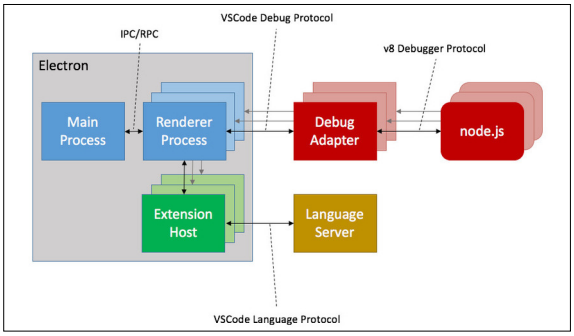

Arooran Thanabalasingam


Giovanni Heilmann

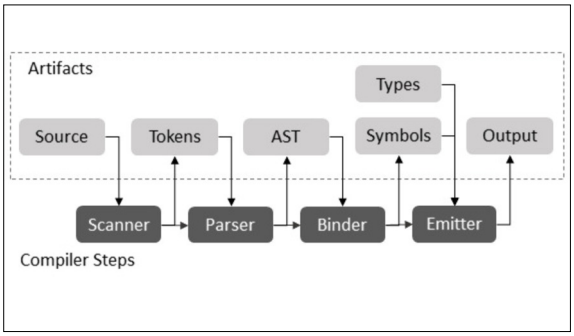
Graduate Candidates	Arooran Thanabalasingam, Giovanni Heilmann
Examiner	Thomas Corbat
Co-Examiner	Lukas Felber, Quatico Solutions AG, Zürich, ZH
Subject Area	Software Engineering - Core Systems

TypeScript Refactorings

for Visual Studio Code



A language server, which provides language analysis, can be developed once and reused for different code editors.



The compiler goes through a number of steps to analyse TypeScript code and transforms it to executable JavaScript code.

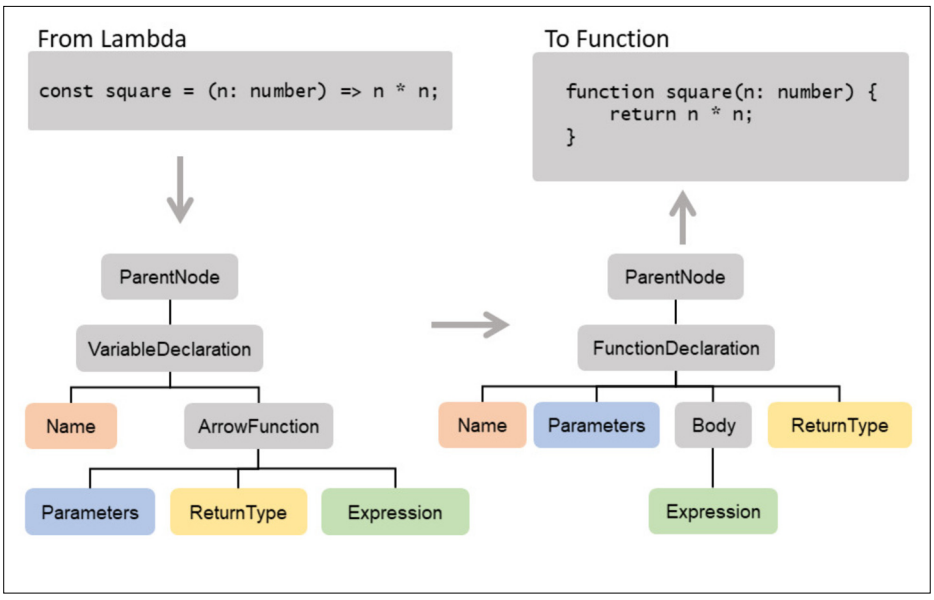
Introduction: Visual Studio Code is an increasingly popular text editor written in TypeScript. Language support and other features can be added via extensions for any programming language. Such language extensions access a language server, which provides static analysis features for the target language. Both the TypeScript compiler and VS Code are open-source projects maintained by Microsoft. The TypeScript compiler has a built-in language server for features like auto-complete, automated refactorings and quick-fixes. A quick fix is an immediate solution for a transient error or deficiency in the code while programming. A refactoring is a behavior-preserving code change. It is used during the process of refactoring whose goal is to make code more maintainable. Proper tooling support for both features automates and accelerates repetitive coding tasks. Besides increased productivity, such automated code changes are less error prone than the same modifications applied manually.

Objective: Visual Studio Code already offered some refactorings for TypeScript. The goal of this project has been to contribute additional refactorings and quick-fixes, chosen by the students as a preliminary task. All new features should be pushed to the official plug-in repository. In this way this project will improve the quality of life and productivity of many TypeScript developers.

Result: In total, four refactorings have been implemented:

- Inline Local
- Inline Function
- Convert Lambda to Function
- Convert String Concatenation to Template Literal

Furthermore, one quick-fix named Interface Stubbing has also been implemented. Some of these features are among the most commonly used among developers, especially Inline local and Inline function, which complement the already existing refactoring: Extract symbol. At the time of writing this document, the pull-request for Convert lambda to function has been reviewed and approved by Microsoft. The other features are waiting for reviews.



Refactorings change the TypeScript code by transforming the Abstract Syntax Tree and rendering the modifications as source code changes.