Advanced Java Programming

Level: intermediate / advanced

Length: 20 – 35 hours, depending on the chosen subjects

Course objectives

- learn and exercise newer aspects related to Java
- use of object oriented programming to solving practical problems by using Java

What you will learn

- Elements of modern Java, starting with version 8
- Particular ways to implement several design patterns
- Exercise soft skills of communication and presentation

Who can participate: Java programmers who want:

- to update himself/herself with newer elements of the language
- to exercise the object oriented programming and use of Java for solving interesting, more complex problems

Prerequisites: practical experience and knowledge of Java at least at medium level

Required facilities: VGA projector, white board, computers, Java development tools (JDK + IDE). A good, free option is IntelliJ IDEA Community edition.

Related courses: Fundamentals of Java, Object Oriented Analysis and Design, Design Patterns

Description

This course is targeted to Java programmers who want to deep their knowledge about the language and ways to use it.

There are discussed issues and details related to latest enhancements to the language brought by the last versions.

The training is highly interactive, the attendees are implied in discussing the ideas and in designing solutions which are ultimately expressed in Java. The main purpose of this training is to exercise object oriented programming by using Java.

Note: the subjects are adapted to the attendees' profile, their background, experience, goals and time constraints. We can approach other subjects depending on the context.

Examples of topics

1. Time based versioning of Java, distributions
2. Modules
3. Lambda functions, functional interfaces, method references, functions composition, patterns
4. Default and static interface members
5. Streams, operations, parallel streams
6. Optional Class
7. Functional programming, async, composition of sync and async operations
8. CompletableFuture Class
9. Annotations – runtime processing
10. Date & Time API – time zones
11. Generics – advanced topics
12. Concurrent programming; low level (language, Object, Thread, Runnable), high level (Lock, Executor, Thread Pools, concurrent collections), patterns
13. Garbage Collectors used by JVM