

Automatisierte Analyse von Clean-Code Regeln mit IDE-Plugins

Diplomanden



Rafael Fuhrer



Pascal René Schneider

Ausgangslage: Clean Code - sauber geschriebener Quelltext - ist ein elementarer Bestandteil der Softwarequalität. Man versteht darunter, dass Code lesbar, sauber strukturiert und einfach verständlich sein soll. Clean Code Prinzipien sollten bei jedem Softwareprojekt eine hohe Priorität genießen, da sie vor allem die Wartbarkeit und die Erweiterbarkeit begünstigen. Während unserer Studienarbeit haben wir einen Prototyp eines Plugins für die Visual Studio Code IDE geschrieben. Dieser war in der Lage Quelltext gegen ein begrenztes Set an Regeln zu prüfen und Verstöße in der IDE farblich hervorzuheben, damit EntwicklerInnen diese Clean Code Prinzipien von Beginn weg einhalten und aufwändige Nachbesserungen vermeiden. Im Rahmen unserer Bachelorarbeit haben wir den Prototyp wieder aufgegriffen, mit dem Ziel ihn weiterzuentwickeln und als Open Source Projekt zu veröffentlichen.

Vorgehen: Vorbereitend für diese Weiterentwicklung haben wir unser Wissen über den Clean Code Begriff durch Literaturrecherchen vertieft. Zusätzlich stellten wir für die Veröffentlichung als Open Source Projekt Recherche bezüglich Lizenzbedingungen, Beitragsrichtlinien, Verhaltenskodex und benötigte Dokumentation an. Das Plugin wurde von einer clientseitigen Erweiterung - welche sich die Ressourcen mit der Entwicklungsumgebung teilt - in eine Language Server Erweiterung mit eigenen Ressourcen umgebaut und der Umfang der Regeln die geprüft werden erhöht. Mit der Durchführung und Auswertung von Nutzertests konnten wir das Plugin um einige nutzerrelevante Funktionen erweitern. Es stehen nun eine Black- und Whitelist sowie Einstellungsmöglichkeiten zum Ein- und Ausschalten

von Regeln zur Verfügung.

Ergebnis: Das aus der Arbeit resultierende Plugin kann von jedem Entwickelnden über den Visual Studio Marketplace installiert werden. Das Projekt wurde als Open Source Projekt veröffentlicht und kann nun von jedem Interessierten weiterentwickelt werden.

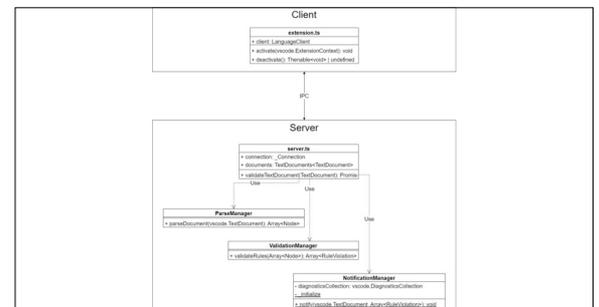
Logo des veröffentlichten Plugins

Eigene Darstellung



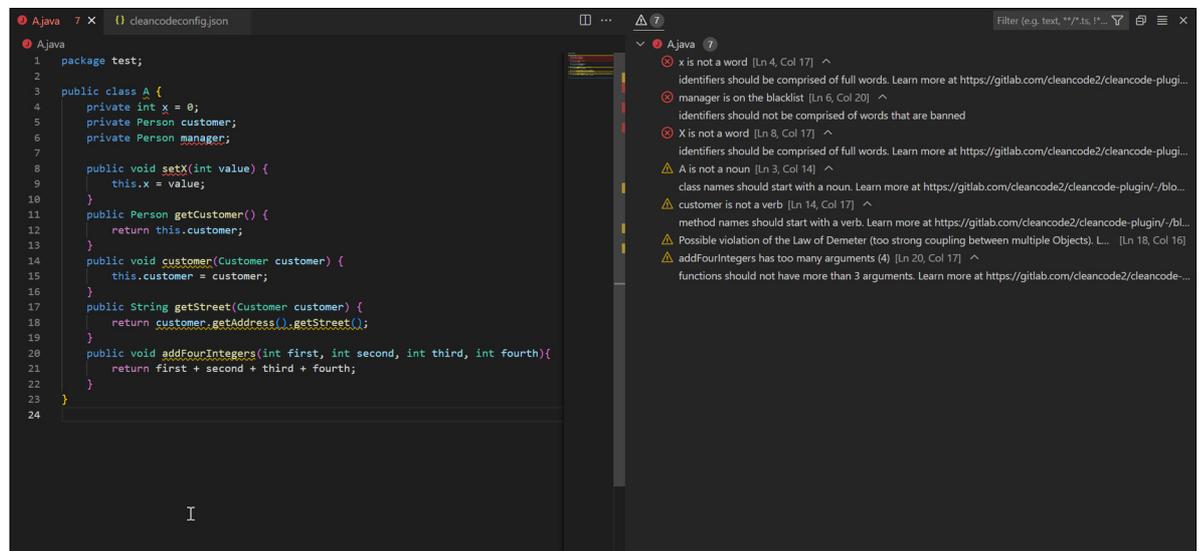
Grobarchitektur des Plugins

Eigene Darstellung



Beispiele von Regelverstößen

Eigene Darstellung



Referent
Prof. Dr. Frieder Loch

Korreferent
Dr. Michael Solfrank,
München, BY

Themengebiet
Software