

Validating the Object Calisthenics

Evaluation and Prototypical Implementation of Tool Support

Fabian Schwarz-Fritz

09.01.2014

Validating the Object Calisthenics

Evaluation and Prototypical Implementation of Tool Support

Jeff Bay's „Object Calisthenics“ is an exercise to improve the quality of object oriented code. Good object oriented code is hard to learn when coming from procedural code. Many developers think object oriented – but do they really write good object oriented software?

The rules of the Object Calisthenics are presented in *The ThoughtWorks Anthology* by Jeff Bay. The rules train developers to enhance their object oriented coding style. The calisthenics are composed of nine rules that the developer has to stick with. Behind every rule there is a purpose why the rule is important and why it leads to better object oriented code.

Usually, a developer doesn't use these rules in real world projects but applies them in short two day exercises in which he designs and implements minimalistic software with little requirements. This could be a Minesweeper or a TicTacToe game for example. These training challenges should lead the developer to write better code and be more aware of code quality in real world projects.

But when completing the training challenge the developer has to observe his own code and check if his own coding style satisfies the nine rules of the Object Calisthenics. Tool support could shorten the time of the training and furthermore guarantee that the developer sticks to the given rules.

In a student research paper, I am currently evaluating the development of tool support for the Object Calisthenics. In the course of the paper I already implemented a prototype. The tool created during the research of the paper is realized in form of an Eclipse plugin. It successfully validates the majority of Jeff Bay's rules and indicates corresponding violations.

This contribution consists of two parts.

The first part compromises the explanation of patterns and principals behind the rules. What is the architectural problem, addressed by the rules? What are the patterns and principals behind the rules? How do these patterns and principles lead to the rules?

The discussion of the challenges validating the compliance is the second and main part of this contribution. What challenges occurred when validating the source code structure? For which rules was it not possible to find an implementation, determining the validity of a rule and what detained it? In this part the implementation of at least one rule validation is exemplified.

Short Bio

Fabian Schwarz-Fritz absolviert derzeit ein Duales Studium (Praxisphasen bei der SAP AG, Walldorf, Theoriephasen an der Dualen Hochschule Karlsruhe). In den Praxisphasen arbeitet er in verschiedenen Abteilungen und leistet einen aktiven Entwicklungsbeitrag.

Hierbei sammelte er bereits Erfahrungen, die von der Entwicklung mit traditionellen SAP ERP Systemen bis hin zur Entwicklung von innovativen Lösungen mit der In-Memory-Plattform HANA reichen.

In seiner Freizeit beschäftigt er sich mit Fragen rund um das Thema Software Craftmanship und besucht Veranstaltungen für Softwareentwickler und Treffen der „Softwerkskammer Karlsruhe“.

Picture

