

Diplomarbeitsvortrag

# **Kleine Refactoring-Muster**

Marko Schulz  
ms@datenreisender.de

<http://datenreisender.de/DA/>

# Überblick

- Wieso? (davor)
- Fragen

---

- Muster
  - Was sind Muster?
  - Muster in der Softwaretechnik
- Refactorings

---

- Refactoring-Muster
- Kleine Refactoring-Muster
- Anwendung kleiner Refactoring-Muster

---

- Fragen + Antworten
- Wieso? (danach)

# Wieso? (davor)

- Große Refactorings
  - Thema von Martins Doktorarbeit
  - Planen (und Durchführen)
- Wie zerlegt man (große) Refactorings?
- Wie wurden Refactorings zerlegt?  
(z.B. von Fowler)
- wiederkehrende Muster?

# Fragen

- Paßt der Ausdruck „Muster“ von der Größe?
- Kann man Refactoring-Muster zu Recht als Muster bezeichnen?
- Findet man mehrere kleine Refactoring-Muster?
- Was wurde bereits geschrieben?  
Was ist an kleinen Refactoring-Mustern neu?

# Überblick

- Wieso? (davor)
- Fragen

---

- **Muster**
  - Was sind Muster?
  - Muster in der Softwaretechnik
- **Refactorings**

---

- Refactoring-Muster
- Kleine Refactoring-Muster
- Anwendung kleiner Refactoring-Muster

---

- Fragen + Antworten
- Wieso? (danach)

# Was sind Muster?

- Mehrfach belegter Begriff
- Konzept basiert auf den Ideen des Architekten Christopher Alexander
  - Architektur mit Mustern beschreiben
  - 1979 - Vol. 1: The Timeless Way of Building
  - 1977 - Vol. 2: A Pattern Language
  - 1975 - Vol. 3: The Oregon Experiment

# Was sind Muster? (2)

„Jedes Muster beschreibt zunächst ein in unserer Umwelt immer wieder auftretendes Problem, beschreibt dann den Kern der Lösung dieses Problems, und zwar so, daß man diese Lösung millionenfach anwenden kann, ohne sich je zu wiederholen.“

(aus: *Eine Muster-Sprache: Städte, Gebäude, Konstruktionen*)

- Problem
- Lösung
- Allgemein anwendbar
- Kräfte
- Name
- Sprache

# Muster in der Softwaretechnik

- 1987 das erste Mal in einem Papier mit Software in Zusammenhang gebracht
  - Entwurfsmuster
  - Mehr als Entwurfsmuster
  - Merkmale ähnlich wie bei Alexander
- |                       |                    |
|-----------------------|--------------------|
| • Problem             | • (Kräfte) Kontext |
| • Lösung              | • Name             |
| • Allgemein anwendbar | • (Sprache)        |

# Was sind Refactorings?

- Restrukturierungen
- Fast zeitgleich mit Mustern populär geworden
  - William Opdyke
  - Extreme Programming
  - Martin Fowler

„A change made to the internal structure of software to make it easier to understand and cheaper to modify without changing its observable behavior.“ (Martin Fowler)

- Was sind *keine* Refactorings?

# Überblick

- Wieso? (davor)
- Fragen

---

- Muster
  - Was sind Muster?
  - Muster in der Softwaretechnik
- Refactorings

---

- Refactoring-Muster
- Kleine Refactoring-Muster
- Anwendung kleiner Refactoring-Muster

---

- Fragen + Antworten
- Wieso? (danach)

# Refactorings als Muster betrachtet

- Darf man das? Einige machen es.

## *Definition:* **Refactoring-Muster**

Ein Refactoring, welches den Charakter von einem Muster besitzt und als solches beschrieben wird.

- Fowlers Refactorings als Muster
- Nicht alle Refactorings sind Muster

# Beispiele für kleine Refactoring-Muster

- „Typische“ Schritte in Refactorings:
  - Teile und herrsche
  - Weiterleitung einführen
  - Kopieren
  - Finde alle
  - Finde Schlüsselement
  - Struktur aufbauen
  - Aufräumen
  - Deprecate

# Wann sind Refactoring-Muster „klein“?

- Ausführungsdauer
- Beibehaltung des Verhaltens
- Einbettung von Refactorings

***Definition:* Eingebettete und umgebende Refactorings**

Wenn mehrere Refactorings zu einem größeren Refactoring kombiniert werden, so sind sie in dieses umgebende Refactoring eingebettet.

# Wann sind Refactoring-Muster „klein“? (2)

- Zweck des Refactoring-Musters
  - Kleine Refactoring-Muster: nach innen gewandt
  - Größere Refactoring-Muster: nach außen

## *Definition:* **Kleines Refactoring-Muster**

Ein Refactoring-Muster, welches in Refactorings eingebettet verwendet wird, um die dort üblichen Probleme zu lösen.

# Das kleine Refactoring-Muster „Finde alle“

- **Problem:** Eine Änderung, erfordert andere Änderungen, welche quer durch den Quelltext verstreut sind.
- **Lösung:** Ermittle ein passendes Suchkriterium und arbeite alle Fundstellen ab.
- Wird unterstützt durch „Finde Schlüsselement“ und „Deprecate“
- Unterstützt „Teile und herrsche“

# Anwendung kleiner Refactoring-Muster

- Triviales „Move Method“:
  - Methode von einer Klasse zu einer anderen verschieben.
  - Jede Stelle, an der die Methode gerufen wird, umstellen.

# Anwendung kleiner Refactoring-Muster

- „Move Method“ als Refactoring (orientiert an Fowler):
    - Leere Methode in der Zielklasse erzeugen Struktur aufbauen
    - Quelltext aus alter Methode kopieren und anpassen Kopieren
    - Quelltext in der alten Methode durch eine Weiterleitung ersetzen Weiterleitung einführen
    - Aufrufe der alten Methode (teilweise) auf die neue Methode umstellen Finde alle
    - Alte Methode entfernen, so nicht mehr verwendet Aufräumen
- Teile und herrsche

# Überblick

- Wieso? (davor)
- Fragen

---

- Muster
  - Was sind Muster?
  - Muster in der Softwaretechnik
- Refactorings

---

- Refactoring-Muster
- Kleine Refactoring-Muster
- Anwendung kleiner Refactoring-Muster

---

- Fragen + Antworten
- Wieso? (danach)

# Fragen + Antworten

- ~~Paßt der Ausdruck „Muster“ von der Größe?~~  
Quatsch!
- Kann man Refactoring-Muster zu Recht als Muster bezeichnen? ✓ Ja.
- Findet man mehrere kleine Refactoring-Muster? ✓ Ja.
- Was wurde bereits geschrieben?  
Was ist an kleinen Refactoring-Mustern neu? ✓ Ja.

# Wieso? (danach)

- Was bringt das alles?
- Wissenschaftlich
  - Konzept von Refactoring-Mustern
  - Konzept von kleinen Refactoring-Mustern
  - Verständnis über den Aufbau von Refactorings
- Pragmatisch / Arbeiten mit Refactorings
  - Refactorings verfassen
  - Refactorings anpassen
  - Werkzeuge zur Unterstützung dieser Muster

**Vielen Dank für die Aufmerksamkeit!**