

Unified Modeling Language

- ?? anerkannte Modellierungssprache für OO Softwareentwicklung
- ?? unterstützt Konzepte des Unified Development Process:
Use Cases, statische Modellierung, dynamische Modellierung

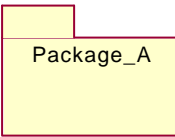
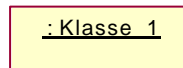
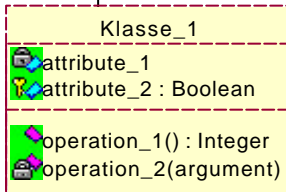


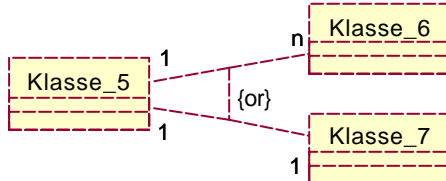
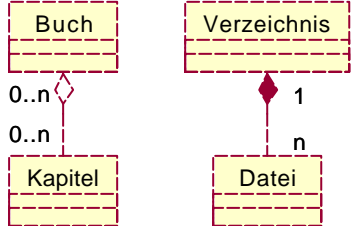
Klassenmodell (statisches Modell)

Generell gibt es 2 Phasen der Modellierung:

OOA (Analyse): Problem Domain => Klassenmodell der Anwenderwelt

OOD (Design): Solution Domain => Klassenmodell der Implementierung

Konzepte und Notation:

Symbol	Erläuterung
	<u>Packet:</u> Kapselung mehrerer zusammengehöriger Klassen, s. Packet in Java
	<u>Objekt:</u> eine Instanz der Klasse „Klasse 1“
	<u>Klasse:</u> Name: Klasse_1 attribute_1: private attribute_2: protected, return type: Boolean operation_1: public, return type: Integer operation_2: private, Parameter: argument
	<u>Interface:</u> Klasse_2 <u>Abstrakte Klasse:</u> Klasse_3
	<u>Vererbung:</u> Klasse_4 erbt von Klasse_3
	<u>Assoziationen:</u> Kardinalitäten für jede Assoziation Constraint: entweder Klasse_5 – Klasse_6 oder Klasse_5 – Klasse_7
	<u>Aggregation:</u> ein Buch kann keines oder mehrere Kapitel enthalten. Buch und Kapitel bleiben unabhängig voneinander bestehen. <u>Komposition:</u> jede Datei muss in einem Verzeichnis stehen. Wird das Verzeichnis gelöscht, werden auch alle enthaltenen Dateien gelöscht.

