Flyweight-Pattern
Strukturmuster

Yassine Bensaleh, Andrea Engel,
Rene Fischer, Sascha Görnert,
Niko Lockenvitz, Julian Rolle

Mannheim, den 16.01.2020

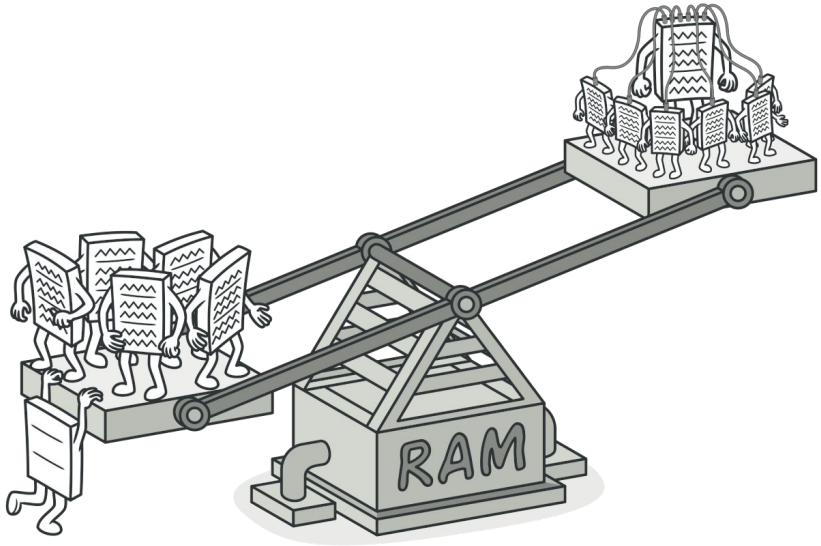
DHBW
Duale Hochschule
Baden-Württemberg

Verwendung

- deutsch: Fliegengewicht
- Speicheroptimierung
- Erzeugen von vielen Objekte
- Objekte teilen sich viele Informationen
- normale Implementierung

  → unnötig (viel) Speicher

- durch Teilung: Minimierung des Speichers



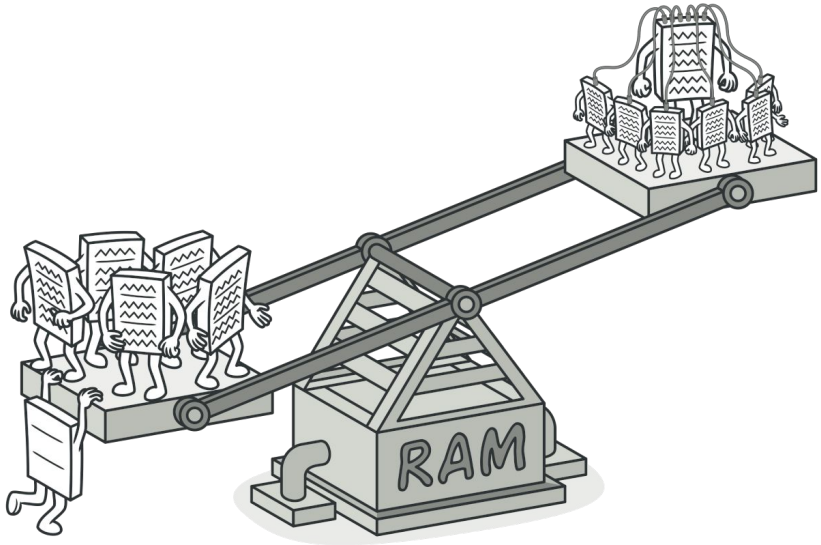
Quelle:
https://refactoring.guru/images/patterns/content/flyweight/flyweight-2x.png

Verwendung

- komplexe Objekte → kostet Performance
- Objekte teilen sich Eigenschaften
    - intrinsisch (nur wirklich)

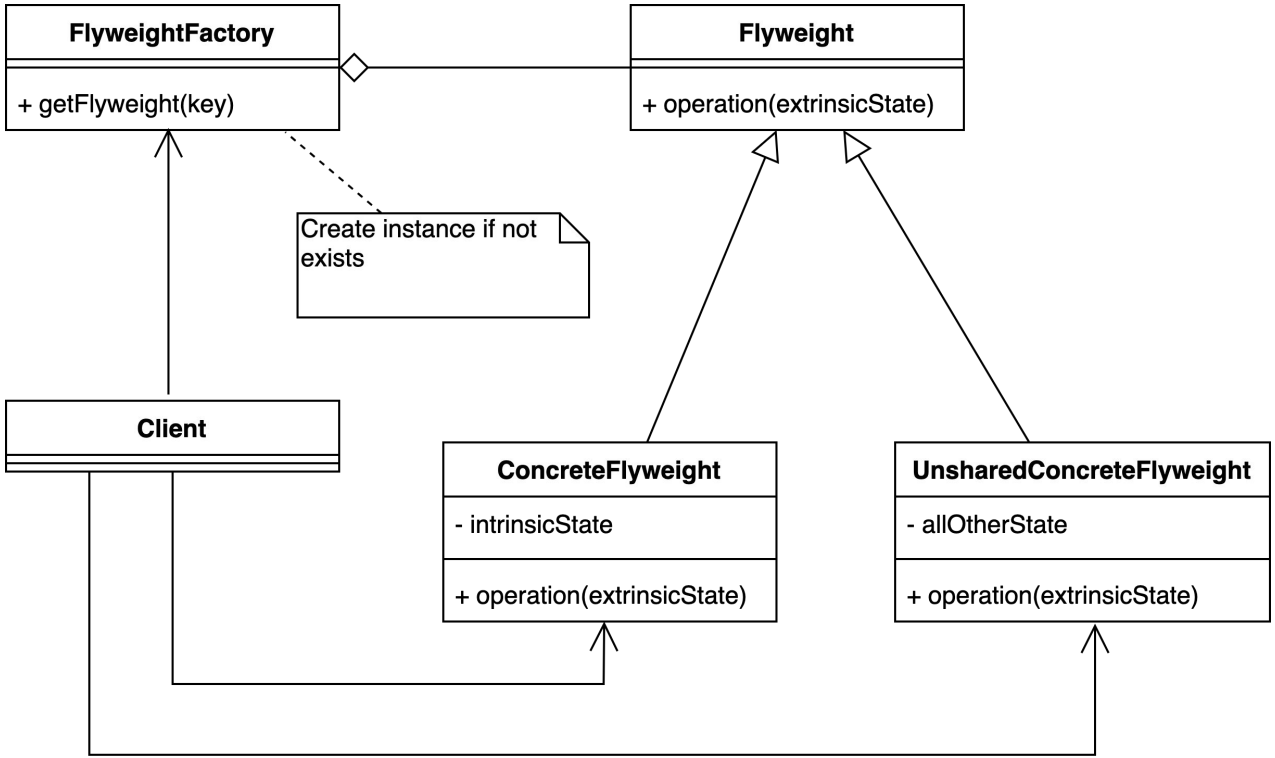
      → Baumart z.B. Tanne
    - extrinsisch (nur Merkmale)

      → Position z.B. (50 | 20)
- Kontext Smart-Home: Lampen, Schalter



Quelle:
https://refactoring.guru/images/patterns/content/flyweight/flyweight-2x.png

UML



Quelle:
Darstellung mittels: https://draw.io

Codebeispiel

Vorteile

- Speicheroptimierung
- Performance
- keine Duplikate

Nachteile

- Code wird komplizierter
- Veränderung Methode
- CPU-Zeit

  → Suchen

  → Neuberechnung aller Objekte

Quellen

- https://refactoring.guru/design-patterns/flyweight, 12.01.2020
- https://de.wikipedia.org/wiki/Fliegengewicht_(Entwurfsmuster), 10.01.2020
- https://www.tutorialspoint.com/design_pattern/flyweight_pattern.htm, 15.01.2020
- https://sourcemaking.com/design_patterns/flyweight, 15.01.2020
- https://www.baeldung.com/java-flyweight, 15.01.2020