



University
of Basel

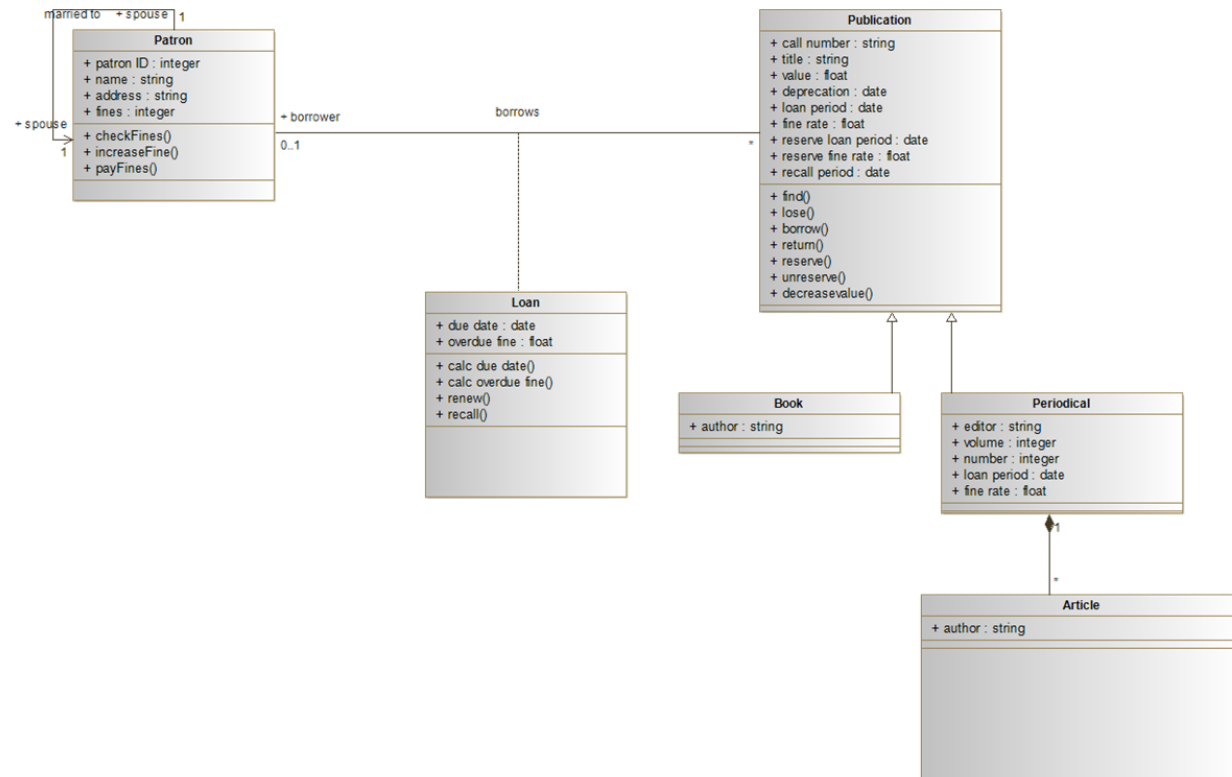
Software Engineering

Marcel Lüthi, Universität Basel

UML Klassendiagramme

Grafische Modellierungssprachen

Erlauben Software auf hoher Abstraktionsebene zu verstehen



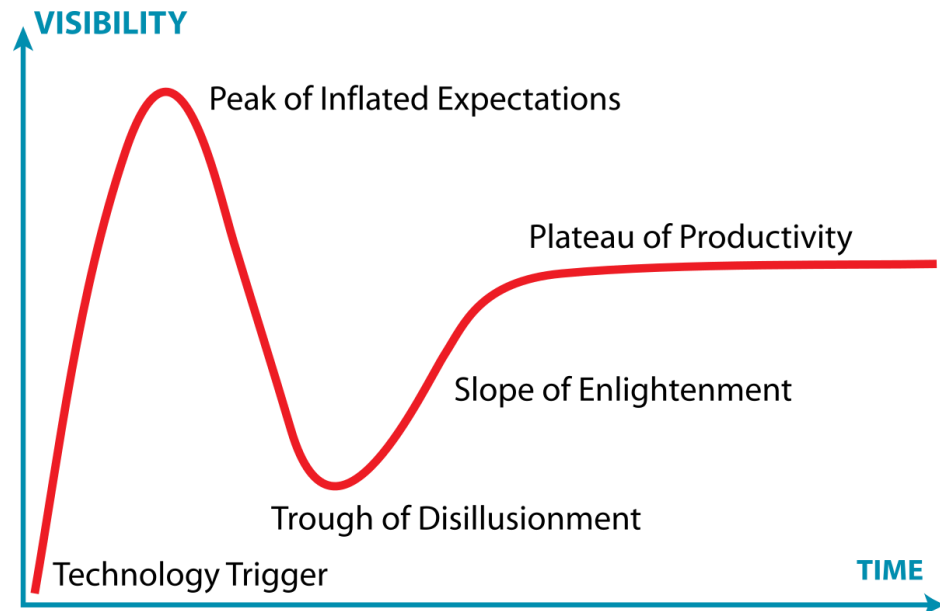
Hilft beim

- Modellieren
- Dokumentieren
- Kommunizieren

Unified Modelling Language

Grafische Modellierungssprache zur Spezifikation, Konstruktion und Dokumentation von Software und anderen Systemen

- Quasi standard zur graphischen Modellierung von OO-Systemen
- Entwickelt 1990 (G. Booch, I. Jacobson, J. Rumbaugh)
- Mittlerweile in Version 2.5



Diagrammarten der UML

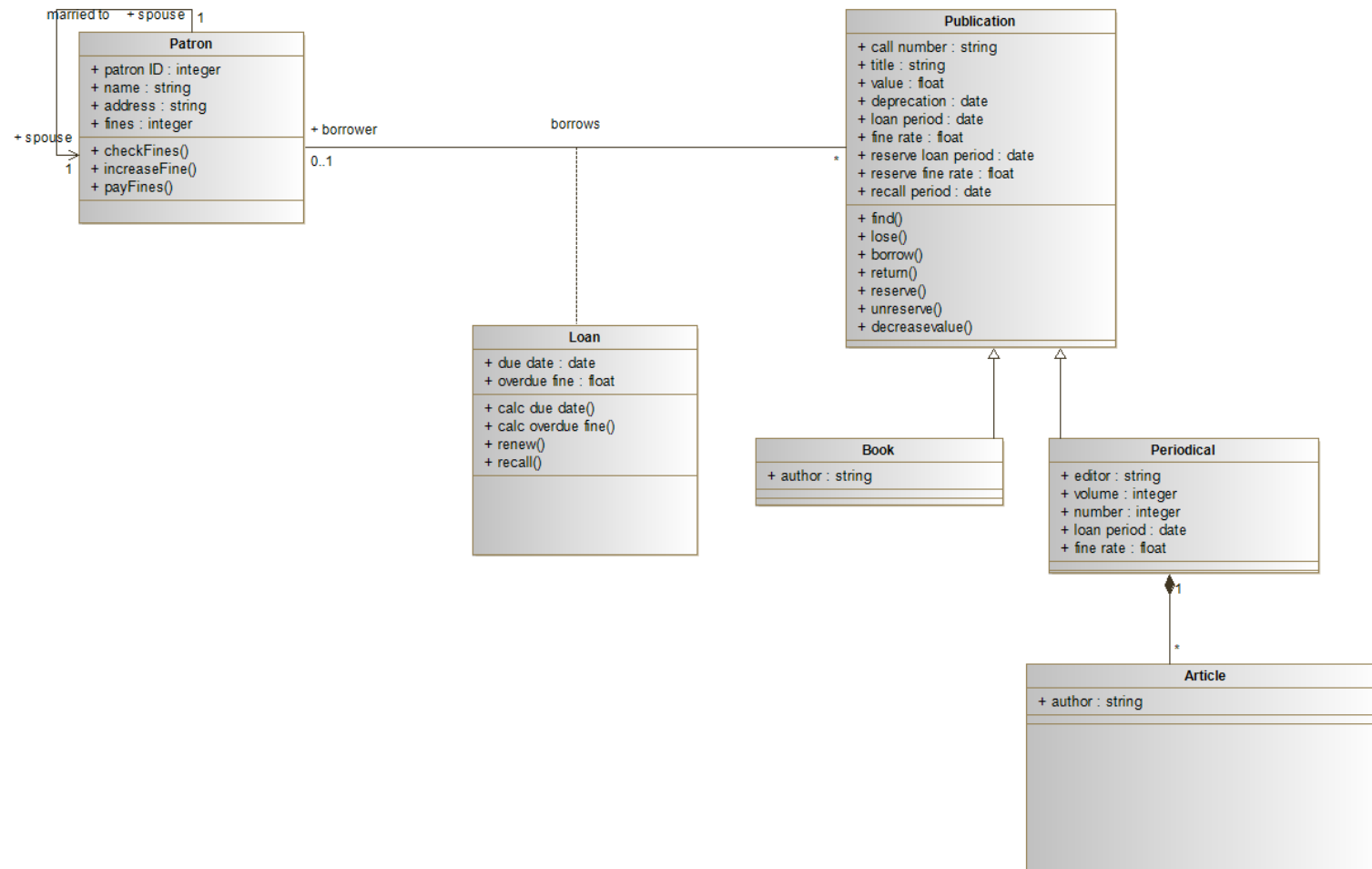
Strukturdiagramme (statisch)

- Klassendiagramm
- Montagediagramm
- Komponentendiagramm
- Verteilungsdiagramm
- Objektdiagramm
- Paketdiagramm
- Profildiagramm

Verhaltensdiagramme (dynamisch)

- Aktivitätsdiagramm
 - Use-case Diagramm
 - Interaktionsdiagramm
 - Kommunikationsdiagramm
 - Sequenzdiagramm
 - Zeitverlaufdiagramm
 - Zustandsdiagramm
-

Klassendiagramm



Klassen

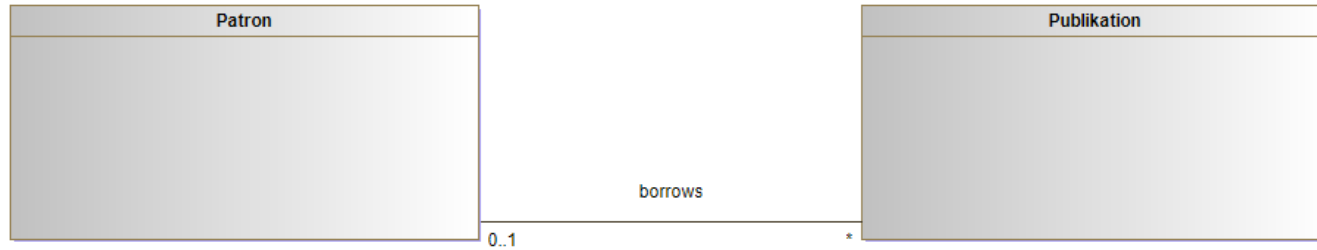
Publication
+ call number : string + title : string + value : float + deprecation : date + loan period : date + fine rate : float + reserve loan period : date + reserve fine rate : float + recall period : date
+ find() + lose() + borrow() + return() + reserve() + unreserve() + decreasevalue()

Name

Attribute

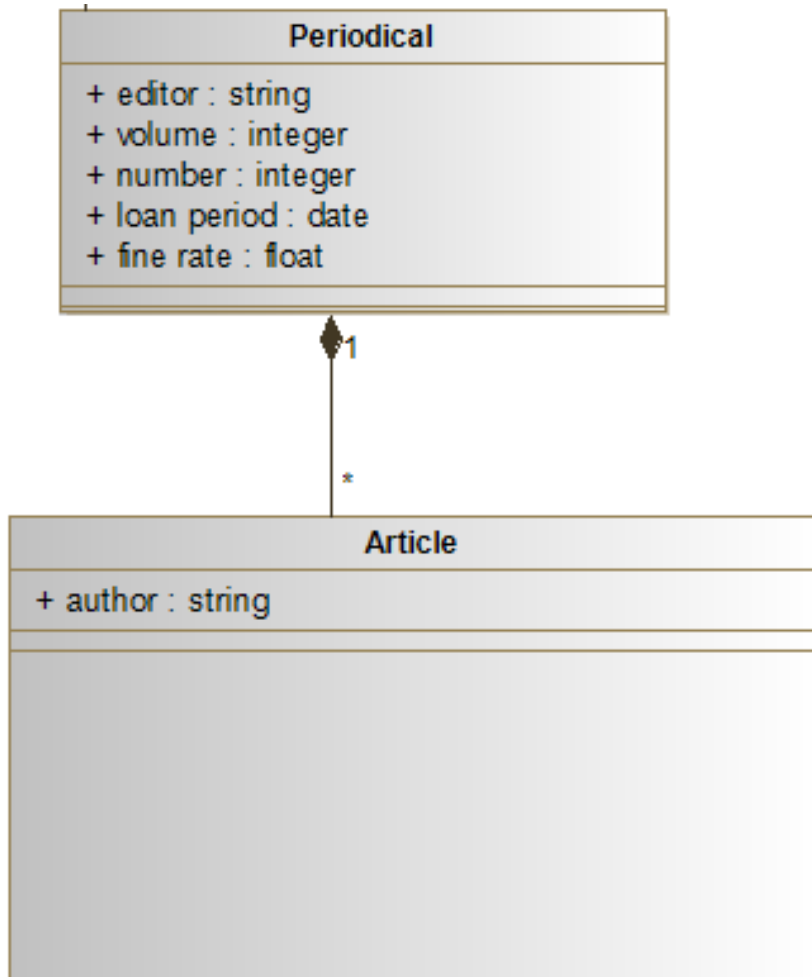
Methoden

Assoziationen



- Beziehungen zwischen Klassen, die Implementation unterstützen muss
 - Optional: Multiplizitäten angeben
 - Definiert implizit eine "Uses" Beziehung
-

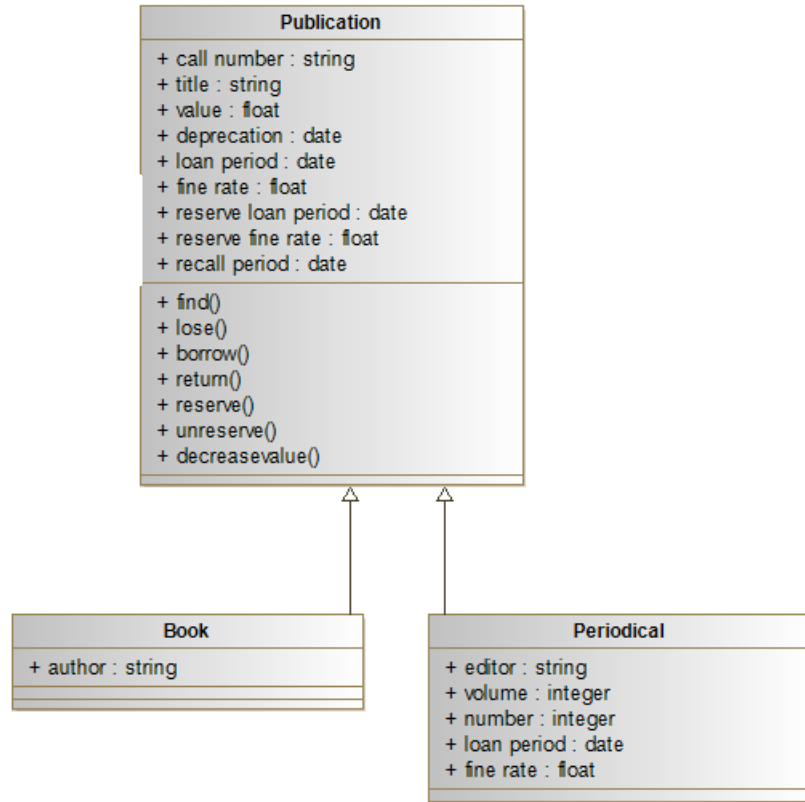
Aggregation



Teil-Ganze Beziehung

- Impliziert “is_component_of” Beziehung

Vererbung



Modelliert Varianten:

- Publikation ist entweder ein Buch oder ein “Periodical”

Assoziationsklassen

