CIKLUM | SPEAKER'S CORNER

October, 10

17:00 (CET)

English

Java and Scala:

Exploring Syntax and Semantics Through Practical Examples

Tomáš Takáč

Senior Scala Developer at Ciklum

Experiences of tomorrow. Engineered together.

CIKLUM

We transform how people experience the business. All through next generation technology.

What we do:

Product Engineering	Intelligent Automation	Data & Analytics

2002 founded	4000+ professionals
20+ offices	300+ clients

Leading companies choose us:

JUST EAT Takeaway.com · METRO MARKETS · etoro · ZURICH · pay · Greens · KANTAR RETAIL · dacadoo

Meet the speaker

- Technologies: Scala, Akka, Apache Spark, PostgreSQL, MongoDB
- Projects: carnival cruising ships, Cloudfarms, Peek and Cloppenburg loyalty system
- Areas of interest: clarity of code, microservices, functional programming

Tomáš Takáč
Senior Scala Developer
at Ciklum

October 2024

Java and Scala: Exploring Syntax and Semantics Through Practical Examples

Tomáš Takáč
Java consultation: Eva Kubranská

CIKLUM

Agenda

CIKLUM

High level comparison

CIKLUM

High level comparison

CIKLUM

Java and Scala: common grounds

- JVM
  - Compiled into bytecode
  - Interpreted languages
- OOP
- Static typing
  - Types of values are known during compilation

Differences

	Java	Scala version 2.12
History	• June 1991 • Sun Microsystems • Language for electronics ◦ smart TV, set-top boxes • 1995 officially	• 2001 • École Polytechnique Fédérale de Lausanne • Martin Odersky • 20.1.2004 officially
Paradigm	• OOP	• OOP + Functional paradigm

Scala syntax 101

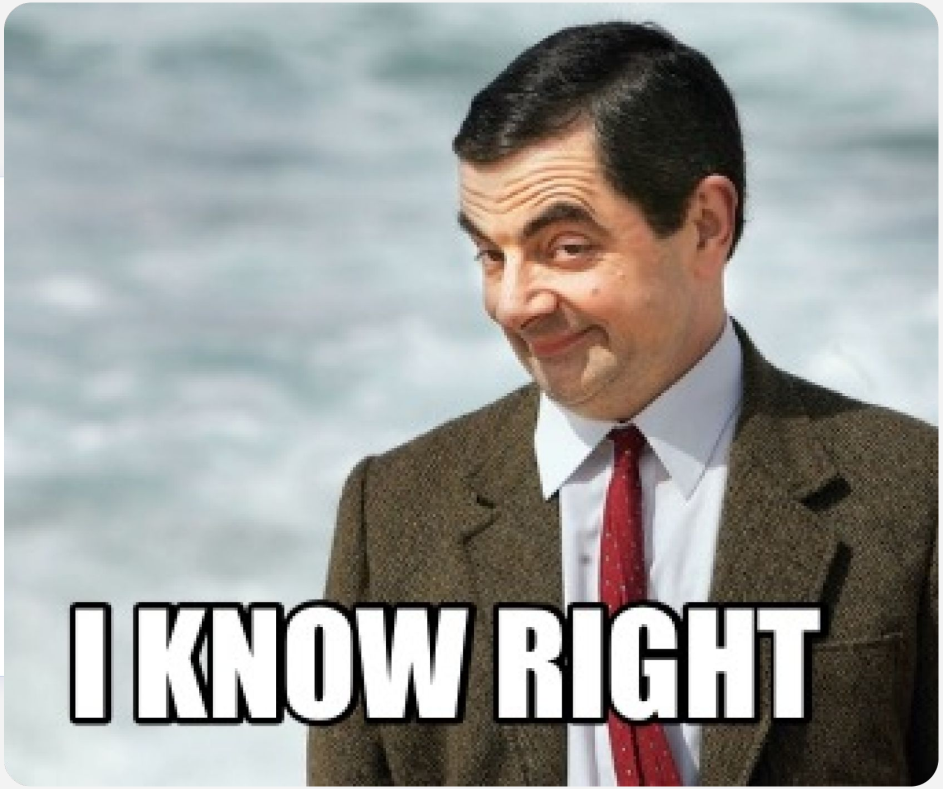
- class
  - in Java: class
- object
  - singleton
  - when named as class, place for “static” methods
- trait
  - in Java: interface
- case class
  - in Java: records
    - toString
    - apply -constructor
    - getters

CIKLUM

Scala syntax 101

WE DON'T NEED:

- “;”
- “Return”
- null



Scala syntax 101

No types before names of expressions (often optionally behind the expression name)

- var
  - variable - mutable value
  - var foo = “my_mutable_foo”
  - foo = “my_new_foo” - compilable
- val
  - value - immutable value
  - val bar = “my_immutable_bar”
  - bar = “my_new_bar” - compilation error
- def
  - definition - method
  - def myDef(a: Int) = a + 1

Scala syntax 101

Pattern matching

- Switch statement on steroids
- myVariable match {

```
            case value1 if someFilter(value1) => some filtered result
            case value2 if someOtherFilter(value1) => some other filtered result
            case value3 => some ordinary result
            case _ => some default result
            }

 myInstance match {
            case MyClass(arg1, arg2, _) => function1(arg1, arg2)
            case MyOtherClass(arg1, _, _, _) => function2(arg1)
            case myFirstElement :: restOfSequence => function3(myFirstElement)
            }
```

CIKLUM

Tick-tock
Code o’clock

https://github.com/tomas-takac/JavaLibraryDemo
https://github.com/tomas-takac/ScalaLibraryDemo

CIKLUM

Thank you!

Any questions?

CIKLUM

Share your feedback!

CIKLUM

Join our team