Kotlin Course (2 days) for Java developers

Day 1 - Morning

The goal of this block is to become immediately productive with Kotlin by focusing on the small elements that make the daily life of the developer simpler. That is to say we concentrate in the low-hanging fruits.

The Basics of Kotlin

- Main
- Declaring a function
- Printing to screen

Kotlin as a better version of Java

- Rich strings
- Data classes
    - lateinit
- Nullability

Setup Development Environment

- Gradle
- IntelliJ IDEA

Kotlin as a Functional Language

- (Im)mutability: val and var
- If as expression
- The when construct
- Methods with the expression body
- Ranges and cycles
- vararg

Extending Existing Classes

- Open/final classes
- Extension methods
- Typealias
- Generics

Day 1 - Afternoon

The goal of this block is to learn how to replace the Java style with the Kotlin one: how to take advantage of Kotlin collections, how to replace *static* elements with *object* ones, etc.

Collections

- Array
- Lambdas
- Collections
- Filter, map
- Find, groupBy
- Fold
- Lazyness (using sequences)
- Flatten

Patterns

- There is no keyword *static* in Kotlin, but there are alternatives:
    - First-level functions (so there is no need to create a class that include them)
    - The keyword *object* is like using the singleton pattern
    - Using *companion object* when it is necessary to access private members of a class

- Default parameters
- Late and lazy initialization
    - Lateinit
    - Lazy() (and delegates)

Metaprogramming

- Reflection
- Annotations

Day 2 - Morning

In this block developers will learn how to design Kotlin software using the most advanced elements.

Advanced Arguments

- Operator overloading
- Lambdas with let, with, apply, run, also
- Destructuring declarations
    - “Returning more values”
- Virtual properties
- Type-safe builders
- Coroutines

Day 2 - Afternoon

The goal of this block is understanding how to integrate or replace Java application with Kotlin ones.

Java Interoperability

- Call Kotlin code from Java
- Using annotations to control Kotlin compilation (JvmName, JvmMultifileClass, JvmField, JvmStatic, JvmOverloads, Throws, JvmSuppressWildcards)
- The type Nothing
- The Java beans convention in Kotlin

Introducing Kotlin in an existing application

- Introducing new tests in Kotlin
- Replacing the Java beans convention with data classes
- Using lambda and interoperability with Java 8 lambdas
- Observation on the JVM code created by the Kotlin compiler

Note: the course program includes exercises and time for discussion between the teacher and the students. These parts are not included in this program.