



# KOTLIN VS. JAVA

What, Why, Differences, Similarities, Comparison

# Wer bin ich?

- Milan Stuhlsatz
- Studium: Softwaretechnik in Lübeck
- Seit 02/2023 bei Obi (OST)
- Junior Software Engineer
- Team Groot (ehm. Decide)



# Bevor wir anfangen...

- Perspektive
  - *Java Entwickler*
  - *Erster Kontakt mit Kotlin*
- Ziel des Vortrags
  - *Grundlagen in Kotlin*
  - *Vor- und Nachteilen gegenüber Java*
- Aufbau
  - *Gegenüberstellung Java und Kotlin*
  - *Jeweilige Vor- und Nachteile*
  - *Demonstration*
  - *Fazit*
  - *Weiterführende Hinweise*

# Java ist...

- Erscheinungsjahr: 1995
- Open Source
- JDK + JVM
- Plattformunabhängig
- Bytecode
- Statically typed und objektorientiert
- Garbage Collection

# Kotlin ist...

- Entwickelt von JetBrains
- Vorgestellt: Juni 2011
- OpenSource (Apache 2 Licence)
- Verwendung:
  - *Standardsprache für Android*
  - *Corda (Banksoftware)*
  - *Buildscripts für Gradle*
  - *Spring*
  - *Android Apps like Trello, Evernote, Atlassian Apps, Facebook*

# Kotlin ist...

- Statically typed, objektorientiert und funktional
- Kompiliert zu:
  - *JVM-Bytecode*
  - *Javascript*
  - *Machinecode(LLVM)*
- “Java, if it was designed today”

# Warum Java?

- Einfach zu erlernen
- Plattformunabhängig dank Bytecode
- Community und Ressourcen
- APIs und Schnittstellen
- Sicherheit
- Checked Exceptions

# Warum Kotlin?

- Conciseness
- Interoperabilit
- Safety
- NULL-Safety
- Data Classes
- Extension Functions
- Smart cast
- Type Inference
- Coroutines



# Kotlin Basics

- Funktionen: fun
- Variable: var
- Read-Only Variable: val
- „;“ am Zeilende nur nach bei >1 Anweisung, sonst optional
- Infered types -> typisierung optional (auch für Funktionen)
  - `var number : Int = 7`
- „?“ markiert Datentypen als Nullable
  - `var number : Int? = Null`

# Hello World

Java

```
public static void main(String[] args) {  
    System.out.println("Hello world!");  
}
```

Kotlin

```
■ fun main() {  
    println("Hello World!")  
}
```



# Fazit

- Kotlin ist moderner und konsequenter im Design
- Kotlin löst alte Strukturen schneller ab, nimmt aber „breaking changes“ hin.
- Kotlin ist effizienter
- Java ist präziser
- Java hat technische Vorteile (z.B. schnellere Builds)
- Java Projekte können mit Kotlin ergänzt werden
- Java kennt „checked exceptions“

# Zur Vertiefung:

- Offizieller Guide: <https://kotlinlang.org/docs/getting-started.html>
- Youtube Reihe (deutsch):  
<https://www.youtube.com/playlist?list=PLNmsVeXQZj7rylgyThgUldHG8KE6Nbc10>
- Baeldung: <https://www.baeldung.com/kotlin/>

