|  |  |
| --- | --- |
|  | Fachhochschul-Bachelorstudiengang  **SOFTWARE ENGINEERING**  A-4232 Hagenberg, Austria |

**Vergleich der Extension-APIs  
in VS Code und IntelliJ**

Bachelorarbeit

zur Erlangung des akademischen Grades

Bachelor of Science in Engineering

Eingereicht von

**Philipp Seiringer**

Begutachtet von Dr. Josef Pichler

Hagenberg, Februar 2024

**Erklärung**

Ich erkläre eidesstattlich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benutzt und die den benutzten Quellen entnommenen Stellen als solche gekennzeichnet habe. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt.

Die vorliegende, gedruckte Bachelorarbeit ist identisch zu dem elektronisch übermittelten Textdokument.

Datum Unterschrift

Kurzfassung

Abstract

Inhaltsverzeichnis

[Kurzfassung 3](#_Toc145095000)

[Abstract 4](#_Toc145095001)

[Inhaltsverzeichnis 5](#_Toc145095002)

[1 Einleitung 6](#_Toc145095003)

[1.1 Motivation 6](#_Toc145095004)

[1.2 Ziel 6](#_Toc145095005)

[1.3 Aufbau der Arbeit 6](#_Toc145095006)

[2 Vergleich der Plattformen 7](#_Toc145095007)

[2.1 VS Code 7](#_Toc145095008)

[2.2 IntelliJ 7](#_Toc145095009)

[3 Vergleich der angebotenen Funktionalität 8](#_Toc145095010)

[4 Codebeispiel: Recent Changes Plugin 9](#_Toc145095011)

[5 Vergleich der Code-Metriken 10](#_Toc145095012)

[6 Conclusion 11](#_Toc145095013)

[Referenzen 12](#_Toc145095014)

1. Einleitung
   1. Motivation

Softwareentwickler arbeiten täglich mit verschiedensten Werkzeugen und Entwicklungsumgebungen, sogenannten IDEs (=Integrated Development Environment). Diese Plattformen bieten teils sehr unterschiedliche Funktionalitäten, die die Softwareentwicklung erleichtern sollen. Dabei bieten sie Unterstützung für verschiedenste Programmiersprachen und Technologien und binden zahlreiche Werkzeuge für spezifische Anwendungsfälle ein. Aufgrund des immer rascher werdenden Entstehens von neuen Technologien bieten mehr und mehr IDEs Möglichkeiten zur Entwicklung von eigenen Plugins, welche dann auch an andere Entwickler bereitgestellt werden können. So können in kürzester Zeit neue Technologien unterstützt werden und Entwickler haben selbst die Macht darüber zu entscheiden welche Plugins sie nutzen möchten und welche nicht.

Vor der Entwicklung solcher Plugins ist es wichtig zu entscheiden für welche IDE das Plugin erstellt werden soll. Dabei spielen Aspekte wie zum Beispiel die Einfachheit und Flexibilität in der Entwicklung, der Umfang an angebotener Funktionalität, die Möglichkeit die Nutzerinteraktion und somit die User Experience zu steuern und viele weitere eine Rolle. Diese Bachelorarbeit versucht in diesen Bereichen einen Überblick zu schaffen und vergleicht hierfür die Plugin Entwicklung in zwei der momentan beliebtesten IDEs, Visual Studio Code und IntelliJ IDEA. Durch den Vergleich der beiden Produkte und dem Herausarbeiten und Aufbereiten der Unterschiede wird es anderen Entwicklern erleichtert diese Entscheidung zu treffen.

* 1. Ziel
  2. Aufbau der Arbeit

1. Vergleich der Entwicklungsplattformen
   1. Visual Studio Code
   2. IntelliJ IDEA

IntelliJ IDEA wurde erstmals im Januar 2001 von dem Unternehmen JetBrains veröffentlicht. Im Gegensatz zu Visual Studio Code handelt es sich bei IntelliJ um ein IDE welches speziell auf die Entwicklung von Programmen in den Programmiersprachen Java und Kotlin ausgelegt ist. IntelliJ IDEA wird in einer frei zu verwendenden, open source „Community Edition“, sowie in einer kommerziellen Form als „IntelliJ IDEA Ultimate“ angeboten.

Aufgrund der Spezialisierung auf Java und Kotlin Projekte unterstützt die IntelliJ Community Edition nur eine relativ kleine Auswahl an Sprachen, Frameworks und Build Tools. Während IntelliJ IDEA Ultimate den Umfang an Features schon deutlich erweitert, bietet JetBrains auch noch weitere (kommerzielle) IDEs an. Diese sind alle für unterschiedliche Programmiersprachen oder Sprachfamilien ausgelegt. Einige der bekanntesten sind dabei CLion für die Sprachen C und C++, Rider für die .NET Sprachen, PhpStorm für PHP, WebStorm für JavaScript und viele weitere. Zum aktuellen Zeitpunkt sind es insgesamt elf verschiedene IDEs die von JetBrains angeboten werden und die alle auf der IntelliJ Platform basieren. Das bedeutet nicht nur, dass sich all diese IDEs in der Verwendung und im Aussehen sehr ähnlich sind, sondern auch, dass ein Plugin, welches für die allgemeine IntelliJ Platform entworfen wurde, relativ problemlos auch für mehrere IDEs dieser Form veröffentlicht werden kann.

Zum aktuellen Zeitpunkt bietet IntelliJ in den unterschiedlichen Versionen folgende features:

Sprachen

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

Frameworks

Ein Bild, das Text, Screenshot, Dokument enthält.

Automatisch generierte Beschreibung

Build Tools

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

TODO irgendwas mit andere IDEs von Jetbrains

Src:

<https://en.wikipedia.org/wiki/IntelliJ_IDEA>

<https://www.jetbrains.com/products/compare/?product=idea-ce&product=idea>

1. Vergleich der angebotenen Funktionalität und deren Anwendbarkeit
2. Codebeispiel: Recent Changes Plugin
3. Vergleich der Code-Metriken
4. Conclusion

Referenzen

<https://code.visualstudio.com/api>

<https://github.com/microsoft/vscode-extension-samples>

<https://plugins.jetbrains.com/docs/intellij/developing-plugins.html>

<https://github.com/JetBrains/intellij-sdk-code-samples/>

<https://github.com/JetBrains/intellij-community/tree/master>

<https://github.com/google/diff-match-patch>

<https://medium.com/cacher-app/building-code-editor-plugins-a-comparison-83b5c21657fe>

<https://survey.stackoverflow.co/2023/>

<https://insights.stackoverflow.com/survey>