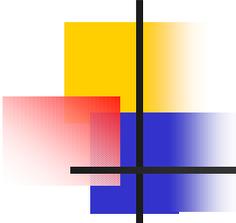


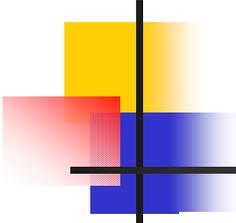
Iterator-Pattern

von Lars Münzberg



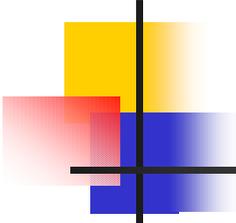
Inhalt

1. Einordnung
2. Motivation
3. Anwendbarkeit
4. Struktur eines Iterator-Pattern
5. Teilnehmer
6. Konsequenzen
7. Implementierung
8. Verwandte Muster
9. Verwendungsbeispiele
10. Quellen



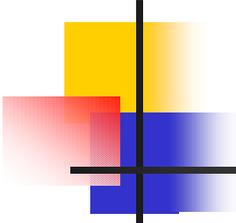
1. Einordnung

- Entwurfsmuster
 - (+) Robustheit des Entwurfes
 - (+) höherer Wiederverwendungsgrad
 - (+) Kommunikation
 - gemeinsames Vokabular
 - höhere Abstraktion
 - leichtere Dokumentation und Verständnis
 - (-) teilweise höhere Kosten



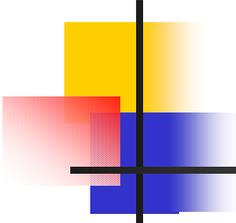
1. Einordnung

- objektbasiert
- Verhaltensmuster
 - Algorithmen
 - Zuweisung von Zuständigkeiten zu Objekten
- Alias: Cursor



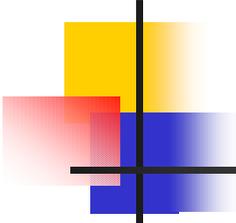
2. Motivation

- zusammengesetztes Objekt (Aggregat) ist Datenhaltung
- Objekt soll Möglichkeit des Elementzugriffs aufweisen
- unabhängig von der inneren Struktur
- Traversierung des Objekt auf verschiedene Arten
- Schnittstelle sollte übersichtlich sein
- Objekt soll gleichzeitig mehrfach traversiert werden können



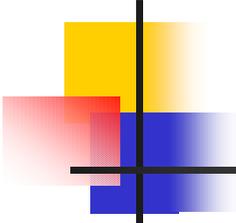
2. Motivation

- Grundidee des Konzepts:
 - Zuständigkeit für den Zugriff
 - Funktionalität zur Traversierungaus aggregiertem Objekt in Iterator-Objekt verlagert



2. Motivation

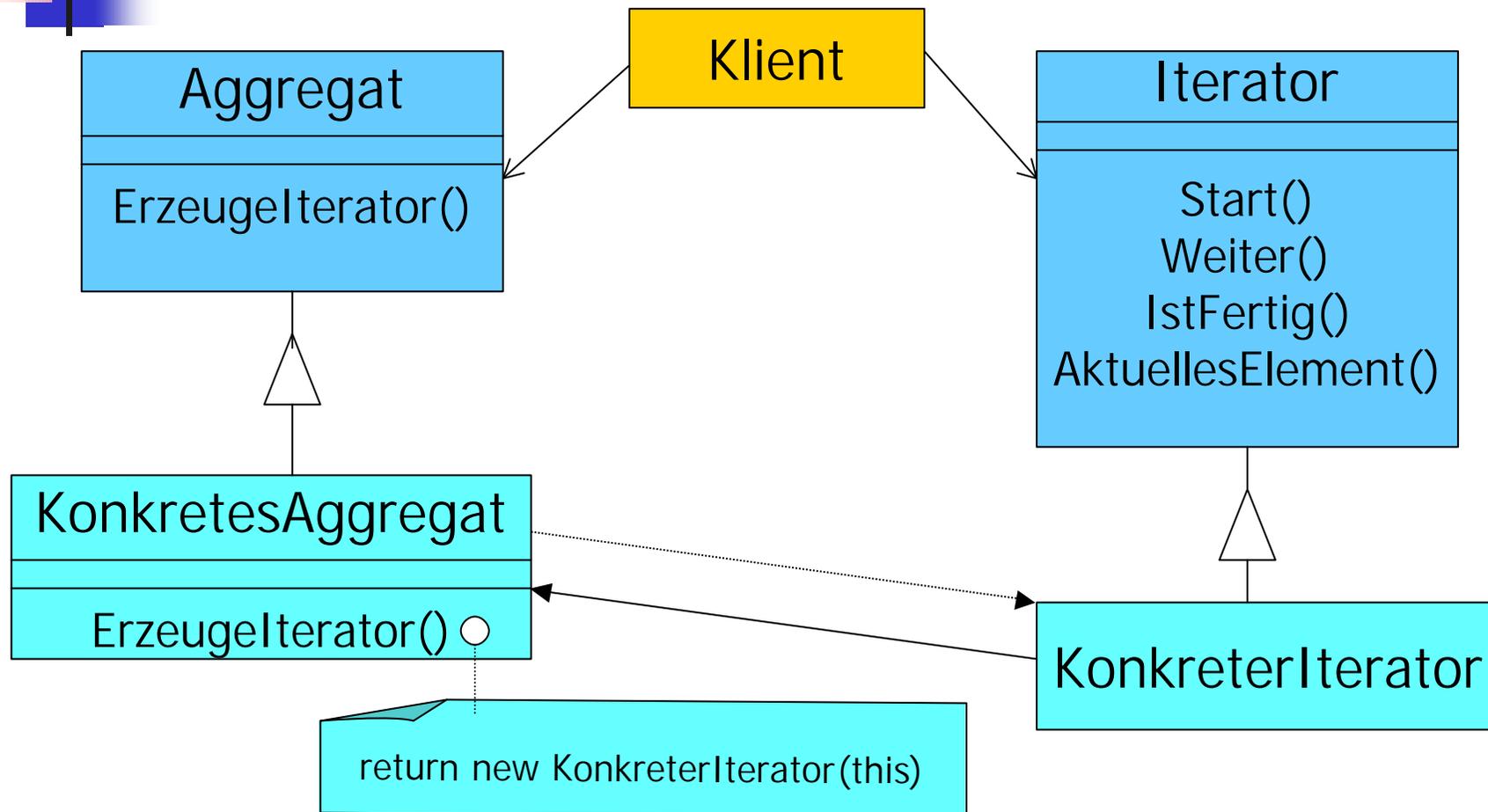
- abstrahieren von Art und Weise
 - Zugriff auf ein Objekt des Aggregats
 - Aggregation zu traversieren
- schirmt Klienten von inneren Struktur des Aggregats ab
- man will Menge von Zugriffsmechanismen definieren, also Kapselung des variierenden Konzepts erforderlich

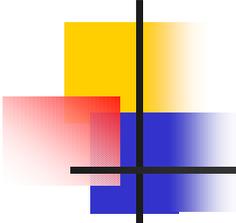


3. Anwendbarkeit

- sequenziellen Zugriff auf Aggregat zu ermöglichen
- ohne innere Struktur offenzulegen
- mehrfache gleichzeitige Traversierung
- einheitliche Schnittstelle

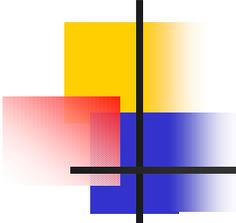
4. Struktur eines Iterator-Pattern





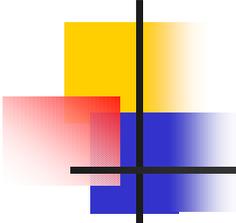
5. Teilnehmer

- Iterator:
 - Schnittstelle für den Zugriff auf und zur Traversierung von Elementen
- KonkreterIterator:
 - implementiert die Iterator-Schnittstelle
 - verwaltet aktuelle Position der Traversierung
- Aggregat:
 - Schnittstelle zum Erzeugen eines Iterator-Objekts
- KonkretesAggregat:
 - implementiert die Aggregat-Schnittstelle



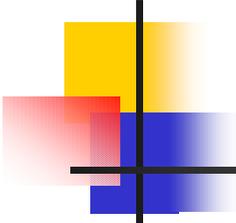
6. Konsequenzen

- Variation in der Art der Traversierung eines Aggregats
- vereinfacht die Aggregatschnittstelle
- ein Aggregat kann gleichzeitig mehrfach traversiert werden
- Kapselung gewinnt man Flexibilität und Wiederverwendbarkeit



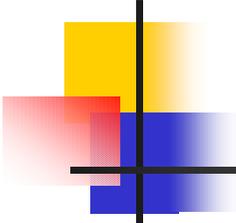
7. Implementierung

- externe (flexibler), interne (einfacher) Iteratoren
- Iteratoralgorithmus nicht im Iterator sondern im Aggregat realisiert
⇒ Cursor
- robuste Iteratoren
- Null-Iterator



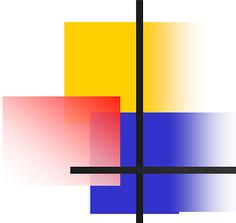
8. Verwandte Muster

- Kompositum
 - ist rekursive Struktur
- Fabrikmethode
 - polymorphe Iteratoren nutzen diese
 - ein Objekt der richtigen Iteratorunterklasse zu erzeugen
- Memento



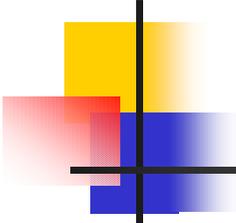
9. Verwendungsbeispiele

- Smalltalk: Collection Klassen
- Java: Enumerator



10. Quellen

- Gamma, Helm, Johnson, Vlissides: „Entwurfsmuster“
- Fraunhofer Institut: Informations- und Datenverarbeitung
- Vlissides: „Entwurfsmuster anwenden“



Iterator-Pattern

Vielen Dank für Ihre
Aufmerksamkeit.