JOHANNES KEPLER
UNIVERSITÄT LINZ

Master's Thesis

Domain-Specific Languages in Kotlin and Scala - A comparison

Student: Clemens Bartl
SKZ/Matr.Nr.: 01255728
Email: clemens.bartl@gmail.com

Advisor: a.Univ.-Prof. Dr. Herbert Prähofer
Start date: October 2020

a.Univ.Prof. Dr. Herbert Prähofer
Institute for System Software

T +43 732 2468 4352
F +43 732 2468 4345
herbert.praehofer@jku.at

A declared goal for the programming languages Scala and Kotlin is to support the development of domain-specific languages (DSLs), i.e., languages that target specific application domains (cf. [1] and [2]). However, both languages provide rather different concepts for building DSLs. For example, Scala has by-name parameters and implicit conversions. Kotlin, on the other side, support extension functions and lambdas with receivers. Both approaches, however, result in similar capabilities.

Therefore, this master thesis should do a detailed comparison of the two approaches of Scala and Kotlin for building DSLs. This includes:

- ☐ a detailed description of the language concepts in the two languages
- ☐ the study of existing DSLs in both languages, e.g.:
  - Spec2 – a language for behavioral-driven design
  - Scala Combinator Parser – a functional library for combinator parsers
  - Scala Spark – A DSL for the Apache Spark framework for cluster computing
  - Kotlin's Exposed – A DSL for data base queries
- ☐ Implementation of sample DSLs in both languages for comparing their capabilities and expressive power

Thesis outline:

1. Introduction and motivation
2. Approaches for building domain-specific languages
3. Language features for building DSLs in Kotlin and Scala
4. Existing DSL examples in Kotlin
5. Existing DSL examples in Scala
6. Case study DSLs in Kotlin and Scala
7. Comparison and conclusion