

STRATEGIE

Lars Briem

(briem.lars@googlemail.com)

Duale Hochschule Baden Württemberg - Standort Karlsruhe

Strategie

- ▶ Definition einer Familie bzw. Sammlung von Algorithmen
- ▶ Algorithmen austauschbar machen
- ▶ Jeder Algorithmus gekapselt
- ▶ Algorithmus kann variieren, unabhängig vom Anwender

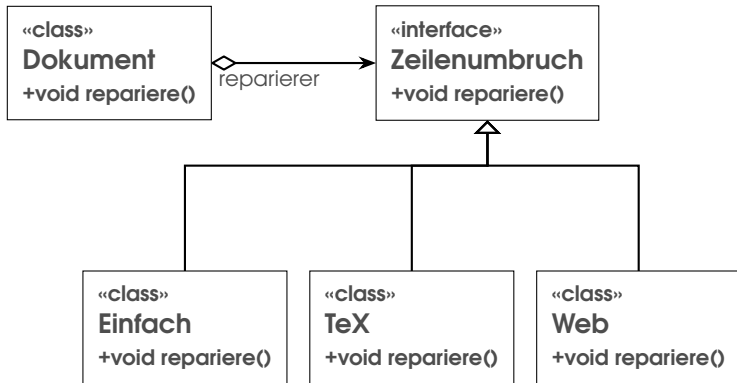
Strategie – Einordnung

- ▶ Objektbasiertes Verhaltensmuster
- ▶ Viele Ausprägungen
- ▶ Auch bekannt als Policy

Strategie – Motivation

- ▶ Integration eines Algorithmus bzw. Verhaltens verkompliziert den Anwender
 - ▶ Anwender wird unnötig größer
- ▶ Ein Algorithmus bzw. Verhalten während der Laufzeit abhängig von verschiedenen Bedingungen
 - ▶ Nicht immer alle zur gleichen Zeit notwendig
- ▶ Verhalten ist nur schwer austauschbar, wenn direkt in Anwender integriert

Strategie – Beispiel



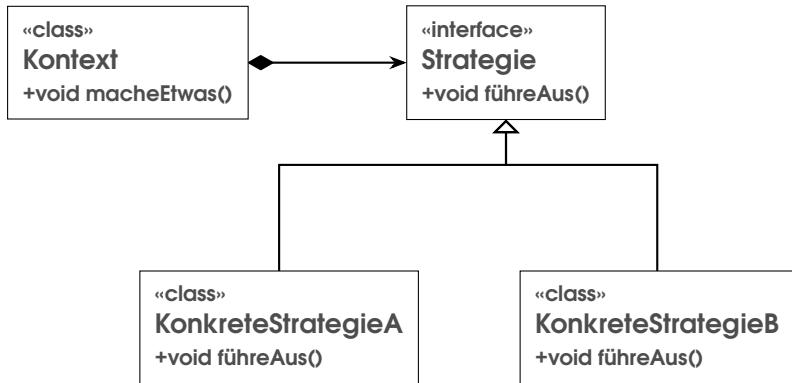
Strategie – Anwendung

- ▶ viele ähnliche Klassen unterscheiden sich nur im Verhalten
 - ▶ 1 Klasse mit unterschiedlichem Verhalten konfigurieren
- ▶ verschiedene Varianten eines Algorithmus notwendig
 - ▶ Abwägung zwischen Zeit-/Speicherverbrauch

Strategie – Anwendung

- ▶ Algorithmus verwendet komplexe Datenstruktur, die für Anwender unwichtig ist
- ▶ Klasse verwendet mehrfach ähnliche Konditionalstrukturen
 - ▶ Zusammengehörendes Verhalten in Strategie auslagern

Strategie – Struktur



Strategie – Akteure

- ▶ Strategie
 - ▶ Definiert gemeinsames Interface für alle unterstützten Algorithmen
 - ▶ Kontext arbeitet mit bzw. kennt nur dieses Interface
- ▶ KonkreteStrategie
 - ▶ Implementiert Strategie
 - ▶ Implementiert konkreten Algorithmus

Strategie – Akteure

- ▶ Kontext
 - ▶ Konfiguriert konkrete Strategie
 - ▶ Verwaltet Referenz zu Strategie
 - ▶ Definiert evtl. Interface für Strategie zum Zugriff auf Daten im Kontext

Strategie – Interaktion der Akteure

- ▶ Kontext gibt alle notwendigen Informationen beim Aufruf dem Algorithmus mit
- ▶ Kontext gibt sich selbst beim Aufruf mit
 - ▶ KonkreteStrategie holt sich Informationen von Kontext

Strategie – positive Auswirkungen

- ▶ Einfache Wiederverwendung von Algorithmen
- ▶ Einfache Erweiterbarkeit für neue Algorithmen (OCP)
- ▶ Dynamische Alternative zu Ableitung bzw. Objekthierarchien
 - ▶ Beliebige Kombination von Strategien möglich
 - ▶ Standardverhalten im Fehlerfall

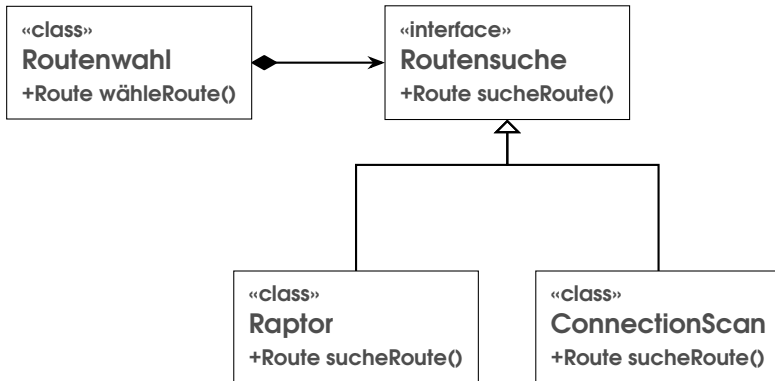
Strategie – positive Auswirkungen

- ▶ Alternative für Konditionalstruktur zur Auswahl von Verhalten
- ▶ Anwender kann Ressourcenverbrauch wählen durch unterschiedliche Algorithmen

Strategie – negative Auswirkungen

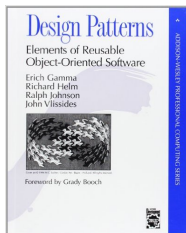
- Anwender muss unterschiedliche Strategien kennen
 - ⇒ Standard vorgeben für zu optionaler Wahl der Strategie
- Zusätzliche Kommunikation zwischen Kontext und Strategie
- Erhöhte Anzahl von Objekten

Strategie – Beispiel



Strategie – Zusammenfassung

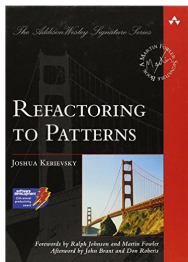
- ▶ Dynamische Änderung des Verhaltens
- ▶ Ähnlich zu Dekorator
 - ▶ Aber Kontext kennt das aktuelle Verhalten
- ▶ Strategie wird oft als Fliegengewicht implementiert
 - ▶ Teilung von gemeinsamen variablen Daten



- ▶ Design Patterns
 - ▶ Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides
 - ▶ Addison-Wesley
 - ▶ ISBN: 978-0201633610

Weitere Infos

- ▶ Entwurfsmuster auf YouTube
 - ▶ John Lindquist erklärt Entwurfsmuster mit StarCraftII
 - ▶ <https://www.youtube.com/playlist?list=PL8B19C3040F6381A2>



- ▶ Refactoring to Patterns
 - ▶ Joshua Kerievsky
 - ▶ Addison-Wesley
 - ▶ ISBN: 978-0321213358