



Advanced Java Programming & Java8

32 Hours

The world of Java is changed. I feel it in my work. I read about it in the blogs. I hear about it the conferences.

Course Overview:

If you want to be a senior java developer, it is not enough to be familiar with java syntax and couple of frameworks. It is not enough to know a few design patterns and coping-pasting them everywhere. You need to know how Java and most popular Java frameworks works under the hood, you should be able to solve different performance issues, to understand in GC and memory concepts, you should be able to write your own frameworks.

You should be familiar with many tools, which are inseparable from Java world today.

Finally, you should be well familiar with functional programming concepts, which are part of Java 8 paradigms.

The purpose of this course is to enrich you Java knowledge and promote you from junior or middle to senior Java developer.

Who Should Attend:

Java developers

Required Skills:

At least 2 years of Java programming language experience

Course Contents:

Design patterns

- Anti-patterns
- SOLID
- Strategy
- Singleton
- Factory
- Builder
- Proxy
- Decorator
- Chain of responsibilities
- Adapter
- Command
- Observer

Multithreading

- Threads
- Runnable / Callable
- How to stop threads correctly
- Synchronization / volatile / locks
- Wait / notify
- Executors



- ThreadLocal
- Immutable objects
- Reactive Java

Reflection, Java internals, 3-d party libraries

- Java optimization
- ClassLoaders
- Classpath
- Class / Method / Constructor / Field
- Annotations
- org.reflections
- How to scan packages
- Dynamic proxy
- Invocation handler
- CGLIB
- Frameworks
- Mockups
- ReflectionUtils
- Generic reflections (Java 8)

GC

- Java memory model
- Heap / stack / perm (till Java 7)
- Young / old generations
- GC & memory parameters
- G1

Profiling Monitoring & Instrumentation

- VisualVM
- MBeans
- VisualGC
- Agents

Lombok

- Annotation Processors
- Pojo annotations
- Functional annotations
- Delombok

New in Java 8 (Optional topic)

- Lambda / functional programming
- Effectively final
- Static and default methods in interfaces
- Stream API
- Stream extensions
- Reflections
- Groovy (this subtopic can be converted to BIG TOPIC 😊)