

Introduzione

Kotlin è un linguaggio nato nel 2012 per opera di JetBrains. Permette la creazione di applicazioni di vario tipo, utilizzando principi sia di programmazione a oggetti sia di programmazione funzionale. Esiste da tempo, ma che solo negli ultimi anni è venuto alla ribalta come linguaggio principale supportato da Google per lo sviluppo di applicazioni Android.

In questo libro il lettore avrà la possibilità di imparare Kotlin in tutti i suoi aspetti principali. Il libro si compone di cinque capitoli ciascuno dei quali è dedicato a un argomento specifico.

- Capitolo 1 – *Introduzione a Kotlin*. In questo primo capitolo impareremo diverse modalità con cui scrivere, compilare ed eseguire codice Kotlin. Scriveremo la classica applicazione *Hello World* e impareremo a eseguirla utilizzando la riga di comando e IntelliJ. In questo capitolo vedremo tutti i tipi principali del linguaggio, facendo un parallelo con quelli in Java. Vedremo come gestire il flusso di navigazione attraverso i principali costrutti del linguaggio. In particolare introdurremo il concetto di tipo opzionale. Vedremo poi che cos'è l'approccio multiplatforma di Kotlin, che ne permetterà l'utilizzo anche in piattaforme diverse da quella Java.
- Capitolo 2 – *Le funzioni in Kotlin*. A differenza dei linguaggi basati sulla definizione delle classi, come Java, Kotlin permette un utilizzo diretto di funzioni, le quali si dicono componenti *first citizen*. In questo capitolo vedremo che cos'è una funzione e come si utilizza. Introdurremo il concetto di *extension function* e vedremo come utilizzare le principali funzioni messe a disposizione dal linguaggio.
- Capitolo 3 – *Kotlin orientato agli oggetti*. Kotlin permette di implementare i principali meccanismi alla base della programmazione a oggetti. In questo capitolo vedremo come creare classi e interfacce e come gestire l'ereditarietà. Vedremo alcuni costrutti introdotti da Kotlin, come le *data class* e i tipi *sealed*. Una parte fondamentale del capitolo riguarda la spiegazione delle *generics* e di come esse permettano la creazione di tipi riutilizzabili. Vedremo come funziona il meccanismo di delega e quali sono le utility che il linguaggio ci mette a disposizione per la gestione delle *collection*.
- Capitolo 4 – *Kotlin funzionale*. Come abbiamo detto, Kotlin permette un approccio sia a oggetti sia funzionale. In questo capitolo vedremo quei concetti che rendono Kotlin un ottimo linguaggio per l'utilizzo di un approccio funzionale. Vedremo che cosa siano le *high order function* e come queste possano essere utilizzate nell'elaborazione

delle *collection*. Vedremo in particolare l'utilizzo delle *sequence*, le quali permetteranno di ottenere prestazioni migliori attraverso un approccio *lazy*.

- **Capitolo 5 – Programmazione avanzata in Kotlin.** Abbiamo già accennato a come Kotlin e Java siano legati tra loro. La prima parte di questo capitolo è dedicata alla cooperabilità tra codice scritto in questi due linguaggi. Vedremo non solo come sia possibile utilizzare codice Java da Kotlin e viceversa, ma anche come scrivere codice nuovo in modo da semplificarne la collaborazione. Vedremo due funzionalità molto interessanti come i *contract* e i *type safe builder*, i quali permettono di creare DSL (*Domain Specific Language*) in modo molto semplice. Infine ci occuperemo di coroutine, le quali rappresentano una delle tecnologie emergenti nella realizzazione di codice concorrente.

Kotlin è un linguaggio in continua evoluzione e questa guida vuole essere un modo per introdurre il lettore a questa nuova tecnologia.

Buona lettura!