### Lecture 06: Class Diagrams I

#### Course Map

- **Software Design, Modelling and Analysis in UML**
- **Lecture 06: Class Diagrams I**
- **Prof. Dr. Andreas Podelski, Dr. Bernd Westphal**
  - **Albert-Ludwigs-Universität Freiburg, Germany**

#### UML ModelInstances

- **N S W E**
- **CD, SM**
- **S = (T, C, V, atr)**
- **SM M = (Σ DS, A S, \(\rightarrow\) SM)**
- **ϕ ∈ OCL expr**
- **CD, SD S, SD B = (Q SD, q 0, A S, \(\rightarrow\) SD, F SD)**
- **π = (σ 0, ε 0) (cons 0, Snd 0) \(\rightarrow\) u 0 (σ 1, ε 1) \(\cdots\) w \(\rightarrow\) u n (σ i, cons i, Snd i) \(i \in N\)**
- **G = (N, E, f)**

#### Mathematics

- **OD UML**
- **Contents & Goals**
  - **Last Lecture:**
    - OCL Semantics
    - Object Diagrams
  - **This Lecture:**
    - Educational Objectives:
      - Capabilities for following tasks/questions.
      - What is a class diagram?
      - For what purposes are class diagrams useful?
      - Could you please map this class diagram to a signature?
      - Could you please map this signature to a class diagram?
    - **Content:**
      - Final notes on object diagrams.
      - Study UML syntax.
      - Prepare (extend) definition of signature.
      - Map class diagram to (extended) signature.
      - Stereotypes – for documentation.

---

#### Example: Object Diagrams for Documentation

- **The Other Way Round**
- **If we only have a picture as below, we typically assume that it's meant to be an object diagram wrt. some signature and structure.**
- **u 1: C u 2: C u 3: D z = 0**
- **Example: Object Diagrams for Documentation**
What Do We (Have to) Cover?

- A class
- Stereotypes
- Name
- Package
- Abstract
- Active
- Operations
- Attributes
- Visibility
- Name
- Type
- Multiplicity
- Order
- Initial Value
- Properties (read-only, ordered, composite)

Wanted: places in the signature to represent the information from the picture.
Mapping UML CDs to Extended Signatures

From Class Box to Extended Signatures

Convention

Extend stereotypes (names of stereotypes) to choose from. But: to unimportant to care.)

All definitions we have up to now principally still apply to represent class diagram, e.g. no "place" to put class.

If the new aspects are irrelevant (for a given context), we simply write or attribute:

- e.g. OCL expresions.

Alternatively, we could add a set of properties: {$a$} := {

- $\forall v, a \in S$:

• $v$,

- $\exists v$, $a \in S$

If details are irrelevant.

The other way round:

- • $v$

Or:

- • $v$

From now on, we assume that each class has:

• $\forall v, a \in S$:

- $\exists v$, $a \in S$

From now on, we assume that each attribute:

• $\forall v, a \in S$:

- $\exists v$, $a \in S$

If the new aspects are irrelevant (for a given context), we simply write or attribute:

- e.g. OCL expresions.

Alternatively, we could add a set of properties: {$a$} := {

- $\forall v, a \in S$:

• $v$

Or:

- • $v$

From now on, we assume that each class has:

• $\forall v, a \in S$:

- $\exists v$, $a \in S$

