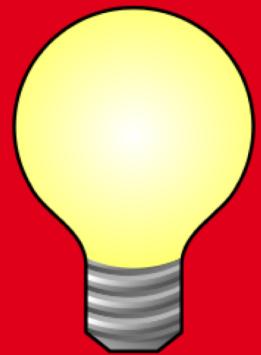




Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

12.11.2020
UML allgemein

Einführung in die UML





Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

AGENDA

Einführung ins Thema

Was ist UML?

Wie geht's jetzt mit der UML weiter?

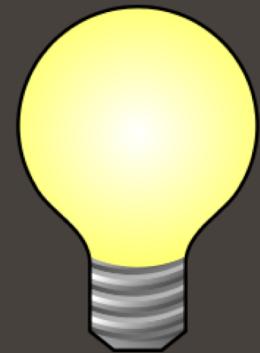


Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

01

EINFÜHRUNG INS THEMA

Ziel:
Die Eckpunkte des Themas kennenlernen



MODELLIERUNGSSPRACHEN



Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

- Bei der Softwareentwicklung haben wir es häufig mit
 - einem großen Anwendungssystem
 - und vielen KollegInnen zu tun
 - Wichtige Fragen bei der praktischen Durchführung:
 - Wie können wir uns auch über verzwickte Sachverhalte austauschen?
 - Wie können uns möglichst viele verstehen?
 - . . .
- Grafische Modellierungssprachen, z.B. UML
- **Verständliche** Darstellung auch **komplexer/komplizierter** Sachverhalte durch **Grafiken (Zeichnungen)**
 - **standardisiert**



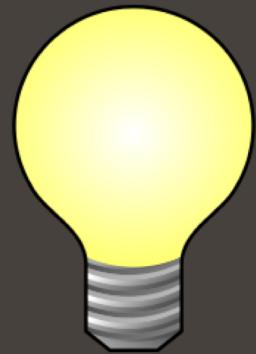
Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

02

Was ist UML?

Ziel:

Die UML grundsätzlich kennenlernen

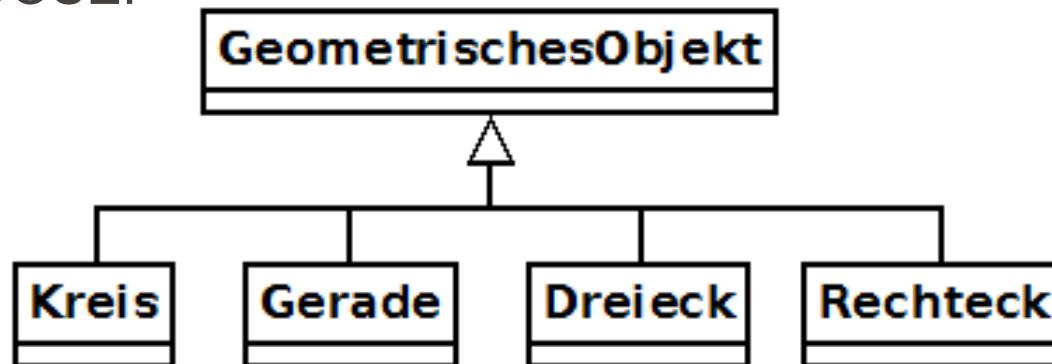




WAS IST UML?

= Unified Modeling Language

- Standard der OMG
 - OMG = Object Management Group - <http://www.omg.org/>
 - Aktuelle Version: UML 2.5
 - > 10 Diagrammarten
- Ziel: Graphischen Modellierung objektorientierter Softwaresysteme
→ Mittlerweile de-facto Standard für Modellierung von Software aller Art
- BSP aus OOSE:



WAS IST UML?

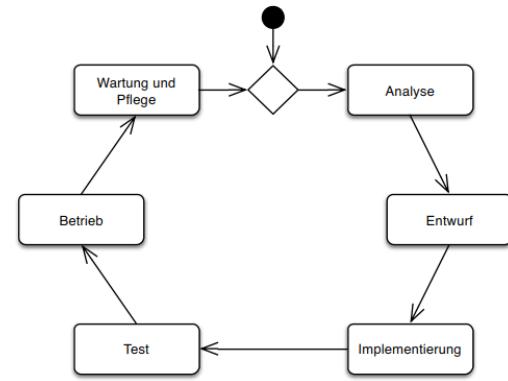


Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

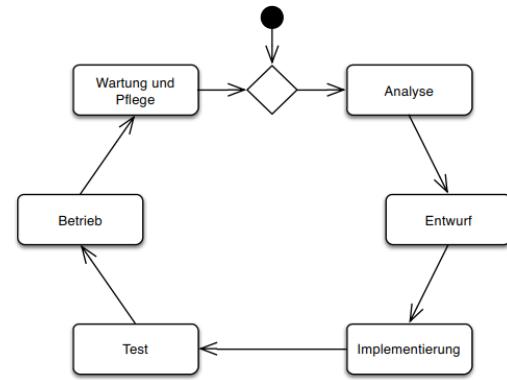
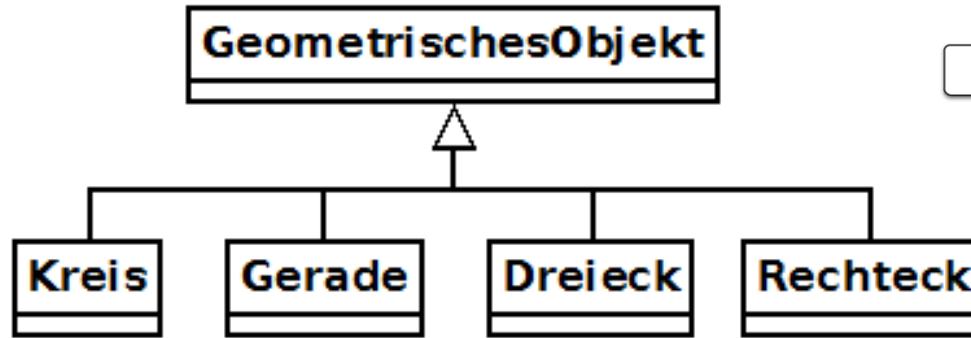
- Für uns:
 - Grundlegende Modellierungssprache
 - Später lernen wir jedoch auch noch FMC kennen
 - Hat gewisse Vorzüge in der Architekturmodellierung
 - Aber mit UML kann man das eigentlich auch machen

UML-DIAGRAMME SIND VIELSEITIG VERWENDBAR

- Typische Verwendungsarten:
 - Anforderungsspezifikation
 - Analyse
 - Entwurf
 - Implementierung (Detailed Design, Nachträgliche Dokumentation)
 - Test (Design eines Testszenarios)
- Das kann verwirrend sein:
 - Diagramm-Syntax ist unabhängig von der Verwendungsart
 - Syntax (= Regelsystem für Form/Aussehen)
 - Diagramm-Semantik ist abhängig von der Verwendungsart
 - Semantik (= Bedeutung)



BEISPIEL: ARTEN VON KLASSENDIAGRAMMEN



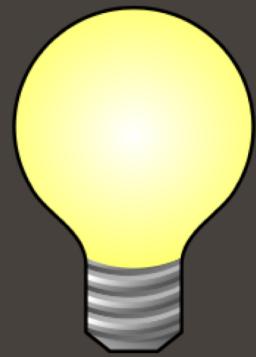
- Implementierungs-Klassendiagramm
 - (fast) gleichwertig zu Quellcode
 - Entwurfs-Klassendiagramm
 - 1:1-Beziehung zum Quellcode (hoffentlich!)
 - Aber: programmiersprachenunabh., oder unabh. von Details
 - Analyse-Klassendiagramm
 - Beschreibt fachliche und **keine** technischen Konzepte
- Gleiche Syntax für alle Arten von Klassendiagrammen
- Semantik (Bedeutung) abhängig von der Verwendungsart



Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

03 Wie geht's weiter

Ziel:
Wie geht es jetzt weiter?



UML-BÜCHER



Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

- H. Störrle: UML 2 für Studenten.
→ Als Einführung geeignet – Leider teilw. nicht auf akt. Stand
- H.J. van Randen; et al.: Einführung in UML
Gratis in Bib. verfügbar: <https://hds.hebis.de/hsrm/Record/HEB386572232>
- Ch. Rupp et al: UML 2 glasklar [BF 500 91]
→ Für Fortgeschrittene geeignet – hilft auch bei kniffligen Fragen
- Spezifikationen im Original auf www.uml.org
→ Man muss wissen, dass es das gibt und wo man das findet.
– Jedoch eher Nicht zum Lernen geeignet

Mein Tip

→ **Besorgen Sie sich ein Buch!**

WAS MACHEN WIR JETZT IN DEN NÄCHSTEN WOCHEN?



Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

- Heute: Objektdiagramm
 - Objekte und deren Beziehungen (Links)
- Nächste Woche: Klassendiagramm
 - Klassen und deren Beziehungen (Assoziationen)
 - Vererbung, ...
- Danach:
 - Verhaltensdiagramme
 - Zustandsdiagramme
 - Aktivitätsdiagramme
 - Interaktionsdiagramme
 - Sequenz- und Kommunikationsdiagramme
- Später:
 - UseCase-Diagramm



Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

AUF GEHT'S!!

SELBER MACHEN UND LERNEN!!



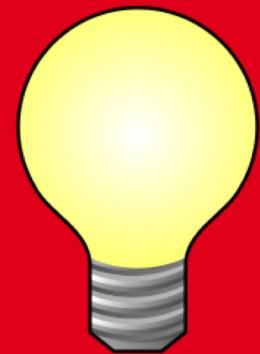


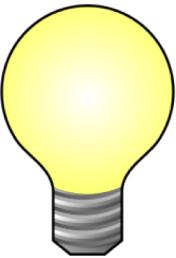
Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

12.11.2020

Das Objektdiagramm

Objektdiagramme nutzen





AGENDA

Einführung ins Thema

Darstellung von Objekten

Verbindungen (Links)

Spezialfall: Konstanten

Weitere Hinweise

Fazit

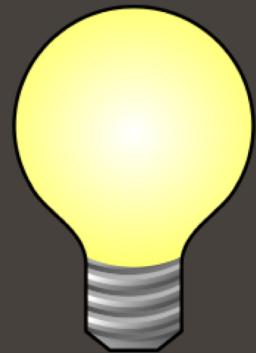


Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

01

EINFÜHRUNG INS THEMA

Ziel:
Die Eckpunkte des Themas kennenlernen





WAS SIND OBJEKTE?

1. Laut Duden: „Objekt“ ist aus dem Lat. \triangleq „Gegenstand“
 - Objekt \triangleq Ding mit Eigenschaften und Fähigkeiten

2. Beim Programmieren (z.B. Java):

Klasse:

\leftrightarrow Objekt:

```
public class Person {  
  
    String nachname;  
    int groesse;  
  
    public Person(String nname  
                  , int groess){  
        nachname= nname;  
        groesse=groess;  
    }  
}
```

- Definitionscode unveränderlich
- Im statischen Speicher

```
Person t= new Person("Turban", 185);
```

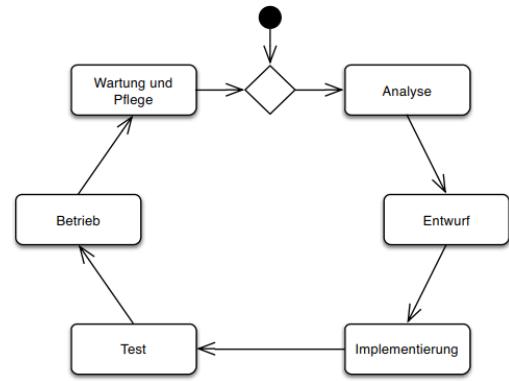
→ Konkrete Instanzen einer Klasse

→ Objektwerte, Anzahl Objekte, ...
sind veränderlich

→ Im dynamischen Speicher

„Turban“
185

WAS SIND OBJEKTE IN DER UML?



1. Laut Duden: „Objekt“ ist aus dem Lat. \triangleq „Gegenstand“
 - Objekt \triangleq Ding mit Eigenschaften und Fähigkeiten

→ Es gibt eine Person mit den Eigenschaften

Nachname == “Turban” und

Grösse == 185 cm

} Fachliches Konzept
(\triangleq Business Object)
z.B. in **Anforderungsbeschreibung** oder in der
Analyse

→ Unabhängig von Programmierung oder einer sonstigen
technischen Umsetzung

→ **Fachliche Sicht**

WAS SIND OBJEKTE IN DER UML?

2. Beim Programmieren (z.B. Java):

Klasse: ↔ Objekt:

→ Ein Objekt der Klasse

```
public class Person {  
    String nachname;  
    int groesse;  
    ...  
}
```

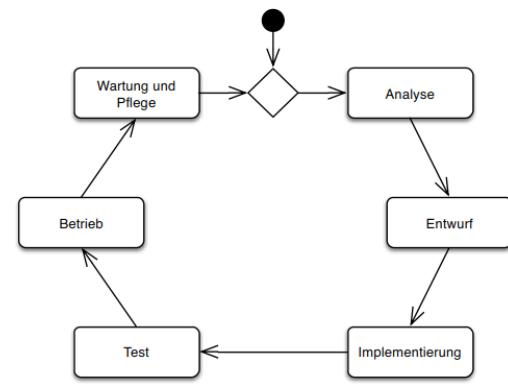
mit Attributbelegungen

„Turban“
185

Konkretes
Technisches Konzept
z.B. in **Design**
oder
Implementierung

→ Abhängig von Programmierung oder einer sonstigen
technischen Umsetzung

→ **Technische Sicht (Implementierung)**

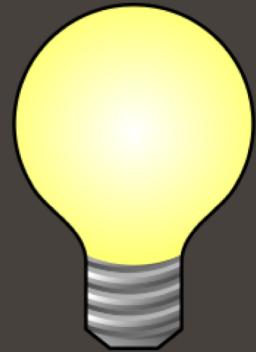




02

Darstellung von Objekten

Ziel:
Eine (mögliche) Darstellung für **beide Sichten**



(MÖGLICHE) DARSTELLUNG FÜR BEIDE SICHTEN



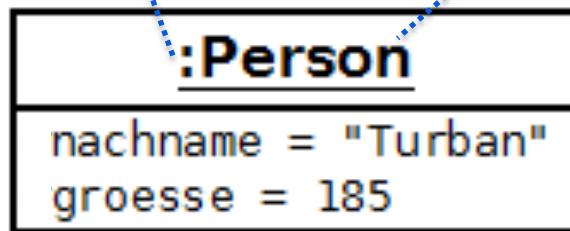
Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

Immer mit : und unterstrichen!

→ Kennzeichnet Objekt
im Gegensatz zu Klasse

Art des Objekts
(Classifier, Class Type)

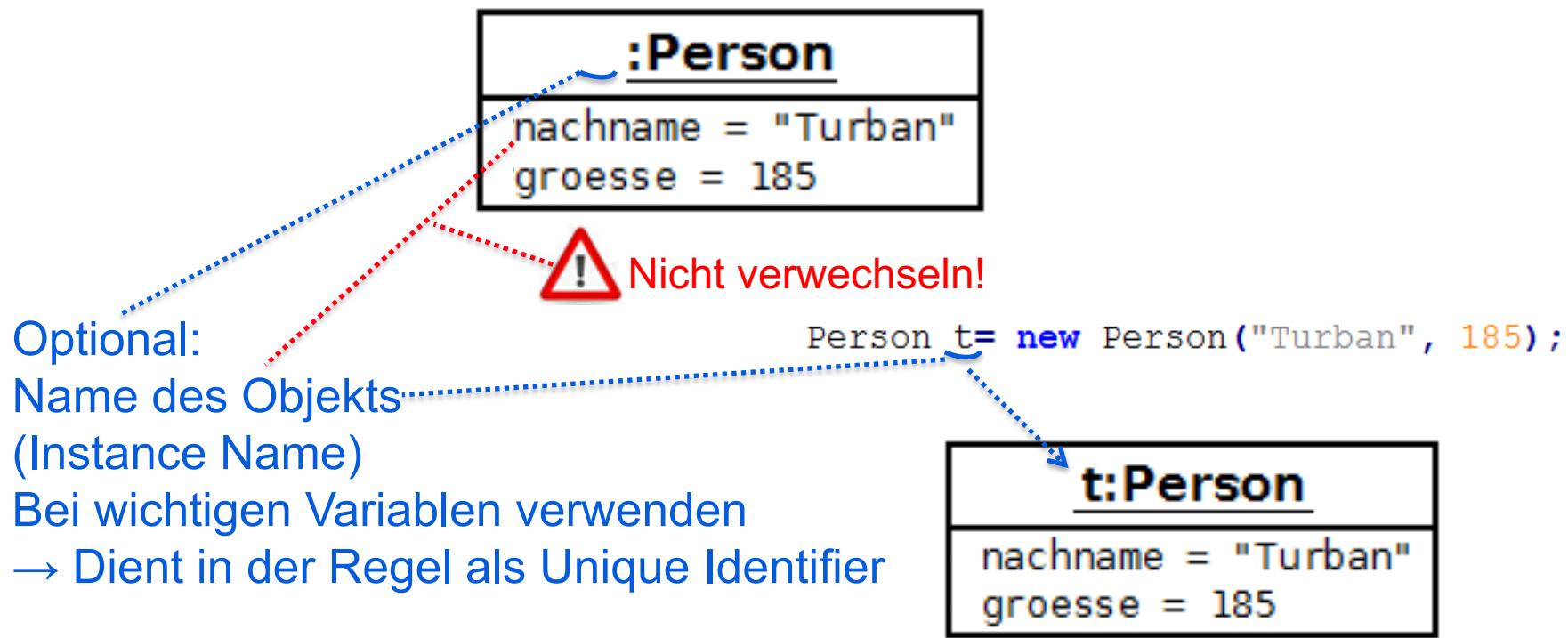
Objekt (Instanz)



(MÖGLICHE) DARSTELLUNG FÜR BEIDE SICHTEN



Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim



BEM:

Bis auf **:** ist alles optional

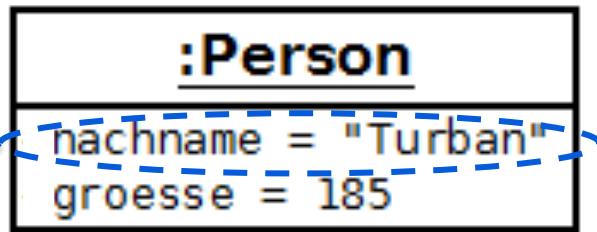


Minimum, aber macht das Sinn?

(MÖGLICHE) DARSTELLUNG FÜR BEIDE SICHTEN



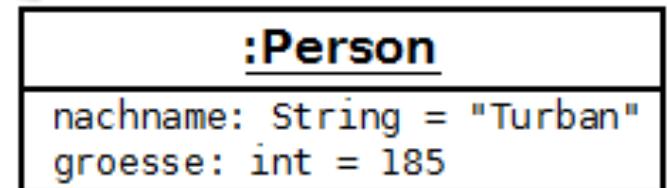
Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim



Konkrete Eigenschaft (Slot)

→ Besteht aus:

- i) Name der Eigenschaft (Feature Name)
- ii) Optional: Typ der Eigenschaft (z.B. :String)
- iii) Belegung / Wert (Value Specification)

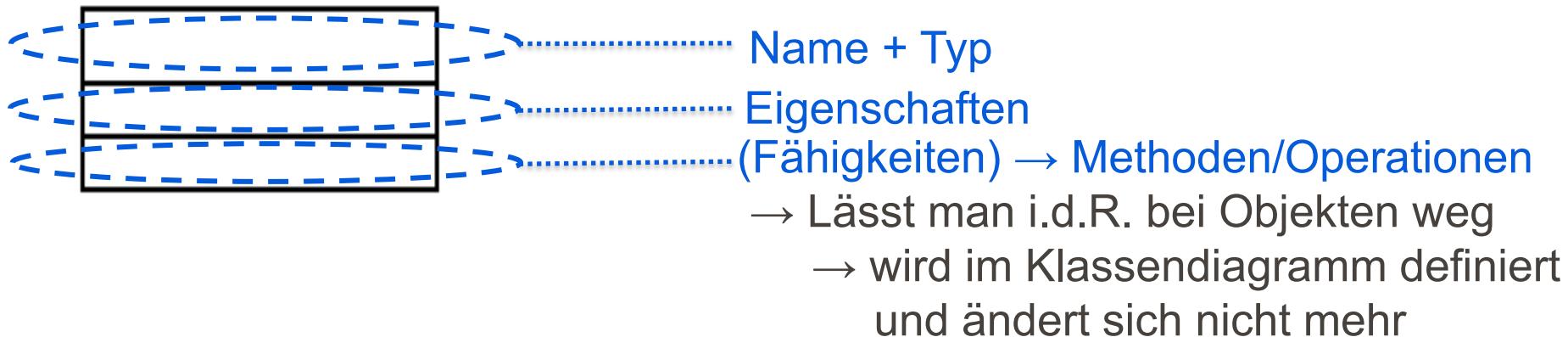


WEITERE BEMERKUNG ZUR DARSTELLUNG



Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

- Ein Objekt kann maximal drei Abschnitte haben:
 - (vgl. auch Klassendiagramm kommende Woche)



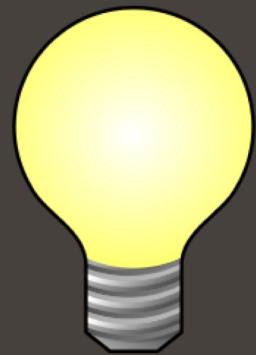


Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

03

Verbindungen (Links)

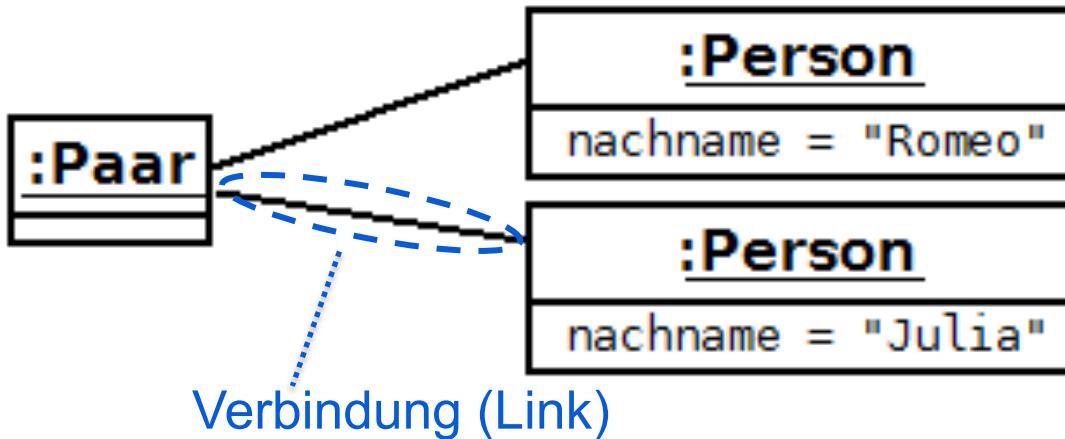
Ziel:
Verbindungen zwischen Objekten darstellen





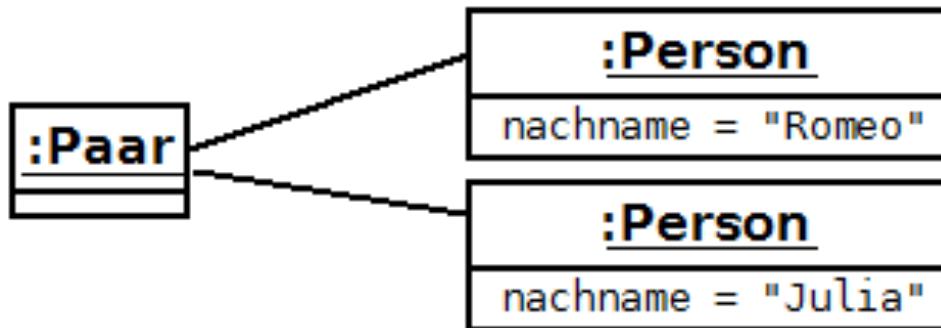
VERBINDUNGEN (LINKS)

- Objekte haben miteinander zu tun
 - Es gibt Verbindungen, die wir modellieren können müssen





VERBINDUNGEN (LINKS)



Mögliche Bedeutungen:

1. Fachliche Sicht:

- Wir haben ein Paar, das aus der Person mit Namen „Romeo“
- und der Person mit Namen „Julia“ besteht

2. Technische Sicht:

```
public class Person { ... }  
  
public class Pair {  
    Person pers1;  
    Person pers2;  
}
```

Oder:

```
public class Pair {  
    Person [] pers= Person[2];  
}
```

Oder:

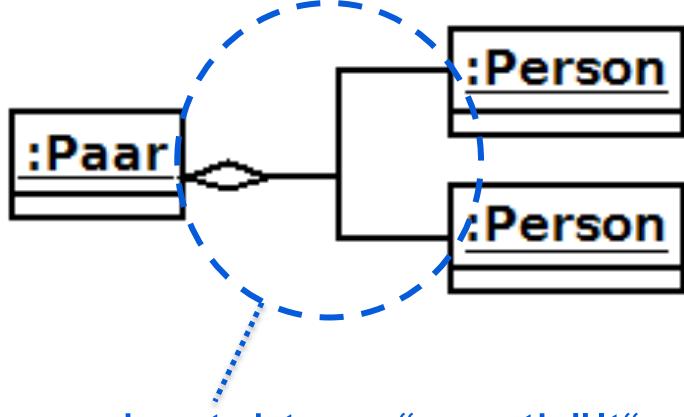
...

VERBINDUNGEN (LINKS) – AGGREGATION & KOMPOSITION

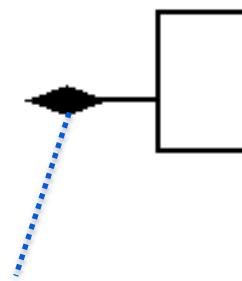


Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

- Betonung des Aspekts „besteht aus“:
(v.a. in Analysediagrammen* nützlich)



„besteht aus“, „enthält“
(Aggregation)



„besteht aus“, „enthält“
(Komposition)
Stärker als

* Diagramme, die im Zuge der Analysephase gemacht werden
→ Siehe Fachmodell in Vorl. 07 Anforderungsanalyse

VERBINDUNGEN (LINKS) – AGGREGATION & KOMPOSITION



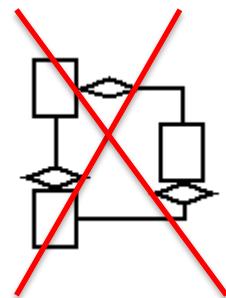
Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim



Vorsicht: UML definiert keine präzise Bedeutung für



- Lediglich:
 - Für beide sind keine zyklischen Abhängigkeiten erlaubt
 - ist irgendwie stärker als
 - Inoffizielle Unterscheidung:
 \triangleq Gesamtteil kann nicht ohne Einzelteile existieren



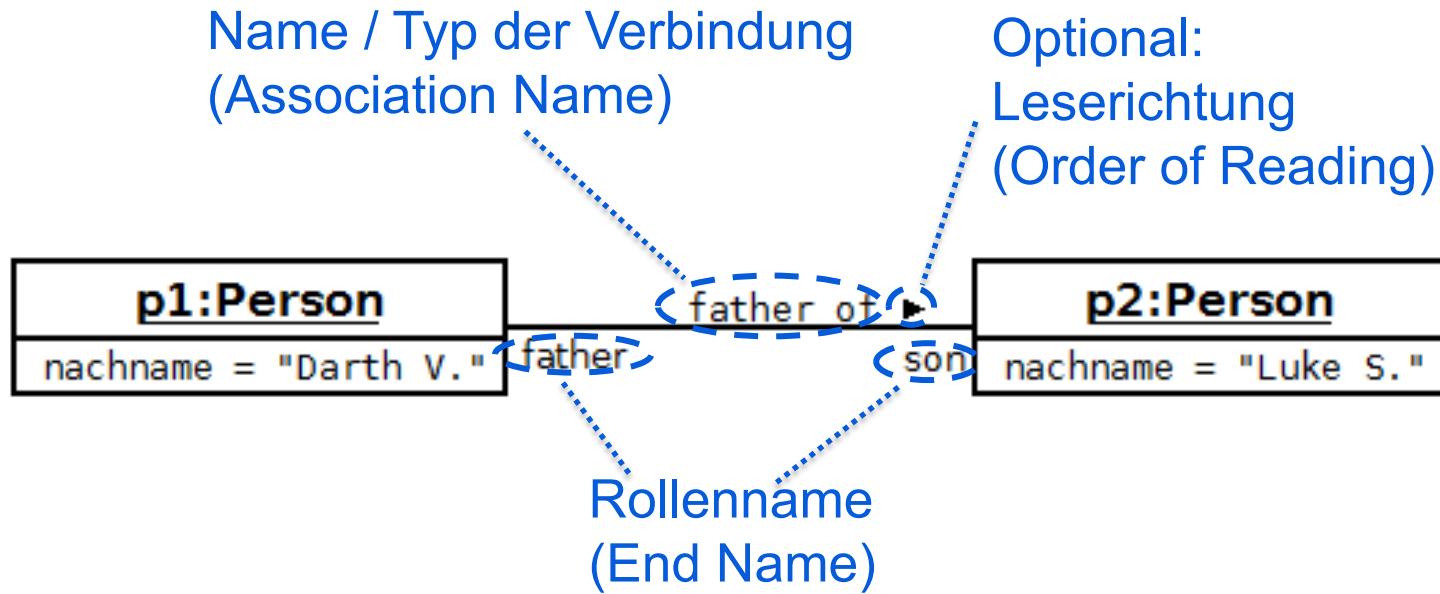
→ TIPP: Nur verwenden, wenn WIRKLICH notwendig

WEITERE INFORMATIONEN ZU OBJEKTVERBINDUNGEN (LINKS)

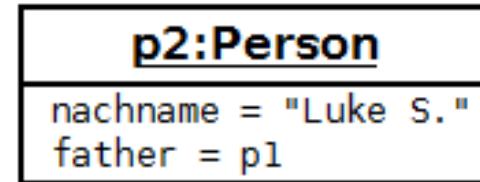
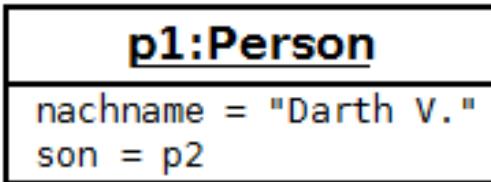


Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

- Zu Objektverbindungen (Links) können weitere Infos hinterlegt werden:



Auch möglich, aber nicht besonders schön:



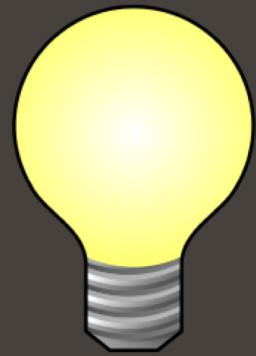


Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

04

Spezialfall: Konstanten

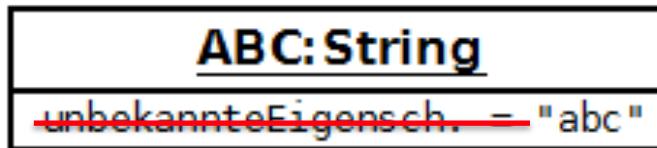
Ziel:
Konstanten darstellen



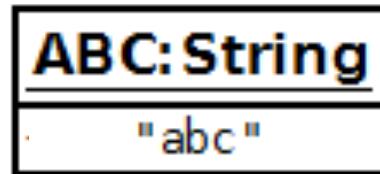
SPEZIALFALL: KONSTANTEN UND PRIMITIVE OBJEKTE



- Konstanten (Werte), oder primitive Objekte, oder berechenbare Werte darstellen:
 - Objekt mit einer Eigenschaft, deren Name nicht bekannt ist



- Da es nur einen Wert gibt, kann man dann auch den Eigenschaftsnamen weglassen:





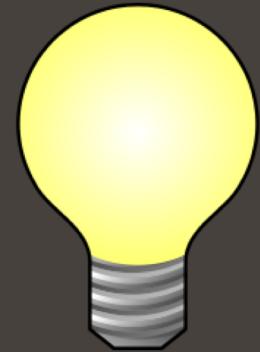
05

Weitere Hinweise

Ziel:

Die Bedeutung des Objektdiagramms

Wie hängen fachliche und technische Sicht zusammen?



DIE BEDEUTUNG DES OBJEKTDIAGRAMMS



Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

- Das Objektdiagramm stellt nur einen bestimmten Schnapschuss (Momentaufnahme) dar
 - Wenig später ist vielleicht folgendes passiert:
 - Neue Objekte erzeugt
 - Neue Verbindungen zwischen Objekt entstanden
 - Es wurden einige Objekte gelöscht
 - Einige Verbindungen zwischen den Objekten wurden gelöst
 - Die neue Situation ist ganz anders
 - Man müsste ein neues Diagramm zeichnen
- Mit dem Klassendiagramm kann ein allgemeingültiges Diagramm aller möglichen Objektkonstellationen zeichnen
- Definiert alle möglichen Objektdiagramme

WIR ERINNERN UNS AN DEN ANFANG:



Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

Programmieren:

Klasse:

```
public class Person { ... }
```

- Definitionscode unveränderlich
- Im **statischen** Speicher



Objekt:

```
Person t = new Person("Turban", 185);
```

- Konkrete Instanzen einer Klasse
- Objektwerte, Anzahl Objekte, ... sind veränderlich
 - Im **dynamischen** Speicher

„Turban“ 185

UML:

- Klassendiagramm
- Eine **statische** Sicht
- UML ermöglicht viele verschiedene Sichten (sog. Sichtenkonzept)

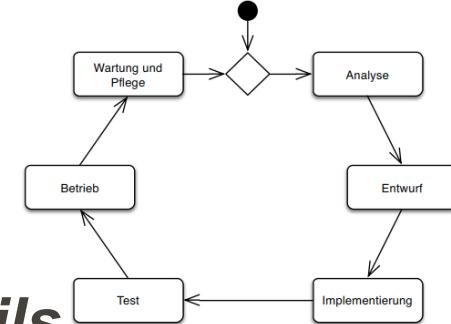
- Objektdiagramm
- Eine **dynamische** Sicht



WEITERE SICHTEN:

1. Laut Duden: „Objekt“ ist aus dem Lat. \triangleq „Gegenstand“
 - Objekt \triangleq Ding mit Eigenschaften und Fähigkeiten
 - ***Fachliche Sicht***
2. Beim Programmieren (z.B. Java)
 - ***Technische Sicht***

FACHLICHE ↔ TECHNISCHE SICHT



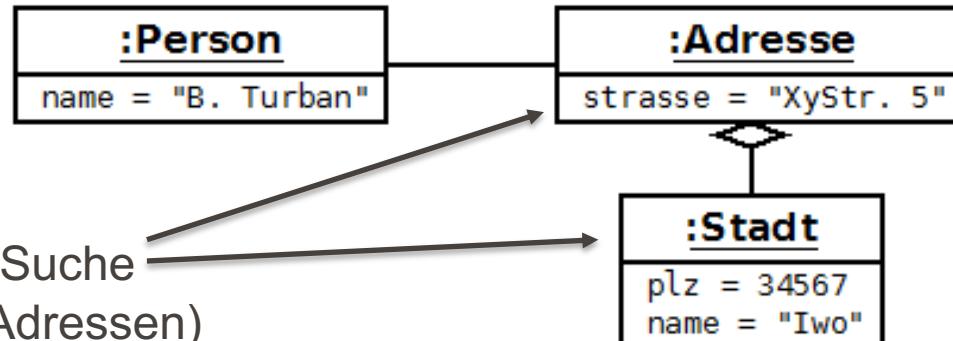
→ ***Fachliche Sicht: alles ohne technische Details***

→ Wird in der Analyse verwendet, um **NUR** die fachlichen Anforderungen zu ermitteln

:Person	
nachname	= "Turban"
groesse	= 185
adresse	= "XyStr. 5, 34567 Iwo"

→ ***Technische Sichten***

- In Architektur, Detailed Design, Implementierungsdoku. verw.
- Es werden auch die für die Lösung **relevanten** technischen Details dargestellt:



- + Technische Lösung
(z.B. für spezifische Suche nach Städten oder Adressen)

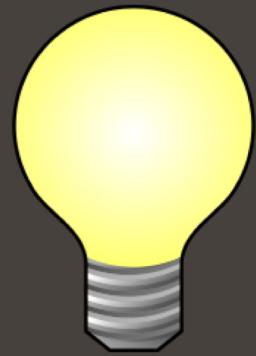


Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

06

Fazit

Ziel:
Was haben wir damit gewonnen?



WAS HABEN WIR GELERNT?



Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

- Verschiedene Sichten
 - Fachliche Sicht (Nur fachliche Zusammenhänge aus Sicht des Kunden)
→ Anforderungsanalyse
 - Technische Sicht (Technische Umsetzung)
→ Design & Implementierung
 - Beide können unterschiedlich zueinander sein
- Objektdiagramm
 - Objekte
 - Rechteckiger Kasten mit max. 3 Unterteilungen
 - Instanzname : Typname, Wertbelegungen
 - Verbindungen
 - Einfache Links, Aggregation, Komposition
 - Verbindungsname, Richtung, Rollennamen
 - Das Objektdiagramm stellt nur einen Schnappschuß dar



Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

AUF GEHT'S!!

SELBER MACHEN UND LERNEN!!

