DHBW
Duale Hochschule
Baden-Württemberg
Karlsruhe

Evaluation and Prototypical Implementation of Tool Support to Validate the Object Calisthenics

längerer Untertitel mit genauerer Tehemenbezeichnung

Bachelorarbeit

for the certification examinations for the
Bachelor of Science

degree course Appliced Computer Science

Baden-Wuerttemberg Cooperative State University Karlsruhe

von

Fabian Schwarz-Fritz

Publication date:	November 11, 2013
Time required for processing:	12 Weeks
Matriculaiton number, course:	123 456, TINF11B2
Vocational training company:	SAP AG, Walldorf
Reviewer:	Softwareschneiderei GmbH

Eidesstattliche Erklärung

Ich versichere hiermit, dass ich meine Bachelorarbeit mit dem Thema

> *Evaluation and Prototypical Implementation of Tool Support to Validate the Object Calisthenics - längerer Untertitel mit genauerer Tehemenbezeichnung*

selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Mir ist bekannt, dass ich meine Diplomarbeit zusammen mit dieser Erklärung fristgemäß nach Vergabe des Themas in dreifacher Ausfertigung und gebunden im Sekretariat meines Studiengangs an der DHBW Karlsruhe abzugeben habe. Als Abgabetermin giltbei postalischer Übersendung der Eingangsstempel der DHBW, also nicht der Poststempel oder der Zeitpunkt eines Einwurfs in einen Briefkasten der DHBW.

Karlsruhe, den November 11, 2013

Fabian Schwarz-Fritz

Sperrvermerk

Die Ergebnisse der Arbeit stehen ausschließlich dem auf dem Deckblatt aufgeführten Ausbildungsbetrieb zur Verfügung.

Abstract

Hier bitte den Abstract Ihrer Arbeit eintragen. Der Abstract sollte nicht länger als eine halbe Seite sein. Bitte klären Sie mit Ihrem Studiengangsleiter ab, ob der Abstract in englischer oder deutscher Sprache (oder möglicherweise sogar in beiden Sprachen) verfasst werden soll.

Contents

Abbreviations

DHBW	Duale Hochschule Baden-Württemberg
OSS	Open Source Software
Sem	Semester

List of Figures

List of Tables

1 Introduction

1.1 Motivation

1.2 Goals of this report

1.3 Structure of this report

2 The Object Calisthenics

For every rule 1-9: - Explain the rule. Use quotes of paper. (research) - Good example with explanation - Bad example with explanation - Describe idea and principal behind the pattern (research) - Summarize the rule's purpose

2.1 Purpose of the Object Calisthenics

asdf

2.2 The Rules

asdf

2.2.1 Use only one level of indentation per method

asdf

2.2.2 Don't use the else keyword

asdf

2.2.3 Wrap all primitives and strings

asdf

2.2.4 Use only one dot per line

asdf

2.2.5 Don't abbreviate

asdf

2.2.6 Keep all entities small

asdf

2.2.7 Don't use any classes with more than two instance variables

asdf

2.2.8 Use first-class collections

asdf

2.2.9 Don't use any getters/setters/properties

asdf

2.3 Similarities of the rules

Categorize rules: Are there similarities from perspective of principle? ??? Together with next chapter?

2.4 Precedence of the rules

Own estimation: what's the most important rule? What do I think? What does the author think? Reason with descriptions and examples given

2.5 Conclusion and outcome of the rules

Give an outcome of the rules.

3 Evaluation of tool support

3.1 ??? Generatal tool introduction

What is a tool? Why do tools help? What makes tools strong? Why do tools matter for developers?

3.2 Introduction

Possible outcome of tool support for the OC's?

3.3 Abstract syntax tree

Describe ast generally. Say that Eclipse provides types representing the parts of code syntax. This is seen as given. Refer to other references. Describe shortly how it is possible to do an AST validation with eclpise. Say that: "Eclipse" terms and standart terms are use. These are not further described in this paper. Give reference for questions about "parameter", "type", "class", "expression" or "statement". Say that the validaiton in the next section is exactly implemented as described. One (???) example is shown in the Prototype chapter.

3.4 Evaluation of rule validation? TODO better title

Say that the priorization of the rules and the "ranking" is given in the end. This chapter is "rule specific", even if the next subsections refer to each other.

Foreach: - Similarities found out in description may be similar in this validation? - Cathegorize the rules in groups form a validation perspecitve - Use examples given in description chapter to describe the typical structure of the rule. - Explane "the positive case": What is the positive structure, satisfying the rule - What checks have to be done for a possible validation - –> solution found/no solution found

- Be self-critical: Now, were a solution is found (or not), describe the problems that occur with the described implementation

3.4.1 Use only one level of indentation per method

3.4.2 Don't use the else keyword

3.4.3 Wrap all primitives and strings

3.4.4 Use only one dot per line

3.4.5 Don't abbreviate

3.4.6 Keep all entities small

3.4.7 Don't use any classes with more than two instance variables

3.4.8 Use first-class collections

3.4.9 Don't use any getters/setters/properties

3.5 Prioritize

Give a summary on how hard it was to implement the rules

4 Prototypical implementation of tool support

4.1 Prototype requirements

Describe requirements for the prototype

4.2 Architecture

Describe overall architecture (???packages)

4.3 An example of rule validation

One example implementation of one rule validation

4.4 The resulting prototype

Show screenshot and describe UI. What is possible, what is not possible. What are ideas that are still out there? How could the product improve?

5 Conclusion

5.1

5.2

Bibliography

[FoBa03] Foo, John; Bar, Belinda: *Titel : Untertitel*, Verlagsort: Verlag, Jahr der Auflage. S. 10-20

[Le01] Autor Name: *Titel des Buches*, New York: Penguin Books, 2001