**Klassendiagramme**

|  |
| --- |
| Klassenname |
| Klassenattributname: Datentyp (z.B. String)  Klassenattributname: Datentyp |
| Methodenname()  Methodenname()  Methodenname (attributname:Datentyp): Datentyp |

Diese Art von Diagrammen unterstützt Entwickler und Kunden bei der Visualisierung von Programmen, die in objektorientierten Programmiersprachen wie Python und Java geschrieben wurden. Dadurch können potenzielle Design- und Architekturfehler früher identifiziert und behoben werden. Oft werden sie sogar direkt als Vorlage für zu erstellenden Code genutzt. Um ein Klassendiagramm zu verstehen, ist es hilfreich bestimmte Definitonen zu kennen.

Eine Klasse ist die Verallgemeinerung gemeinsamer Objekte, und Objekte sind die spezifischen Ausprägungen einer Klasse. Beispielsweise können durch die Klasse Hund verschiedene Objekte instanziiert werden,

z. B. ein Schäferhund oder ein Pudel.

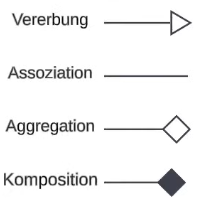
*Abb.3: Beispiel für ein Klassendiagramm*

|  |  |  |
| --- | --- | --- |
| **Zeichen** | **Zugriffs-modifizierer** | **Zugriff** |
| + | public | innerhalb und außerhalb der Klasse |
| - | private | innerhalb der Klasse |
| # | protected | innerhalb der Klasse und durch spezielle Klassen (z.B. Vererbung) |
| ~ | package | innerhalb des gleiches Pakets |

Diese Objekte haben bestimmte Eigenschaften oder Merkmale, die als Attribute bezeichnet und als Variablen in der Klasse hinterlegt werden. Die Objekte können Aktionen durchführen; diese werden Methoden genannt und sind durch einen Strich von den Attributen im Klassendiagramm getrennt. Attribute und Methoden, die für alle Objekte gleich sind, werden unterstrichen, siehe Abb.3.

Durch Zugriffsmodifizierer wird im Programmcode die Sichtbarkeit, d. h. die Datenkapselung, realisiert. Diese werden im Klassendiagramm als Zeichen den Attributen und Methoden vorangestellt.

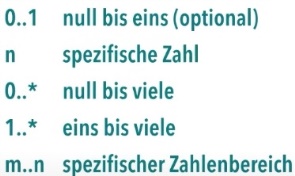
Wesentlich beim Verständnis dieser Diagrammart sind die Beziehungen zwischen den einzelnen Klassen. Je nach Beziehungstyp werden sie unterschiedlich dargestellt.

 *Abb.4: Zeichen für Zugriffsmodifizier*

Bei der Vererbung erbt die Kindklasse (die auch abgeleitete Klasse, untergeordnete Klasse oder Subklasse bezeichnet wird) die Attribute von der abstrakten Elternklasse (die auch Basisklasse oder übergeordnete Klasse genannt wird). Abstrakte Elternklassen können selbst nicht instanziiert werden.

Eine Assoziation beschreibt eine Verbindung zwischen zwei Klassen.

*Abb.5: Beziehungsarten*

Eine Aggreagtion beschreibt eine „hat eine“-Beziehung zwischen zwei Klassen, bei der ein Objekt eine andere Klasse enthält oder zu ihr gehört, wobei die Einzelteile auch ohne das Ganze bestehen können. Beispiel: Die Mitglieder einer Fußballmannschaft können auch ohne das Team existieren.

*Abb.6: Mulitplizität*

Die Komposition ist ein Spezialfall der Aggregation. Hierbei können die Elemente nicht ohne das Ganze bestehen. Beispiel: Zimmer können nicht ohne ein Haus oder eine Wohnung bestehen.

Multiplizität beschreibt die Anzahl der Instanzen einer Klasse, die über eine bestimmte Assoziation mit einer Instanz einer anderen Klasse verbunden werden können.

Quelle:

https://luna.informatik.uni-mainz.de/eis20/aufgaben/blatt07/abschnitte/uml.md

https://lecture2go.uni-hamburg.de/l2go/-/get/v/42245

https://oer-informatik.de/uml-klassendiagramm-sichtbarkeitsmodifikatoren

https://mi-learning.mi.hs-offenburg.de/SWE/UML/klassenDiagramm/fakten/index.html