

Fundamentals of Kotlin

Level: foundation / intermediate

Length: 35 hours / 21 hours (for Java programmers)

Course Objective: solid introduction to Kotlin programming language, mainly its support for procedural, object oriented and functional programming, how it is used in order to solve problems. The shorter version targets Java programmers who want to enter the Kotlin world.

What You Will Learn

- How the code is organized, software development cycle
- Language elements: data structures, control structures, basic types, functions
- Support for object oriented programming (OOP) & functional programming
- Exercise programming with Kotlin
- Exercise the soft skills of communication, team work, presentation of ideas and solutions

Who can participate: there are two main targets:

- anybody who wants to learn Kotlin
- Java programmers

Prerequisites

- Basic notions of programming: familiar with the notion of data type, variable, flow control, ability to describe an algorithm
- Comfortable with using the host operating system
- The knowledge of another (object oriented) programming language, preferable Java

Required infrastructure: workstation with Java installed and an IDE for writing Kotlin code, for example IntelliJ Community Edition

Description

This is a fundamental training for any programmer who wants to use Kotlin regardless the context – desktop apps, mobile apps, web or server side apps. It contains the mandatory language elements and constructs used by any Kotlin program.

The course is designed for programmers who want a solid introduction to Kotlin, the focus is the language, its elements – data structures, control structures,

how the programs are organized, particularities of using it in procedural programming, object oriented programming and functional programming.

Kotlin is exercised by examples and practical assignments, it is shown how the problems are approached and solved, particularities and stereotypes of using this language.

Usually Kotlin is approached by Java programmers because the language history and the intimate link with Java. The reason is that Kotlin was originally an alternative solution to Java, proposed by JetBrains company, in order to improve Java on several directions: simpler and clearer syntax, constructs not supported by Java, asynchronous programming, etc.

Contents

1. Introduction to Kotlin. Its place among other programming languages. Main characteristics, how it relates with Java, code life cycle, distribution.
2. Building and running Kotlin programs. Quick introduction to basic language constructs.
3. Predefined data types for representing common entities like numbers and strings. Arrays. Identifiers and operators, type checks and (safe) casts, exceptions
4. Collections. The three main categories, main operations, how are they used.
5. Control structures and functions. Define functions, safe call operator, Elvis operator, not-null assertion operator. Lambda expressions and higher-order functions. Capture of variables from context, exercise them.
6. Packages – organizing and partitioning the code.
7. Object oriented programming support – classes, interfaces, inheritance, nested classes, enum classes, object declarations and expressions, delegation
8. Generics – variance, covariance, contravariance, type projections.
9. Input / output – console, files, streams, serialization.
10. Mixing Kotlin with Java.