

Inhaltsverzeichnis

1	Design & Konzept	1
1.1	Grundlegende Designentscheidungen	1
1.1.1	Android als Plattform	1
1.1.2	Kotlin als Programmiersprache	1

1 Design & Konzept

In diesem Kapitel ...

1.1 Grundlegende Designentscheidungen

1.1.1 Android als Plattform

Das Framework richtet sich ausschließlich an Entwickler die Applikationen für die Plattform Android entwickeln. Es ist damit nicht kompatibel zu iOS, dem Web oder Serverseitigen Anwendungen. Die Spezialisierung lässt es jedoch zu, besser auf mögliche Eigenheiten der Plattform einzugehen. Ein weiterer Grund für diese Entscheidung stellt die Tatsache dar, dass MVI seinen Anfang in der Entwicklung von Webseiten fand und es sich im Fall Android um einen Nachzügler handelt.

1.1.2 Kotlin als Programmiersprache

Die Applikationen in Android und das Android-SDK selbst sind bis vor wenigen Jahren fast ausschließlich in der Sprache Java entwickelt wurden. Seit der Google I/O 2017 gehört jedoch eine weitere Sprache zu den offiziell unterstützen: Kotlin. Sie wird von dem Unternehmen JetBrains entwickelt, die unter anderem die Entwicklungsumgebung IntelliJ für Java produzieren. Dieses bildet auch die Grundlage für Android Studio. Kotlin hat in den letzten Jahren an Bodenhaftung gewonnen und findet auch intern bei Google Verwendung.

Die Sprache wird als statisch typisierte, objektorientierte Programmiersprache bezeichnet und verfügt über eine hohe Interoperabilität zu Java. Dies bedeutet, dass innerhalb eines in Java geschriebenen Programms ohne viel Aufwand Kotlin genutzt werden kann. Dies ist ein wichtiger Faktor für die immer weiter ansteigende Beliebtheit, da es eine einfache Integration und bisherige Projekte gestattet. Kotlin bringt eine verbesserte Syntax mit und macht beispielsweise die Verwendung von null explizit. Zu den Verbesserungen gehören dabei auch:

- Ableitung von Typen
- Alles ist eine Expression
- Funktionen sind First-Class-Funktionen und bilden eine funktionale Grundlage
- Datenklassen machen den Umgang mit unveränderlichen Datenstrukturen einfach
- Erweiterungsfunktionen
- Kovarianz und Kontravarianz werden explizit angewendet
- Standardwerte für Parameter

Abbildungsverzeichnis

Book References

- [2] Martin Fowler. *Patterns of Enterprise Application Architecture*. Addison-Wesley Professional, 13. Jan. 2013, S. 19–22.
- [3] Donald Wolfe. *3-Tier Architecture in ASP.NET with C sharp tutorial*. SitePros2000.com, 13. Jan. 2013.
- [4] Adam Boduch. *Flux Architecture*. Packt Publishing, 30. Jan. 2017, S. 27, 198, 312.
- [5] Ilya Gelman und Boris Dinkevich. *The Complete Redux Book*. Leanpub, 30. Jan. 2017, S. 6–7.
- [6] Adam Boduch. *Flux Architecture*. Packt Publishing, 30. Jan. 2017.
- [8] Robin Wieruch. *Taming the State in React: Your journey to master Redux and MobX*. CreateSpace Independent Publishing Platform, 5. Juni 2018, S. 3.
- [9] Ajdin Imsirovic. *Elm Web Development: An introductory guide to building functional web apps using Elm*. Packt, März 2018.
- [10] Ajdin Imsirovic. *Elm Web Development: An introductory guide to building functional web apps using Elm*. Packt, März 2018, S. 50–65.
- [14] Sergi Mansilla. *Reactive Programming with RxJS: Untangle Your Asynchronous JavaScript Code*. PRAGMATIC BOOKSHELF, 12. Jan. 2016, S. 103–110.

Artikel Referenzen

- [1] Thomas A Wadlow. „The Xerox Alto Computer“. In: (Sep. 1982). URL: <https://tech-insider.org/personal-computers/research/acrobat/8109-e.pdf>.
- [11] Jagatheesan Kunasaikaran und Azlan Iqbal. „A Brief Overview of Functional Programming Languages“. In: *electronic Journal of Computer Science and Information Technology (eJCSIT)* Vol. 6, No 1 (2016). URL: ejcsit.uniten.edu.my/index.php/ejcsit/article/view/97/39.

Online References

- [7] Facebook Developers. *Hacker Way: Rethinking Web App Development at Facebook*. 4. Mai 2014. URL: <https://www.youtube.com/watch?v=nYkdrAPrdcw> (besucht am 08.07.2019).
- [12] Andre Staltz. *What if the user was a function? by Andre Staltz at JSConf Budapest 2015*. Youtube. 4. Juni 2015. URL: <https://www.youtube.com/watch?v=1zj7M1LnJV4> (besucht am 14.07.2019).

- [13] Hannes Dorfmann. *Model-View-Intent on Android*. 4. März 2016. URL: <http://hannedorfmann.com/android/model-view-intent> (besucht am 23.05.2019).