



Fakultät für Informatik

Studiengang Software- und Systems-Engineering

# Erkennung von Design Patterns in Quellcode durch Machine Learning

Master Thesis

von

Mehmet Aslan

Datum der Abgabe: tt.mm.jjjj

Erstprüfer: Prof. Dr. Marcel Tilly

Zweitprüfer: Prof. Dr. Kai Höfig

#### EIGENSTÄNDIGKEITSERKLÄRUNG / DECLARATION OF ORIGINALITY

Hiermit bestätige ich, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe. Die Stellen der Arbeit, die dem Wortlaut oder dem Sinn nach anderen Werken (dazu zählen auch Internetquellen) entnommen sind, wurden unter Angabe der Quelle kenntlich gemacht.

*I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.*

Rosenheim, den tt.mm.jjjj

Vor- und Zuname

# Kurzfassung

text

Schlagworte:

# **Inhaltsverzeichnis**

<b>A</b>	<b>Erstes Kapitel des Anhangs</b>	<b>2</b>
	<b>Literaturverzeichnis</b>	<b>3</b>

# **Abbildungsverzeichnis**

# **Tabellenverzeichnis**

# Einführung

## Einführung in Design Patterns

Bei der Entwicklung von Software-System stößt man auf Probleme und Situationen, denen man bereits in der gleichen Form oder in einer ähnlichen Variation entgegenkommen ist. Daher tendiert man, einen bereits bekannten Lösungsansatz hier wieder einzusetzen. Die Idee der Wiederverwendbarkeit von bekannten Lösungsansätzen für wiederaufkehrende zu lösende Probleme in der Software-Entwicklung von Software-Systemen stellen den Kern der Entwurfsmuster oder *Design Patterns* dar. Allgemeiner betrachtet bilden Entwurfsmuster eine Art Lösungsblaupause für wiederaufkehrende Probleme dar, die für die jeweilige Situation angepasst werden. *Design Pattern - Elements of Reusable Object-Oriented Software* von Gamma et. al [Gam94] ist wohl das bekannteste Werk, das sich mit dieser Thematik auseinandersetzt. Normalerweise werden Design Patterns verwendet, um abstrakte Probleme in der Software-Architektur von objekt-orientierten Programmiersprachen zu lösen, die sich mit der Kreation, Struktur oder Verhalten auseinandersetzen. Obwohl Design Patterns als bekannte und erprobte Blaupause für wiederaufkehrende Probleme in Software-Systemen eingesetzt werden können, bringen Design Patterns Konsequenzen für die zu betrachtende Situation mit sich, die in Erwägung gezogen werden sollten. Diese sind meistens erhöhte Zeit- und Speicherkomplexität als Austausch für eine bessere Flexibilität im Design des Software-Systems, so dass dessen Adaptierbarkeit für neue oder sich ändernde Anforderungen sichergestellt werden können.

Die Beschreibung von Design Patterns sind detailliert und beinhalten unter anderem dessen Namen, Absicht, Motivation, Anwendbarkeit, Struktur, Teilnehmer und deren Zusammenarbeit und andere Informationen. Die Semantik des Design Patterns inkludieren die Absicht, Motivation und Anwendbarkeit, die angeben, was das Entwurfsmuster macht, warum dieses benötigt wird und wo es sinnvoll eingesetzt werden kann. Die Teilnehmer reflektieren deren Natur als Blaupause, da diese Rollen darstellen, die die Klassen im Kontext des Design Patterns darstellen, während die Struktur und die Zusammenarbeit zwischen den Teilnehmern die Interaktionen zwischen diesen beschreibt. Diese Menge an Informationen ist in den Entwurfsmustern und deren Beschreibung enkodiert und wird durch Software-Entwickler in das Software-System während dessen Implementierung eingeflochten. Der Einsatz von Design Patterns ist auf Design-Entscheidung während der Entwicklung des Software-Systems zurückzuführen und des Öfteren wird diese und die jeweiligen Hintergrundgedanken nicht weiter dokumentiert. Davon ebenfalls betroffen sind Anpassungen und konkrete Implementierungsdetails des Entwurfsmusters. Schlussendlich tauchen diese Einzelheiten in Quellcode des Software-Systems ab. Das Wiederfinden dieser enkodierten Information im Quellcode für die Zwecke der Weiterentwicklung und Wartung ist der Hauptmotivator für die Erkennung von Entwurfsmustern.

## **A Erstes Kapitel des Anhangs**

Wenn Sie keinen Anhang benötigen, dann bitte einfach rausnehmen.



# Literaturverzeichnis

- [Gam94] E. Gamma, R. Helm, R. Johnson und J. M. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional, 1 Aufl., 1994.