Programming in Scala

Course period: from 10/11/2014 to 19/12/2014
Meeting schedule: Monday, Wednesday, Friday from 5:00PM to 8:00PM
18 sessions/54hours

One-sentence announcement

Scala – your stairway to superior programming

Description

Scala is a modern hybrid programming language that provides support for object-oriented and functional styles of programming, in addition to support for traditional structured programming, event-based programming, and efficient parallel programming.

Scala is one of the languages that belong to the JVM (Java Virtual Machine) ecosystem. This results in two significant advantages: first, Scala programs run in a mature and highly optimized virtual environment, and second, Scala programs can use a huge codebase found in the Java libraries. Scala code is concise and expressive and can be easily organized into logical modules and seamlessly adapted for parallel processing in multicore and multiprocessor machines. All this makes Scala one of the most comprehensive and most capable programming languages of today.

Because of its support for different programming paradigms and large number of features, Scala is not easy to learn. In order to cope with this problem, this course is based on learning principles introduced in Eckel and Marsh's book "Atomic Scala". This means that various Scala concepts are presented in very short (atomic) lessons, which are followed by illustrative practicing examples and coding problems. This approach turns out to be particularly suitable for fast-paced professional training seminars. In order to make it suitable for attendees with different programming skill levels, including those with very limited experience in software development, all the topics to be covered in this course will be introduced through broad introductory explanations and discussions.

List of topics

Session 1	Introduction to the course and installation of software tools
Session 2	Trying Scala in the REPL environment

Session 3	Data types, values and variables
Session 4	On expressions in general, conditional and compound expressions
Session 5	Classes, objects, packages
Session 6	Vectors and loops
Session 7	Pattern matching
Session 8	Named and default arguments, constructors
Session 9	Functions as objects, map and reduce
Session 10	Pattern matching with types and case classes
Session 11	Defining operators, tuples
Session 12	Inheritance, initialization, overriding
Session 13	Abstract classes, enumerations, traits
Session 14	Polymorphism and composition
Session 15	Sequences, lists and recursion
Session 16	Sets and maps
Session 17	Exceptions, error handling and reporting
Session 18	Software engineering with Scala