# GraalJS

GraalJS ist ein Interpreter und Compiler für eine kleine Teilmenge von JavaScript, der Graal IR erzeugt.

Die Teilmenge unterstützt dabei:

* Variable Deklarationen
* If- und While-Statements
* Zuweisungen
* binary comparison Operatoren

## Phasen

Die Ausführungen von Scripten wird in vier Phasen durchgeführt:

GraphBuilder

Interpreter

PrePass

Parser

* Parser

Mit der Hilfe des Parser von Rhino wird der Source Code geparst und ein AST erzeugt, auf dem alle restlichen Phasen operieren

* PrePass

In dieser Phase wird

* + der AST linearisiert
  + die maximale Operanden Stack Größe berechnet
  + die Anzahl der Variablen berechnet
  + If- und While Statements werden in Sprünge umgewandelt
  + bei Expressions wird der Index im linearen Buffer gespeichert
  + und bei allen Namen von Variablen wird der Index der Variable gespeichert
* Interpreter

Der Interpreter interpretiert das gesamte Script einmal mit Hilfe von zwei Arrays, die die lokalen Variablen und den Operanden Stack enthalten. Dabei wird bei allen Expressions der Type des Ergebnisses nach der Interpretation gespeichert.

* GraphBuilder

In dieser Phase wird der AST traversiert und mit dem im Interpreter gespeicherten Type-Feedbacks Graal Nodes erzeugt, die visualisiert und ausgeführt werden können.

## Klassenübersicht:

* PrePass, Interpreter und GraphBuilder

Implementiert die entsprechenden Phasen

* SafeAddNode

Graal Node, der eine Deoptimize-Operation erzeugt, und damit einen Sprung in den Interpreter durchführt

* DeoptHandler

Wird von Graal bei einer Deoptimize-Operation aufgerufen und ruft daher dementsprechend den Interpreter auf

* FrameModifierImpl

Modifiziert den Frame der Mock-Methode so, dass sie später vom Interpreter auch verwendet werden kann

* Main

Parst den Source Code und ruft die verschiedenen Phasen auf