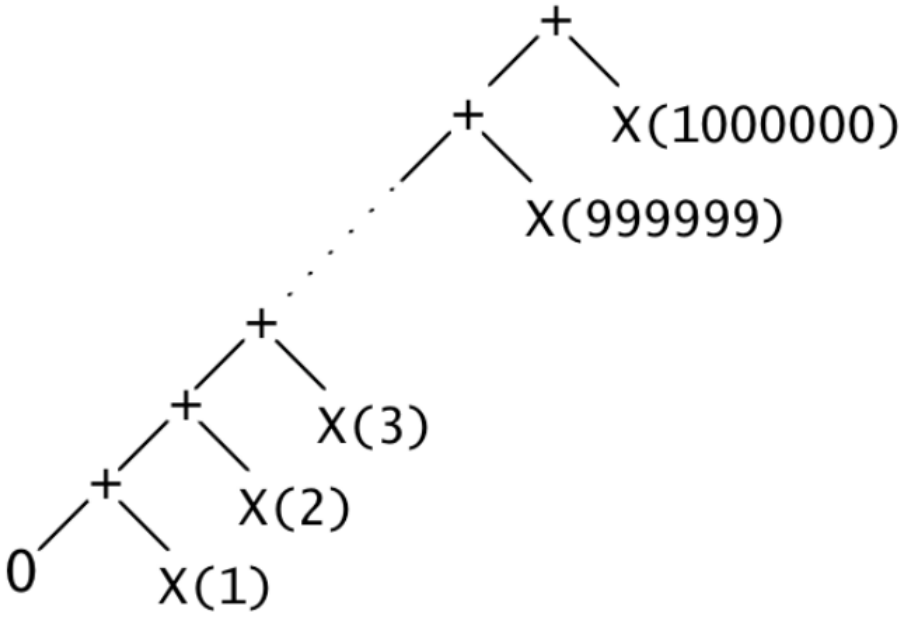
- Does all the work

- sum, max, avg, count
- findFirst
- noneMatch, allMatch
- forEach
- reduce / collect

Sequential processing

Sequential Computation Tree

```
SUM = 0
DO I = 1, 1000000
  SUM = SUM + X(I)
END DO
```



http://strangeloop2010.com/talk/presentation_file/14299/GuySteele-parallel.pdf

Collector

- How to supply initial value - supplier
- How to add a new value - accumulator
- How to combine two results - combiner

Terminal operations

- Initiate the execution
- Reducing / collecting

Do not use parallel streams in multithreaded environment!

Completable Futures

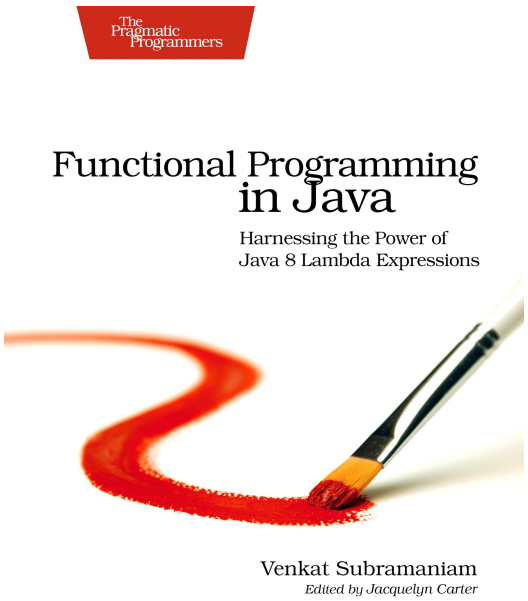
- Not in libraries yet
- Maybe too late

Summary

- Little big change in the language
- Will have to change our approach
- Libraries should change as well

Sources

- JavaDoc + Sources
- Devoxx keynote
- Java Parallel Calamity
- How to Think about Parallel Programming: Not!
- https://github.com/lukas-krecan/java8-playground



java: no suitable method found for collect(java.util.stream.Collector<java.lang.Object,capture#1 of ?,java.util.Map<java.lang.Object,java.util.List<java.lang.Object>>>,java.util.stream.Collector<java.lang.Object,capture#2 of ?,java.lang.Long>)

method java.util.stream.Stream.<R>collect(java.util.function.Supplier<R>,java.util.function.BiConsumer<R,? super java.lang.String>,java.util.function.BiConsumer<R,R>) is not applicable

(cannot infer type-variable(s) R

(actual and formal argument lists differ in length))

method java.util.stream.Stream.<R,A>collect(java.util.stream.Collector<? super java.lang.String,A,R>) is not applicable

(cannot infer type-variable(s) R,A

(actual and formal argument lists differ in length))