

Kompositum

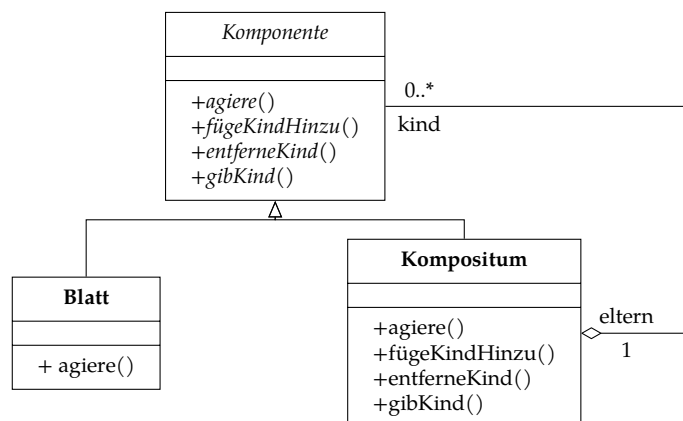
Weiterführende Literatur:

- Wikipedia-Artikel „Kompositum (Entwurfsmuster)“
- Gamma u. a., *Design Patterns CD*, Seite 139-147
- Eilebrecht und Starke, *Patterns kompakt*, Seite 102-104
- Siebler, *Design Patterns mit Java*, Kapitel 14, Seite 173

Das Kompositum

Das Kompositum (englisch *composite* oder *whole-part*) ist ein Entwurfsmuster aus dem Bereich der Softwareentwicklung. Das Kompositionsmuster (*composite pattern*) wird angewendet, um *Teil-Ganzes-Hierarchien* zu repräsentieren, indem Objekte zu *Baumstrukturen* zusammengefügt werden. Die Grundidee des Kompositionsmusters ist, in einer *abstrakten Klasse* sowohl *primitive Objekte* als auch ihre *Behälter* zu repräsentieren. Somit können sowohl einzelne Objekte als auch ihre Kompositionen *einheitlich behandelt* werden.¹

Ein anderes Beispiel sind die *Klassendefinitionen* der *grafischen Benutzeroberfläche von Java*. Alle Elemente wie Schaltflächen und Textfelder sind *Spezialisierungen* der Klasse *Component*. Die Behälter für diese Elemente sind aber ebenfalls *Spezialisierungen* derselben Klasse. Mit anderen Worten: Alle Standardelemente werden wesentlich durch eine einzige (Kompositum-)Klasse definiert.²



Details zum Nachlesen: Klett: Informatik 4, S.27 - 29

Zweck

Das Composite-Muster ermöglicht die Gleichbehandlung von Einzelelementen und Elementgruppierungen in einer verschachtelten Struktur (z. B. Baum), so dass aus Sicht des Clients keine explizite Unterscheidung notwendig ist.³

¹Qualifizierungsmaßnahme Informatik: Algorithmen und Datenstrukturen 4.

²Wikipedia-Artikel „Kompositum (Entwurfsmuster)“.

³Eilebrecht und Starke, *Patterns kompakt*, Seite 102.

Das Kompositum (englisch *composite* oder *whole-part*) ist ein Entwurfsmuster aus dem Bereich der Softwareentwicklung. Das Kompositionsmuster (*composite pattern*) wird angewendet, um *Teil-Ganzes-Hierarchien* zu repräsentieren, indem Objekte zu *Baumstrukturen* zusammengefügt werden. Die Grundidee des Kompositionsmusters ist, in einer *abstrakten Klasse* sowohl *primitive Objekte* als auch ihre *Behälter* zu repräsentieren. Somit können sowohl einzelne Objekte als auch ihre Kompositionen *einheitlich behandelt* werden.⁴

Ein anderes Beispiel sind die *Klassendefinitionen* der *grafischen Benutzeroberfläche von Java*. Alle Elemente wie Schaltflächen und Textfelder sind *Spezialisierungen der Klasse Component*. Die Behälter für diese Elemente sind aber ebenfalls *Spezialisierungen derselben Klasse*. Mit anderen Worten: Alle *Standardelemente* werden wesentlich durch eine einzige (Kompositum-)Klasse definiert.⁵

Literatur

- [1] Karl Eilebrecht und Gernot Starke. *Patterns kompakt. Entwurfsmuster für effektive Softwareentwicklung*. 2019.
- [2] Erich Gamma u. a. *Design Patterns CD. Elements of Resuable Object-Oriented Software*. 1995.
- [3] *Qualifizierungsmaßnahme Informatik: Algorithmen und Datenstrukturen 4. Listen*. https://www.studon.fau.de/file2601468_download.html, https://www.studon.fau.de/file2861024_download.html.
- [4] Florian Siebler. *Design Patterns mit Java*. 1. Aufl. Hanser, 2014. ISBN: 978-3-446-44111-8.
- [5] *Wikipedia-Artikel „Kompositum (Entwurfsmuster)“*. [https://de.wikipedia.org/wiki/Kompositum_\(Entwurfsmuster\)](https://de.wikipedia.org/wiki/Kompositum_(Entwurfsmuster)).

⁴Qualifizierungsmaßnahme Informatik: Algorithmen und Datenstrukturen 4.

⁵Wikipedia-Artikel „Kompositum (Entwurfsmuster)“.