Freie Universität Berlin

Überbrückung der Sprachgrenze zwischen Java und JavaScript in der GUI des Saros-Plug-ins

Gliederung

1. Einführung
2. Motivation
3. Evaluation
4. Implementierung
5. Zusammenfassung
6. Ausblick

Einführung

Saros:

- Plug-in für verteilte Paarprogrammierung
- Für Eclipse und IntelliJ IDEA
- Evaluation einer HTML-GUI

MOTIVATION

Motivation:

- Mehrfache Implementierung verhindern
- Statische Codeprüfung erleichtern
- Einarbeitung in Quellcode vereinfachen

Evaluation der Technologien

Anforderungen:

- JVM-Bytecode oder kompatibel mit Java
- Code in JavaScript kompilierbar
- Einarbeitung erleichtern
- Ant und Gradle kompatibel

Kandidaten:

- Fantom
- Scala
- Kotlin

Ergebnisse der Evaluation

	Fantom	Scala	Kotlin
Java kompatibel	++	++	++
Ausgabe JS-Code	+	+/-	++
Integration Build-Prozess	-	+	++
Einarbeitung	+/-	+	++

Implementierung

Herausforderungen:

- Debugging
- PicoContainer-Framework
- SWT-Browser
- Defekte im Plug-in für Kotlin

Zusammenfassung

Ergebnisse

- Kotlin als „beste“ Lösung
- UI-Modul vollständig implementiert
- Übernahme der Unit-Tests mit wenig Aufwand
- Lediglich vier Model-Klassen

Fazit

- Kotlin erfüllte alle Anforderungen
- Weitere Vorteile von Kotlin
  - Veringert Quellcode
  - Koroutinen für Nebenläufigkeit
- Viel Aufwand für wenig Modellklassen

Ausblick

Ausblick

- UI.Frontend-Modul in Kotlin implementieren
- Umstieg von Java zu Kotlin für das Saros-Projekt

Vielen Dank

Modellklassen:

- ProjectTree + ProjectNode
- State
- Contact
- Account