

Die Quadratur des Klassendesigns

Karl Nieratschker, SKT Nieratschker

Abstract

Der Entwurf von Klassen für ein objektorientiertes Softwaresystem gehört zu den anspruchsvollsten Aufgaben eines Softwareentwicklers. Für ein gutes Klassendesign wird viel Erfahrung und Kreativität benötigt, insbesondere dann, wenn es um Embedded Systeme geht. Die Klassen müssen nicht nur alles enthalten, was die aktuelle Anwendung benötigt, sondern sollten gleichzeitig auch optimal wiederverwendbar und erweiterbar sein, damit das Rad nicht immer wieder neu erfunden werden muss. Gleichzeitig sollten sie auch leicht verständlich sein, damit neue Mitarbeiter sich schnell einarbeiten können und die Wartung nicht zum Albtraum wird. Und nicht zuletzt müssen auch noch Speicherplatz- und Laufzeitanforderungen erfüllt werden. Angesichts dieser Herausforderungen zeigt dieser Vortrag nicht nur, wie die vielen Fragen, die sich in diesem Zusammenhang stellen, mithilfe von klassischen Designprinzipien wie z.B. DRY, YAGNI und SOLID beantwortet werden können, sondern auch, was dabei zu beachten ist.

Inhalt

Der Vortrag gibt Antworten auf viele Fragen, die generell für den Entwurf von objektorientierten Softwareanwendungen wichtig sind, wie z.B.:

- Was sind wichtige Ziele für einen guten Softwareentwurf?
- Was ist Sinn und Zweck einer Klasse?
- Wie findet man Klassen?
- Welche Rolle spielt die Namensgebung in diesem Zusammenhang?
- Welchen Funktionsumfang sollte eine Klasse besitzen?
- Was kostet eigentlich eine Klasse?
- Welche Methoden sollte eine Klasse haben, und wie findet man diese?
- Wie geht man am besten mit Typvarianten um?
- Wie gestaltet man die Zusammenarbeit von Klassen?
- Wie sorgt man für Flexibilität und gute Anpassbarkeit?
- ...

Nutzen

Die Erfahrung zeigt, dass der Entwurf von Klassen auch bei noch so viel Erfahrung nie zu einer Routineangelegenheit wird, denn bei jedem neuen Projekt gibt es wieder neue Gegebenheiten und Anforderungen. Dieser Vortrag stellt einen Leitfaden dar, der es dem Teilnehmer nicht nur ermöglicht, seine Software nach anerkannten und praxiserprobten Designprinzipien zu entwerfen, sondern insbesondere auch, Designfehler zu vermeiden, die sich häufig erst sehr viel später auswirken und deshalb, wenn überhaupt, nur mit viel Aufwand wieder behoben werden können.

Autor:

Basierend auf einem Studium der technischen Informatik verfügt Karl Nieratschker über eine mehr als 30-jährige Erfahrung im Bereich der Embedded-Programmierung und betriebssystemnahen Softwareentwicklung. Als selbständiger Trainer, Softwareberater und Coach unterstützt er seit 1999 Kunden bei der Einführung neuer Softwaretechnologien. Besondere Schwerpunkte dabei sind „Objektorientierte Programmierung in Embedded Systemen“, sowie „Multithreading-“ und „Multicore-Programmierung“.

Internet: www.skt-nieratschker.de

Email: office@skt-nieratschker.de