

Kotlin

Seminar Programmiersprachen

Daniel Karl

5. Februar 2024

Geschichte

- Kotlin wurde von JetBrains entwickelt
- **Intention:** eine moderne Alternative zu Java
- 2010:
 - Projekte wurden zuvor mit Java entwickelt
 - wachsende Codebasis führte zu erschwerter Wartbarkeit
 - Fokus auf Prägnanz und Interoperabilität mit Java
- 2016:
 - Veröffentlichung der ersten Version von Kotlin
- 2017:
 - Google berücksichtigt Kotlins Features bei der Weiterentwicklung der Android-Plattform
- Heute:
 - wird aktiv weiterentwickelt
 - aktuelle Version: 1.9.22

Typsystem

- statisches Typsystem
- Typinferenz macht explizite Typangabe optional
- nullbare und nicht-nullbare Typen
- implizite Typkonversion nur auf Supertypen
- alle Typen in Kotlin basieren auf einer Klassendefinition

Interoperabilität mit Java

- Quellcode wird zu Java Bytecode übersetzt, wenn die JVM als Zielplattform gewählt wurde
- Java Klassen können direkt in Kotlin Projekte eingebunden werden
- nullable Java Objekte werden zu Plattformtypen konvertiert
 - besonderer Typ der nullable und nicht-nullable sein kann
 - erleichtert Einbindung von Java Code
- Unterstützung von sämtliche Java Bibliotheken

Prägnanz

- *val* und *var* zum Deklarieren von Variablen
- getter und setter werden automatisch generiert
- **Datenklassen** bieten eine vereinfachte Schreibweise für Klassen die nur aus Properties bestehen
- **Higher-Order Funktionen**, nehmen Funktion als Parameter und/oder geben Funktion zurück. Erleichtern bspw. das Arbeiten mit Collections.
- **Extensions**, erweitern Klassen ohne Vererbung

Coroutines

- Coroutines erlauben es effizient laufende asynchrone Programme zu schreiben
- werden innerhalb eines Threads ausgeführt und besitzen ihren eigenen Kontext
- lassen sich mit suspendierenden Funktionen anhalten
- in der Zwischenzeit können andere Coroutines im Thread ausgeführt werden
- sind unabhängig von einem Thread

Infrastruktur

- Buildsysteme: Gradle, Maven
- Dokumentation: KDoc (analog zu JavaDoc)
- Entwicklungsumgebung: IntelliJ, Android Studio, Eclipse
- Java to Kotlin converter (J2K), konvertiert automatisch Java Klassen zu Kotlin Dateien
- IntelliJ Profiler, als Werkzeug zur Optimierung von Speichernutzung und Laufzeit eines Programms

Anwendungsbereiche

- Android App Entwicklung: von Google empfohlene Programmiersprache
- Desktop- und serverbasierte Anwendungen
- Multiplatform (Android und iOS)
- Data Science und Machine Learning

Bekannte Projekte

- Google Home App, Duolingo Android App
- Umstieg von Java zu Kotlin
- verbesserte Wartbarkeit und deutliche Reduzierung der Codezeilen

Implementierung

■ rekursive Dateisuche:

- existierende Java Bibliotheken enthalten hilfreiche Kotlin-Extension Funktionen (isRegularFile(), isDirectory())

■ Erkennen von Binärdateien:

- Binärdateien haben typischerweise vermehrt Steuerzeichen und ASCII Zeichen aus dem Erweiterten ASCII Zeichen Satz
- Algorithmus nimmt 1024 Bytes einer Datei und zählt die typischen und untypischen ASCII Zeichen
- falls über 50% typische ASCII Zeichen, wird die Datei als Binärdatei betrachtet

Implementierung

■ Suchen von regulären Ausdrücken:

- Einbindung des java.io Pakets
- vereinfachte Iteration durch die Textzeilen einer Datei mithilfe von Higher-Order Extension Funktionen
- Suche von regulären Ausdrücke mit dem java.util.regex Paket

■ Command Line Interface:

- mit der externen clikt Bibliothek umgesetzt

Vielen Dank für Ihre Aufmerksamkeit!