

Klasse
+publicAttribut -privateAttribut <u>+publicKlassenAttribut</u> <u>-privateKlassenAttribut</u>
+publicKonstruktor() -privateKonstruktor +publicMethode() -privateMethode() <u>+publicKlassenMethode()</u> <u>-privateKlassenMethode()</u>

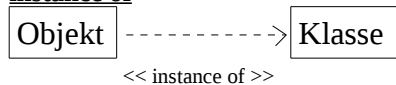
BibMitglied
<u>-mitgliederZaehler: int</u> -name: String -privAnschrift: String -mitgliedsNummer: int
+Mitglied(n: String, pA: String) <u>+getMitgliederZaehler()</u> <u>+setMitgliederZaehler(mz: int)</u> +getName(): String +setName(n: String) +getPrivAnschrift(): String +setPrivAnschrift(pA: String) +getMitgliedsNummer(): int +setMitgliedsNummer(mnr: int) +print()

Rückgabebetyp void bei (Klassen-) Methoden wird weggelassen

Um Verwechslungen in der Darstellung zwischen Objekten und Klassen zu vermeiden, werden

Objektbezeichner unterstrichen

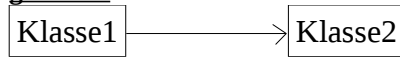
instance of



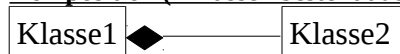
ungerichtet



gerichtet



Komposition ("Klasse1 besteht aus Klasse2, kann ohne sie nicht existieren")

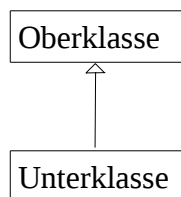


Name & Rollennamen



> gibt die Leserichtung an ("Person leitet Projekt")

Vererbung



Alle nicht-private Attribute und Methoden werden an die Klasse Unterklasse vererbt. Möchte man ein Attribut lediglich in Ober- und Unterklasse, aber nicht außerhalb haben verwendet man protected (#)

Textanalyse

1. Substantive im Text

- Kandidaten für Klassen, wenn damit etwas Systemrelevantes beschrieben wird, über das mehrere Informationen zu speichern sind, z. B. Mitglied, Buch
- Kandidaten für Attribute, wenn damit nur einzelner Wert beschrieben wird, z. B. Name, Mitgliedsnummer

2. Verben im Text

- Kandidaten für Methoden, wenn damit etwas Systemrelevantes beschrieben wird