



Android Entwicklung mit Kotlin und Jetpack

Seminardauer: 5 Tage, 09:00 Uhr bis 17:00 Uhr

Schulungsunterlagen: nach Absprache

Android ist das weltweit führende mobile Betriebssystem, für das lange Zeit v.a. in Java entwickelt wurde. Allerdings konnten meist nicht die hilfreichen Features von Java 8 eingesetzt werden, da dieses nur auf neuen Geräten unterstützt wird. Somit war die Programmierung an vielen Stellen schwerfällig und wirkte insbesondere im Vergleich zu iOS mit der modernen Sprache Swift antiquiert.

Google gab 2017 auf der I/O-Kongferenz Kotlin als weitere offiziell unterstützte Sprache frei, und viele Entwickler sind inzwischen umgestiegen. Auch von Google selbst werden inzwischen die meisten Android-Materialien in Kotlin erstellt.

Im Kurs werden mehrere Apps mit einer aktuellen Release-Version des Android Studio beispielhaft entwickelt, außerdem werden Referenzbeispiele von Google auf Github untersucht. Dabei wird besonderes Gewicht auf die Architekturempfehlungen von Google gelegt, die auf der I/O 2018 unter dem Namen Jetpack veröffentlicht wurden.

Zielgruppe

der kurs richtet sich an Entwickler.

Kurs Voraussetzungen

Kenntnisse in Kotlin,
mindestens Vertrautheit mit Android als Nutzer
hilfreich sind erste Erfahrungen der Androidprogrammierung mit Java

Agenda

Einführung in die Androidentwicklung

- Geschichte von Android
- Überblick über Varianten der Entwicklung (Nativ, plattformübergreifende Ansätze ...)
- Vergleich mit iOS-Entwicklung und Swift
- Android Studio als Entwicklungsumgebung
- Gradle als Buildtool

Programmierung Grundlagen

- Intents und Activities
- Die wichtigsten Komponenten
- Layouts, besonders detailliert: ConstraintLayout
- Master/Detail mit ListView und GridView
- Moderner mit RecyclerView und ViewHolder
- Android KTX: Die Kotlin Extensions zur Erleichterung
- Fragmente für modulare Gestaltung
- Master-Flow in Varianten für Smartphone und Tablet
- Hintergrundaufgaben mit AsyncTask
- Services
- Unit und UI Testing (Espresso)

Einsatz der Android Architecture Components

- Entstehungsgeschichte
- MVVM mit ViewModel, Lifecycle und LiveData
- Das bereitgestellte Navigationskonzept - nur noch eine Activity, mehrere Fragmente
- DataBinding: Viel Programmierarbeit fällt weg
- Room als ORM Framework
- Paging: Sukzessives Laden von Daten

- Workmanager für Backgroundjobs

Weitere Bestandteile von Jetpack

- AppCompatActivity für die vielen verschiedenen Android-Versionen
- Neue Package-Struktur mit androidx.*
- Notifications
- Permissions
- Media und Playback
- Sharing

Guidelines für das UI: Material Design

- Erhebungen, Licht und Schatten als Bausteine
- Layout
- Navigation
 - Der Navigation Drawer
 - AppBars
- Farben
- Schriftarten
- Bildgestaltung
- Spezielle Komponenten u.a.
 - Floating Action Button
 - CardView
 - Sheet
 - Snackbar

Überblick und praktische Demo mit häufig eingesetzten Bibliotheken

- retrofit und Volley für Restful Netzwerkrequests
- Gson und Moshi für Json Handling
- Picasso und Gradle für Laden und Cachen von Bildern
- Dagger 2 für Dependency Injection