

Master's Thesis

Realization of a JavaWiz backend system running on a JavaScript runtime

Dr. Herbert Prähofer
Institute for System Software

T +43 732 2468 4352
F +43 732 2468 4345
herbert.praehofer@jku.at

Student: Felix Schenk
SKZ / Matr.Nr.: 921 / K11904005
Email: felix.schenk@jku.at
Advisor: Dr. Herbert Prähofer
Start date: Oktober 2023

JavaWiz is a tool currently being developed at the Institute for System Software at Johannes Kepler University Linz to support software development education in the first semesters. It is best described as a visual debugger, as it offers different views tailored to different problems and tasks with the main goal of being helpful and easy to understand for novice programmers.

For example, a view of executed statements allows students to see the history of statements executed and what changes were made to which program variables. Another view allows students to observe the current state of the call stack as well as the heap, with a focus on depicting how they change during program execution.

The frontend of JavaWiz is implemented in TypeScript, it is based on the D3.js visualization library, and is currently offered as a plugin for Visual Studio Code. To debug the user-provided source codes, JavaWiz uses its own trace-generating debugger written in Java/Kotlin using JDI (Java Debug Interface).

Instead of being available as a plugin for Visual Studio code (that starts a local debugger process behind the scenes), JavaWiz should be usable as a pure online tool that can be easily embedded into web pages. This requires the execution and debugging of Java programs directly in the web page instead of a separate Java process.

In order to achieve this, a method must be implemented that allows Java programs to be executed in the browser through JavaScript runtime environments. Thus, the following approach should be followed in this thesis:

- In a first step, the JavaWiz backend system should be implemented based on Espresso, Oracle's Java interpreter implementation based on the Truffle language implementation framework (<https://www.graalvm.org/latest/graalvm-as-a-platform/language-implementation-framework>).
- The implementation of the JavaWiz backend system in Espresso then must be translated into a JavaScript application. This can be accomplished by the WebImage tool which is capable of transforming Java into JavaScript.