Eine Java-Erweiterung mit Typinferenz für polymorphe Typen

Andreas Stadelmeier, M.Sc. | Wissenschaftlicher Mitarbeiter
07451 521-421 | a.stadelmeier@hb.dhbw-stuttgart.de

Betreuer

DHBW Stuttgart Campus Horb: Prof. Dr. Martin Plümicke

Albert-Ludwigs- Universität Freiburg, Institut für Informatik: Prof. Dr. Peter Thiemann

Bearbeitung

seit 3/2020

Java ist eine statisch typisierte objektorientierte Programmiersprache. Sie bietet den Vorteil, komplexe Programme in kleinere Einheiten zu untergliedern, die sogenannten Objekte. Zusätzlich hilft das statische Typsystem, grobe Fehler im Programm frühzeitig zu erkennen.

Java's Typsystem hat allerdings den Nachteil, dass der Programmierer selbst die korrekten Typen für sein Programm vergeben muss. Zudem geht ein gewisser Teil der Typinformationen beim Kompiliervorgang verloren. „Type erasure“ nennt sich dieser Effekt, welcher der Programmiersprache zusätzliche Einschränkungen auferlegt. Bestimmte Programme, die eigentlich Fehlerfrei und korrekt typisiert sind, lassen sich aufgrund von Type erasure nicht kompilieren und ausführen.

Abb.: Java und Java-TX im direkten Vergleich

Das DHBW-Forschungsprojekt Java-TX (Java Type Extend) versucht eine Lösung für diese beiden Probleme zu finden. Der eigens entwickelte Java-TX Compiler kann Typen für Java-Programme inferieren und diese anschließend kompilieren. Typinferenz bedeutet, dass der Programmierer Typangaben im Quellcode auslassen kann. Diese werden anschließend automatisch berechnet und eingesetzt.

Die Promotion soll eine Lösung für das Problem „Type erasure“ erarbeiten und dabei den Java-TX Compiler erweitern. Ohne „Type erasure“ könnte der Java-TX Typinferenzalgorithmus zusätzliche Programme kompilieren, für welche es momentan keine korrekte Typisierung gibt.