

IDE-Plugin Schnittstelle für Java-TX mittels Magpie-Bridge

Beschreibung: Das Java-TX Projekt entwickelt einen Typinferenzalgorithmus für Java. Der Java-TX Compiler kann den optimalen Typ für ein Java Programm berechnen. Ein IDE-Plugin soll diese Funktionalität nun benutzbar machen. Mithilfe von Anmerkungen im Code könnten dem Programmierer mögliche Typeinsetzungen direkt in der Entwicklungsumgebung (IDE) vorgeschlagen werden.

Das MagpieBridge-Projekt ist ein Framework für das Erstellen von IDE-Plugins. Es ermöglicht die Erstellung eines `LanguageServers`, welcher dann mithilfe des Language Server Protocols (LSP) mit allen gängigen IDEs kommuniziert.

Mithilfe von Magpie-Bridge soll in dieser Studienarbeit ein Plugin für die Java-TX (Java Type eXtended) Sprache entwickelt werden. Dabei kann der Java-TX Compiler benutzt werden um fehlende Typangaben zu berechnen und diese anschließend mittels Magpie-Bridge dem Programmierer in der IDE als Code-Completion anzubieten.

Konkrete Aufgabe: Implementierung einer Schnittstelle zwischen einer Java-IDE (beispielsweise IntelliJ) und dem Java-TX Compiler:

- Der Java-TX Compiler fungiert als Language-Server
- Magpie-Bridge kann als Framework benutzt werden
- Mithilfe des LSP-Protokolls soll das Tool mit den gängigsten IDE's funktionieren

erforderliche Kenntnisse: Generelles Interesse an der Programmierung mit Java sind von Vorteil. Es sind KEINE vertieften Compilerbaukenntnisse nötig!

Betreuer:

- Andreas Stadelmeier, a.stadelmeier@hb.dhbw-stuttgart.de
- Martin Plümicke, pl@dhbw.de

Diese Studienarbeit legt ihren Schwerpunkt auf den Teil der praktischen Umsetzung. Konkret bedeutet das, dass der größte Zeitaufwand die Einarbeitung in das MagpieBridge-Projekt und die eigentliche Java-Programmierung einnehmen wird.

Herausforderungen dieser Studienarbeit

- Erweiterung des Java-TX Compiler um Typeinsetzungen in vorhandenen Source-Code
- Einarbeiten in das MagpieBridge-Projekt
- Ausarbeiten des User-Workflows für das Arbeiten mit Java-TX als IDE-Plugin

Bei Interesse können sie sich auch bereits vorab bei Daniel Holle melden. Er gibt gerne eine Einführung ins Projekt und steht für Fragen zur Verfügung.

Antlr mit Records als Ausgabe

Beschreibung: Der ANTLR Parsergenerator generiert Parser, welche eine Eingabe zu einem sogenannten ParseTree verarbeiten. Die Knoten dieses Baums sind als Java-Klassen realisiert.

Die mit Java eingeführten Record Datenstrukturen würden sich hier allerdings besser eignen, denn Records erleichtern es funktionale Programmierpatterns auf die Baumstruktur anzuwenden. In dieser Studienarbeit soll eine Erweiterung für ANTLR erarbeitet werden.

Konkrete Aufgabe: Der von ANTLR generierte ParseTree soll mithilfe von Records aufgebaut werden

erforderliche Kenntnisse: Generelles Interesse an der Programmierung mit Java sind von Vorteil

Betreuer:

- Andreas Stadelmeier, a.stadelmeier@hb.dhbw-stuttgart.de
- Martin Plümicke, pl@dhbw.de

Diese Studienarbeit legt ihren Schwerpunkt auf den Teil der praktischen Umsetzung. Konkret bedeutet das, dass der größte Zeitaufwand die eigentliche Java-Programmierung einnehmen wird.

Diese Studienarbeit bietet die Möglichkeit

- Einblick in ein großes Softwareprojekt zu erhalten (25 000 Zeilen Code)
- Eine Erweiterung für ein in der Praxis wertgeschätztes Parsergenerator-Tool zu erarbeiten
- eng mit einem erfahrenen Team zusammenzuarbeiten
- seine Kenntnisse im Bereich von Java, funktionalen Programmier-Patterns und Parsergeneratoren zu vertiefen.

Bei Interesse können Sie sich auch bereits vorab melden. Er gibt gerne eine Einführung ins Projekt und steht für Fragen zur Verfügung.