

Course code: **SCALA/J**

Course title: **Object-functional programming in Scala language for Java programmers**

Days: 3

Description:

Course intended for:

Java programmers, who would like to enhance their effectiveness and develop a better code in the new object-functional programming language on the Java platform.

Course objective:

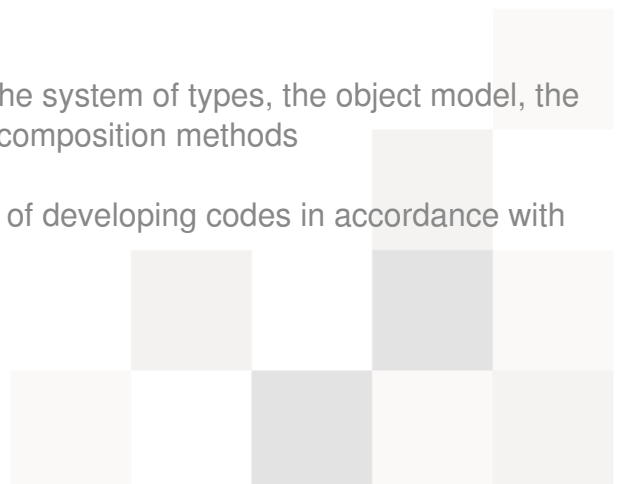
The participants will learn how to use Scala language for very fast development of high quality applications operating on the Java platform.

In particular, the participants are to:

- Get familiar with the Scala language syntax shown against the background of known structures from Java language,
- Get familiar with new language components, not present in Java, including: higher order functions, adjustment of patterns, implicit arguments and conversions, structural types,. Continuations, abstract types, operator overloading.
- During practical exercises, they will learn to use the compiler and IDE to build their first software using the standard Scala libraries, as well as Java libraries
- They will also get familiar with integration of modules written in Java with those developed in Scala.

The following will also be discussed:

1. Control instructions, constant and variable, the system of types, the object model, the error management mechanism, the code decomposition methods
2. The best practices of code design and ways of developing codes in accordance with the functional programming paradigm



Course strengths:

After the training, the participants will be ready to write complex applications in the Scala language and develop the existing Java language applications by components developed in Scala language. Particular emphasis will be placed on learning the best practices of development of high quality codes – coherent, easy to read and maintain.

Requirements:

Java programming skills are required (to be learned at the course J/JP).

Course parameters:

3*8 hours (3*7 net hours) of lectures and workshops (with a visible emphasis on workshops). During the workshops, apart from simple exercises, a simple complex application is designed and implemented.

Group size: no more than 8-10 participants.

Course curriculum:

1. Introduction
2. First steps with Scala language
 - I. The first program
 - II. Interpreter (REPL)
 - III. Compiler
 - IV. IDE
 - V. Source code organization
 - VI. Application building (SBT)
 - VII. Documentation
3. Classes and objects
 - I. Basic types of objects and operations
 - II. Constants, variables, methods



- III. Operator overloading
 - IV. In-built control instructions
 - V. Constructors
 - VI. Equivalence of objects
 - VII. Concrete classes
 - VIII. Inheritance and multi inheritance
 - IX. Abstract properties
 - X. Interfaces
 - XI. Singleton objects
 - XII. Class importing
 - XIII. Visibility range control
 - XIV. Embedded classes
 - XV. Sealed class hierarchies
4. Functions
- I. Functional objects
 - II. Lambda expressions
 - III. Currying and partial function application
 - IV. Function composition
 - V. Partial functions
 - VI. Typical applications of lambda expressions in collections
5. Programming without nulls: Option, Some and None
6. Returning of many objects from function simultaneously: short
7. Pattern matching or what the Java switch cannot do



- I. Simple applications
 - II. Adjustment with embedding
 - III. Defining of own extractors
 - IV. Defining of a partial function on the basis of cases
 - V. Management of exceptions
8. Standard collections
- I. Mutability versus persistence
 - II. Write lines
 - III. Tables, lists, sets and dictionaries
 - IV. Iterators
 - V. Streams
 - VI. Collection generation
 - i. Iterative: a better for loop
 - ii. Recurrent: with use of streams
9. Generic types
- I. Generic methods
 - II. Generic classes
 - III. Specification of type limitations
 - IV. Variance control
 - V. specialization
10. implicit conversions
11. implicit parameters
12. regular expressions



13. working with XML

14. annotations

15. assertions

16. concurrence

I. threads

II. asynchronous exchange of messages

17. Cooperation with Java code

