

Das Klassendiagramm in der UML 2.x

1 – Darstellung von Klassen

Inhaltsverzeichnis

1 Einleitung.....	1
2 Klasse.....	1
2.1 Einfache Darstellung.....	1
2.2 Ausführliche Darstellung.....	2
2.2.1 Bezeichnung der Klasse.....	2
2.2.2 Attribute der Klasse.....	2
2.2.3 Methoden der Klasse.....	4

1 Einleitung

Ähnlich wie man Struktogramme, Programmablaufpläne oder Pseudocode zur Planung und Dokumentation von Algorithmen verwenden kann, kann man die UML nutzen, um sich vor der objektorientierten Programmierung eine exakte Planung der zu verwendenden Klassen, Attribute, Methoden und Beziehungen zu erstellen.

Dafür nutzt man ein sogenanntes **Klassendiagramm**. Die im UML-Standard von der Object Management Group (OMG) festgelegten Notationselemente und Schreibweisen werden in den folgenden Unterkapiteln ausführlich an Beispielen beschrieben. Als Szenario für die Beispiele dient eine geplante Software für eine Kinokette.

2 Klasse

Um eine Klasse in einem UML-Diagramm darzustellen, gibt es eine einfache und eine ausführliche Notation, die sich gemischt in einem Diagramm verwenden lassen.

2.1 Einfache Darstellung

In der einfachen Darstellung wird eine Klasse einfach nur als ein mit Namen versehenes Rechteck dargestellt. Für ein Kino der Kette würde dies also so aussehen:



Abbildung 1: einfache Darstellung einer Klasse

2.2 Ausführliche Darstellung

Bei einer ausführlichen Darstellung der Klasse werden neben dem Klassennamen noch die Attribute der Klasse und die wichtigsten Methoden der Klasse mit angegeben.

Dazu wird das Rechteck in drei Bereiche geteilt.

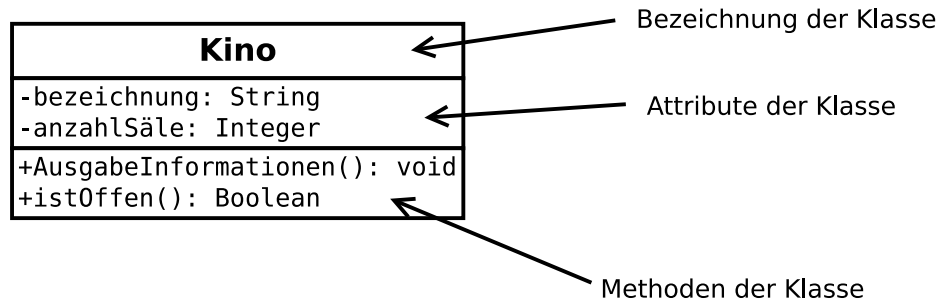


Abbildung 2

2.2.1 Bezeichnung der Klasse

In den obersten Bereich trägt man, wie auch bei der einfachen Darstellung, den Namen der Klasse ein.

2.2.2 Attribute der Klasse

In den mittleren Bereich trägt man die (wichtigsten) Attribute der Klasse ein. Für jedes Attribut wird eine eigene Zeile im Bereich verwendet. Hierfür gibt es mehrere verschiedene Notationselemente. Diese werden in der Schreibreihenfolge **von links nach rechts** beschrieben. Ein Übersichtsbeispiel gibt es am Ende dieses Unterkapitels.

- Sichtbarkeit: public (+), protected (#), private (-), package (~)
- Name des Attributs:
 - In der Regel bestehen die Namen, wegen der Einschränkungen gängiger Programmiersprachen, aus Buchstaben (ohne Sonderzeichen), Zahlen, Bindestrich und Unterstrich (Empfehlung / Konvention).
 - Attributnamen beginnen mit einem Kleinbuchstaben (Empfehlung, Konvention)
- Typ des Attributs:
 - Zuerst folgt hinter dem Namen ein Doppelpunkt.
 - Danach wird der Typ des Attributs angegeben. Jeder Datentyp oder Klassenname kann hier angegeben werden.

- **Multiplizität (optional):**
Wie oft gibt es dieses Attribut zur Laufzeit des Programms in einem Objekt der Klasse mindestens und höchstens?
An dieser Stelle sind verschiedene mögliche Angaben zu beachten:
 - Die Angabe erfolgt in der Schreibweise [untere Grenze..obere Grenze].
 - Die untere Grenze gibt die Mindestanzahl der Ausprägungen des Attributs an, die vorhanden sein muss.
 - Die obere Grenze ist optional. Sie gibt die maximale Anzahl an Ausprägungen des Attributs an. Fehlt sie, so wird eine exakte Anzahl an vorhandenen Elementen dieses Attributs angegeben.
Gibt man ein * für die Obergrenze an, so ist diese beliebig.
 - [*] oder [0..*] bedeutet eine beliebige Anzahl an Ausprägungen dieses Attributs.
- **Defaultwert des Attributs (optional):**
 - Möchte man angeben, welchen Defaultwert das Attribut bekommt, so gibt man zuerst ein = an.
 - Danach wird der Wert angegeben. Er muss dem Typ des Attributs entsprechen. Zeichenketten werden beispielsweise in " " eingeschlossen. Zeichen in ' '.
 - Gibt es das Attribut mehr als ein Mal (Multiplizität > 1, siehe oben), dann werden die Defaultwerte in { } als durch Kommata getrennte Liste angegeben.
- **Eigenschaften (optional):**
 - Eigenschaften werden in { } als letzte Angabe in einer Attributszeile angegeben.
 - Die UML spezifiziert unterschiedliche Eigenschaften. Für uns ist eigentlich nur {readOnly} interessant. Es gibt ein konstantes Attribut an. (Java: Schlüsselwort final).
- **Klassenattribute (Java: Schlüsselwort static)** werden unterstrichen dargestellt. Dabei wird die gesamte Zeile für das Attribut unterstrichen.

Beispiel für die Angaben bei obiger Kino-Klasse:

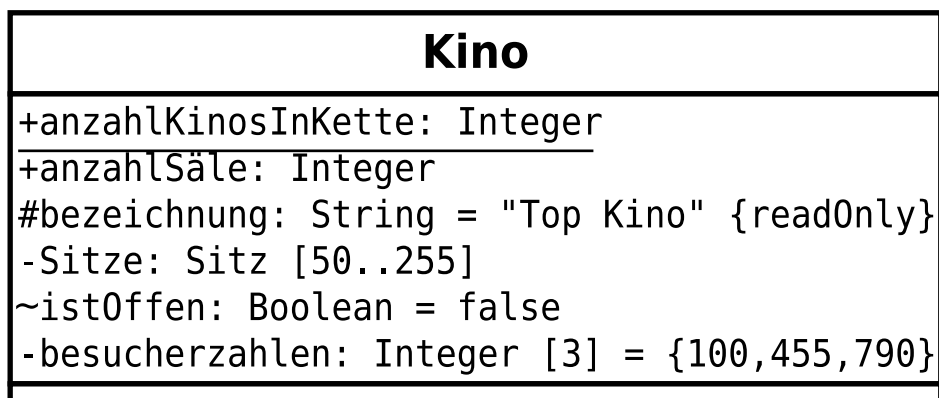


Abbildung 3: Beispiel zu obigen Notationselementen

2.2.3 Methoden der Klasse

In den untersten Bereich trägt man die Methoden der Klasse ein, die man im Diagramm angeben möchte. Hier beschränkt man sich meist auf die wichtigsten Methoden. Die get-/set-Methoden werden hier zum Beispiel in der Regel nicht angegeben.

Auch hier sind verschiedene Notationselemente zu berücksichtigen. Die Angaben erfolgen wieder in der Schreibreihenfolge **von links nach rechts**.

- Sichtbarkeit: public (+), protected (#), private (-), package (~)
- Name der Methode. Empfehlenswert sind dieselben Einschränkungen wie bei Attributbezeichnungen.
- Öffnende runde Klammer. Die Liste wird durch Kommata unterteilt. Jeder Listeneintrag folgt der unter *Parameter* angegebenen Schreibweise
- Parameterliste und Übergabeparameter (optional):
Hinter der runden Klammer kann eine Liste von Übergabeparametern angegeben werden. Verschiedene Parameter werden durch Kommata getrennt. Für jeden Parameter der Liste gilt die folgende Schreibweise:
 - Wie bei Attributen wird hier zuerst der Name und dann nach einem Doppelpunkt der Typ des Parameters angegeben.
 - Es kann danach eine Multiplizität angegeben werden. Die Schreibweise entspricht der unter *Multiplizität* bei Attributen angegebenen Schreibweise.
 - Default-Wert: Auch hier kann, wie bei den Attributen, mit der bei Attributen angegebenen Schreibweise ein Default-Wert angegeben werden.
- Schließende runde Klammer. Diese Klammer beendet die Parameterliste.
- Rückgabetyt (optional): Dieser kann (außer bei Konstruktoren) angegeben werden.
 - Es folgt zunächst ein Doppelpunkt.
 - Danach erfolgt die Angabe eines Rückgabetyps. Hier können alle Datentypen und Klassen angegeben werden.
 - Fehlt der Rückgabetyt, so wird automatisch void angenommen (außer bei Konstruktoren!)

Beispiel für die Angaben bei obiger Kino-Klasse:

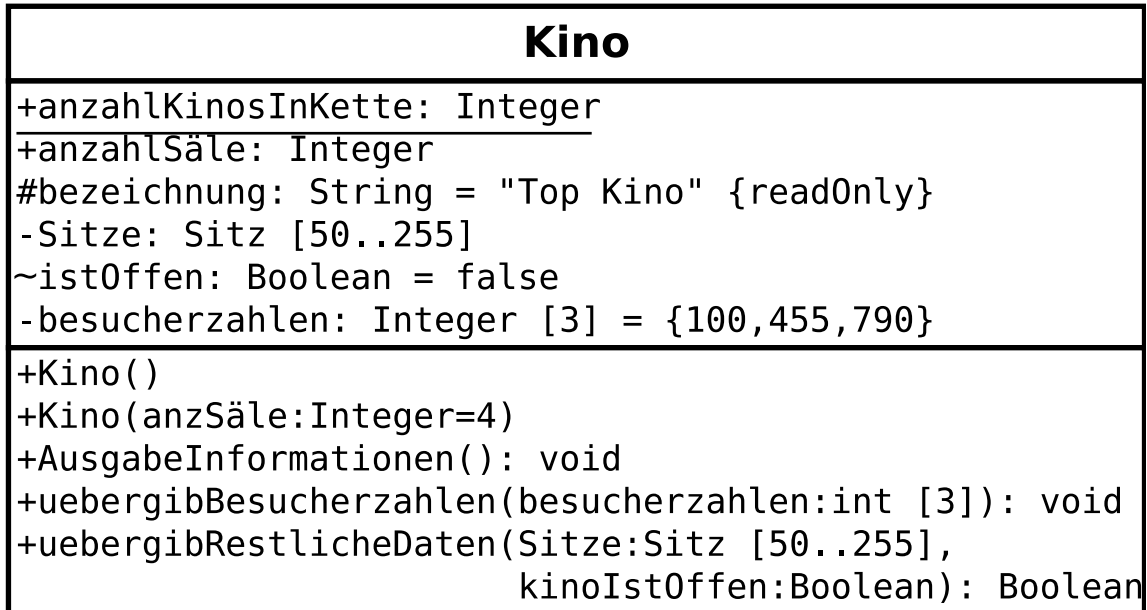


Abbildung 4: Beispiel für eine Klasse in UML mit Attributen und Methoden