

Was ist ein Compiler?

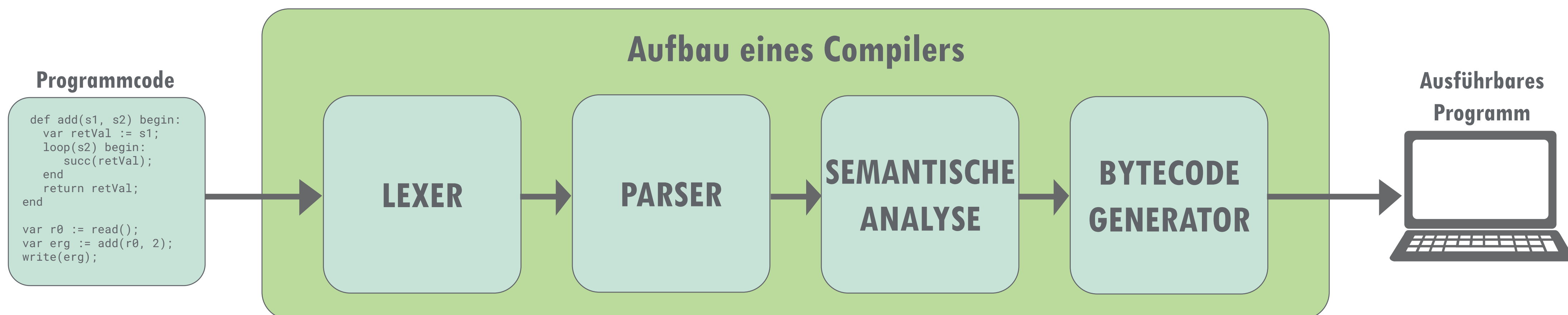
Bei einem Compiler handelt es sich um ein Computerprogramm, welches als Eingabe ein Programmcode erhält und als Ausgabe eine Datei erzeugt, welche von einem Computer ausgeführt werden kann. Das Ziel dieser Arbeit war es, eine eigene Sprache zu spezifizieren und für diese Sprache einen Compiler zu entwickeln, welcher Java Bytecode erzeugt. Dieser kann anschließend mithilfe der Java virtual machine ausgeführt werden.

Was ist die Programmiersprache WHILE?

Die Programmiersprache WHILE besitzt einen Datentyp: positive ganze Zahlen. Es ist möglich, Variablen einen Wert zuzuweisen und sie um einen Wert zu erhöhen oder zu reduzieren. Werte für Variablen können über die Konsole eingelesen und ausgegeben werden. Es existieren zwei Schleifentypen: while und loop. Der Nutzer darf eigene Funktionen definieren. Funktionen wie die Addition müssen selbst implementiert werden.

Was ist eine Grammatik?

Eine Grammatik besteht aus einer Menge von Grammatikregeln, welche spezifizieren wie der Programmcode einer Programmiersprache aufgebaut sein muss und welche Konstruktionen erlaubt sind. Durch die Grammatikregeln lässt sich beispielsweise festlegen, mit welchem Schlüsselwort eine Variablendeklaration beginnt und ob Schleifen innerhalb von Schleifen in der Programmiersprache zulässig sind oder nicht.



Was macht der Lexer?

Der Lexer, oder auch Scanner genannt, ist die erste Komponente in einem Compiler. Die Aufgabe des Lexers ist es, die einzelnen Zeichen in einem Programmcode zu einer Reihe von Token entsprechend einer Grammatik zusammen zu fügen. Ein Token ist eine zusammengehörige Menge von Zeichen in einer Programmiersprache. Beispielsweise ist das Schlüsselwort „var“ ein Token, genau so wie der Name einer Variable oder der Zuweisungsoperator „:=“.

Was ist ein Parser?

Ein Parser ermittelt ob ein Programmcode der Grammatik entspricht. Der Parser erhält vom Lexer alle Token und prüft, ob die Token in der spezifizierten Reihenfolge auftauchen. Der Parser erkennt dabei Fehler wie beispielsweise einen fehlenden Variablennamen in in einem Schleifenkopf. Der Parser erzeugt einen Abstract-Syntax-Tree, welcher das Programm abbildet. Lexer und Parser können mithilfe eines Parsergenerators und einer Grammatik automatisch generiert werden.

Wozu die semantische Analyse?

Die semantische Analyse ist notwendig, da es schwer ist, logische Fehler durch Grammatikregeln zu verhindern. Es ist syntaktisch korrekt, eine Variable auf die Konsole auszugeben, welche noch nicht definiert wurde aber semantisch falsch. Um solche Fehler zu erkennen, wird durch den erzeugten Abstract-Syntax-Tree gelaufen und jede definierte Variable oder Funktion in einer Datenstruktur notiert. Anschließend kann bei der Nutzung geprüft werden, ob diese bereits existieren.

Wie funktioniert die Codegenerierung?

Zuletzt wird ein ausführbares Programm erzeugt. Für die Codegenerierung wird die freie Bibliothek „Java ASM“ genutzt. Damit ist es möglich, durch das Aufrufen von Java-Methoden ausführbaren Bytecode zu erzeugen. Dazu wird die Datenstruktur genutzt, welche bei der semantischen Analyse erzeugt wurde. Jedes Konzept der Programmiersprache WHILE wird auf eine Menge von Anweisungen an die Java virtual machine abgebildet.