

Aufgabe 1: Einrichten des Projekts

1. Ein Parser zum Einlesen von Decaf-Syntax ist bereits zum Teil vorhanden und als ZIP-Archiv unter <http://www2.ba-horb.de/~stan/Decaf.zip> verfügbar. Importieren sie das Projekt in ihre IntelliJ Entwicklungsumgebung. Es handelt sich um ein Maven-Projekt, dieses lässt sich in IntelliJ importieren mittels: *File* → *Open* → *pom.xml auswählen* → *Open as Project*.
2. Generieren des Parsers: (2 Möglichkeiten)
 - (a) Auf der Kommandozeile mittels: `mvn antlr4:antlr4`. Das ANTLR-Maven-Plugin generiert den Parser in das Verzeichnis `target`. IntelliJ sollte den Parser automatisch erkennen und das Projekt ausführbar sein.
 - (b) Ansonsten kann der Parser auch manuell erstellt werden. Zuerst konfigurieren mittels *Configure ANTLR*. Hier nicht vergessen den Haken bei *generate parse tree visitor* zu setzen. Anschließend wie gehabt den Parser ins `src` Verzeichnis generieren lassen.

Aufgabe 3: Erstellen eines abstrakten Syntaxbaums

a):

Vervollständigen sie die Klasse `ExpressionGenerator`. Die Methodenrumpfe sind bereits vorgegeben. Sie können ihre Implementierung anhand des `ExpressionGeneratorTest` verifizieren.

b)

Implementieren sie eine Klasse `StatementGenerator`. Diese soll den `Statement`-Teil des Abstrakten Syntaxbaums generieren. Hier kann auch wieder der `DecafBaseVisitor` verwendet werden. Schreiben sie zusätzlich im `StatementGeneratorTest` die Testfälle zu `If`, `IfElse` und `FunctionCall` statements.

c)

Vervollständigen sie die Klasse `ASTGenerator`. Der `CompilerTest` sollte anschließend erfolgreich durchlaufen.