

Programmiermethodik

Übung 12

Sommersemester 2010
Fachgebiet Software Engineering

Andreas Scharf
andreas.scharf@cs.uni-kassel.de

Agenda

- **Organisatorisches**
- **Vorstellung HA 10**
- **Vererbung und Polymorphie I**
- **Praktische Übung I**
- **Vererbung und Polymorphie II**
- **Praktische Übung II**

Organisatorisches

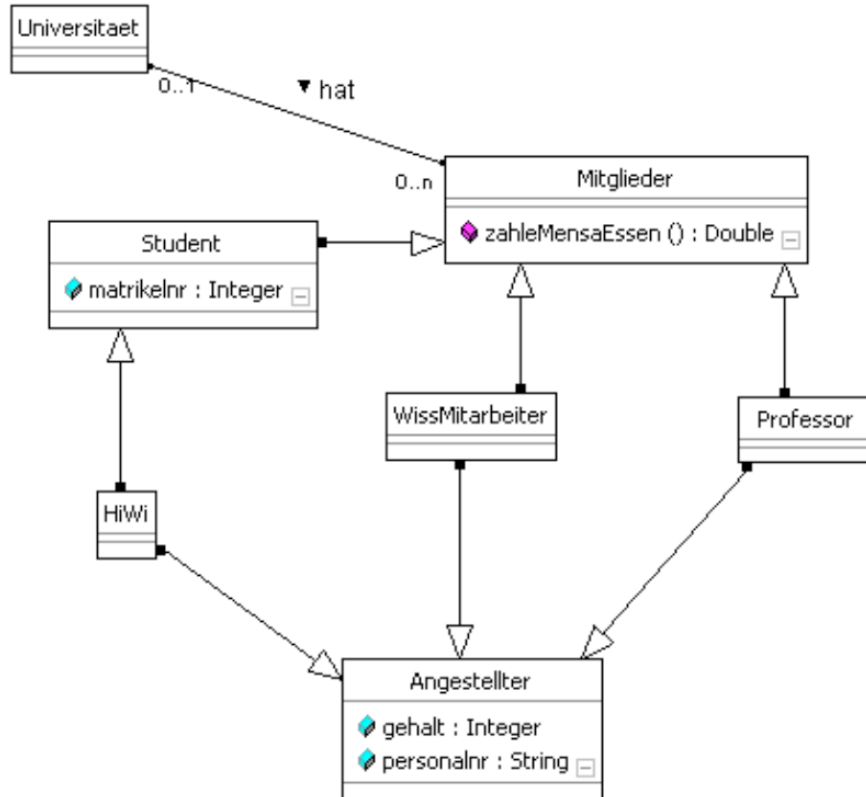
- **HA 10 ist das letzte Hausaufgabenblatt**
- **Am 16.07.2010 wird ein Zusatzblatt geben**
 - Um weitere Punkte zu holen (ein schlechteres Blatt ausgleichen)
 - Klausurzulassung erlangen wenn man schon 3x n.a. oder <50% hat
- **Nächsten Freitag (16.07.2010):**
 - Weitere Informationen zum kommenden SE1
- **Übungsbewertung:**
 - Es werden die besten 10 Abgaben bewertet.
 - Nicht abgegeben = 0%.
 - Mit dem Zusatzblatt kann man das schlechteste (oder ein nicht abgegebenes) Blatt ausgleichen
 - Es wird NICHT gerundet! Beispiel: 88% = 2 Notensprünge.

Vorstellung HA 10

- **Wiederholung mit German BBQ:**
 - Aufgabe 1: Projekt anlegen
 - Aufgabe 2: Textuelles Szenario auf Englisch
 - Aufgabe 3: Objektdiagramme
 - Aufgabe 4: Klassendiagramm
 - Aufgabe 5: Storyboard
 - Aufgabe 6: Storydiagramm
- **Sinn: Kompletten Workflow nochmal (in Minimalkonfiguration) durchlaufen**

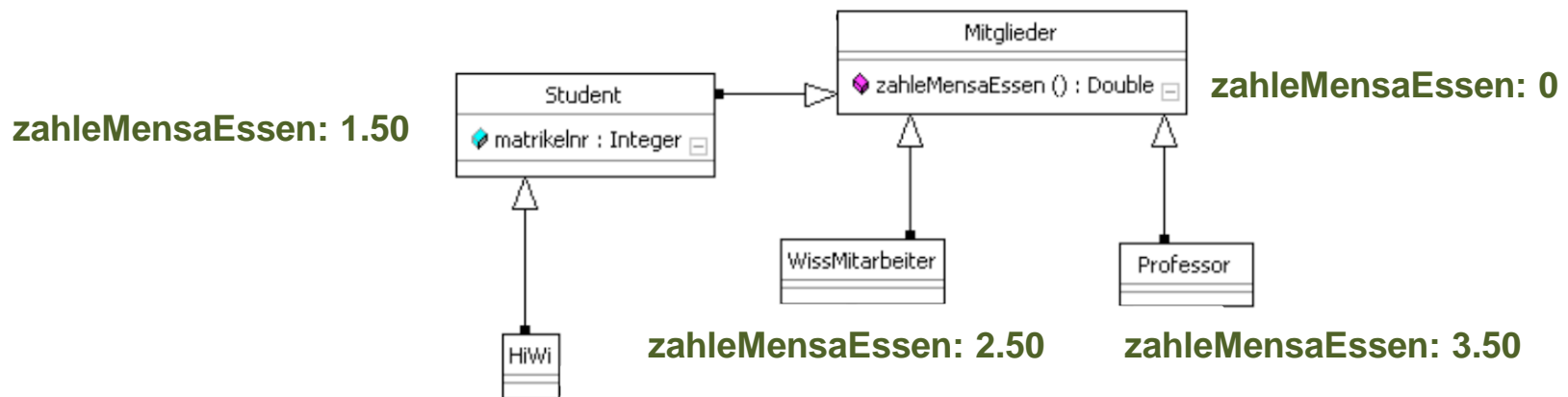
Vererbung und Polymorphie I

- Klassendiagramm mit Mehrfachvererbung und Polymorphie



Vererbung und Polymorphie I

- **Polymorphie:** „Fähigkeit eines Bezeichners, abhängig von seiner Verwendung unterschiedliche Datentypen anzunehmen.“
- **Polymorphe Methoden:**
 - Treten immer im Zusammenhang mit Vererbung und Schnittstellen auf
 - Eine Methode ist polymorph, falls sie in verschiedenen Klassen in der (Vererbungs-)hierarchie die gleiche Signatur hat



Praktische Übung I

- Implementiert die Klassen **Universitaet**, **Mitglieder**, **Student**, **WissMitarbeiter** und **Professor** in Java (von Hand **OHNE** Fujaba!)

- Attribut und Methode nicht vergessen

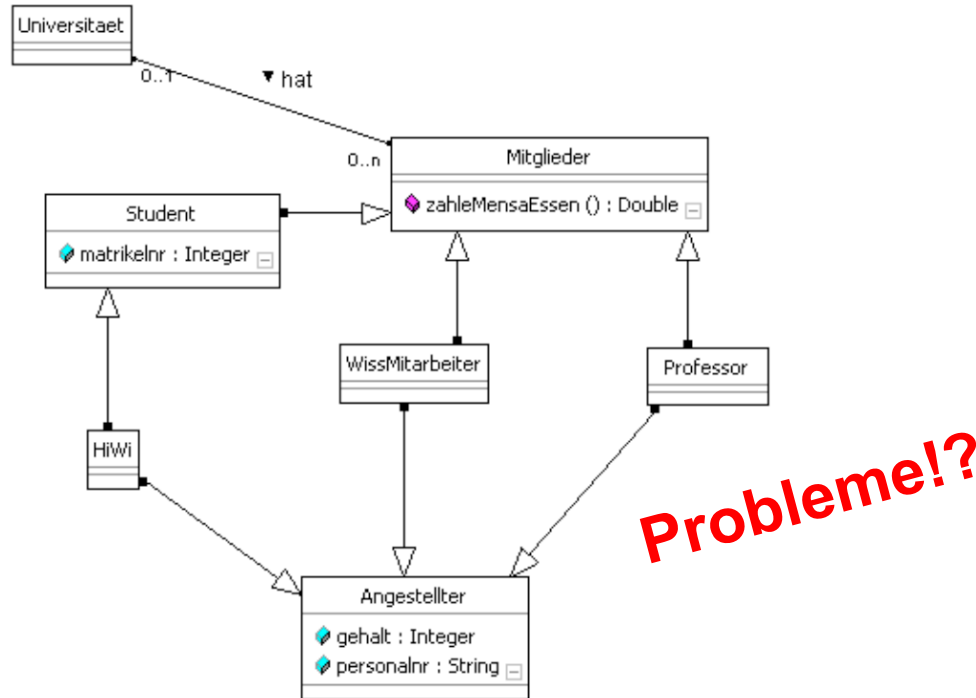
- **Erstellt eine Klasse mit einer main() Methode:**

Mitglied m = new Mitglied();	
System.out.println(m.zahleMensaEssen());	➔ 0
Mitglied m2 = new Professor();	
System.out.println(m2.zahleMensaEssen());	➔ 3.50
Mitglied m3 = new WissMitarbeiter();	
System.out.println(m3.zahleMensaEssen());	➔ 2.50
Mitglied m4 = new Student();	
System.out.println(m4.zahleMensaEssen());	➔ 1.50

- **Zeigt das Ergebnis einem Betreuer!**
- **Zeit 20 Minuten.**

Vererbung und Polymorphie II

- **Mehrfachvererbung:**



- **Java unterstützt keine Mehrfachvererbung!**
- **Lösung: Verwendung von Interfaces (... implements) statt Vererbung (... extends)**

Praktische Übung II

- Erstellt die Klassen Hiwi und Angestellter
- Probiert

```
public class Hiwi extends Student, Angestellter{  
  
}
```

Welchen Compilefehler ergibt dies?

- **Konvertiert die Klasse Angestellter zu einem Interface**
 - Benutzt nun statt `extends` das Schlüsselwort `implements`
 - Fügt dem Interface die Methoden `getGehalt()` und `getPersonalNr()` hinzu
- **Implementiert die Methoden aus dem Interface Angestellter in den drei implementierenden Klassen. Fügt dort auch die passenden Attribute hinzu.**

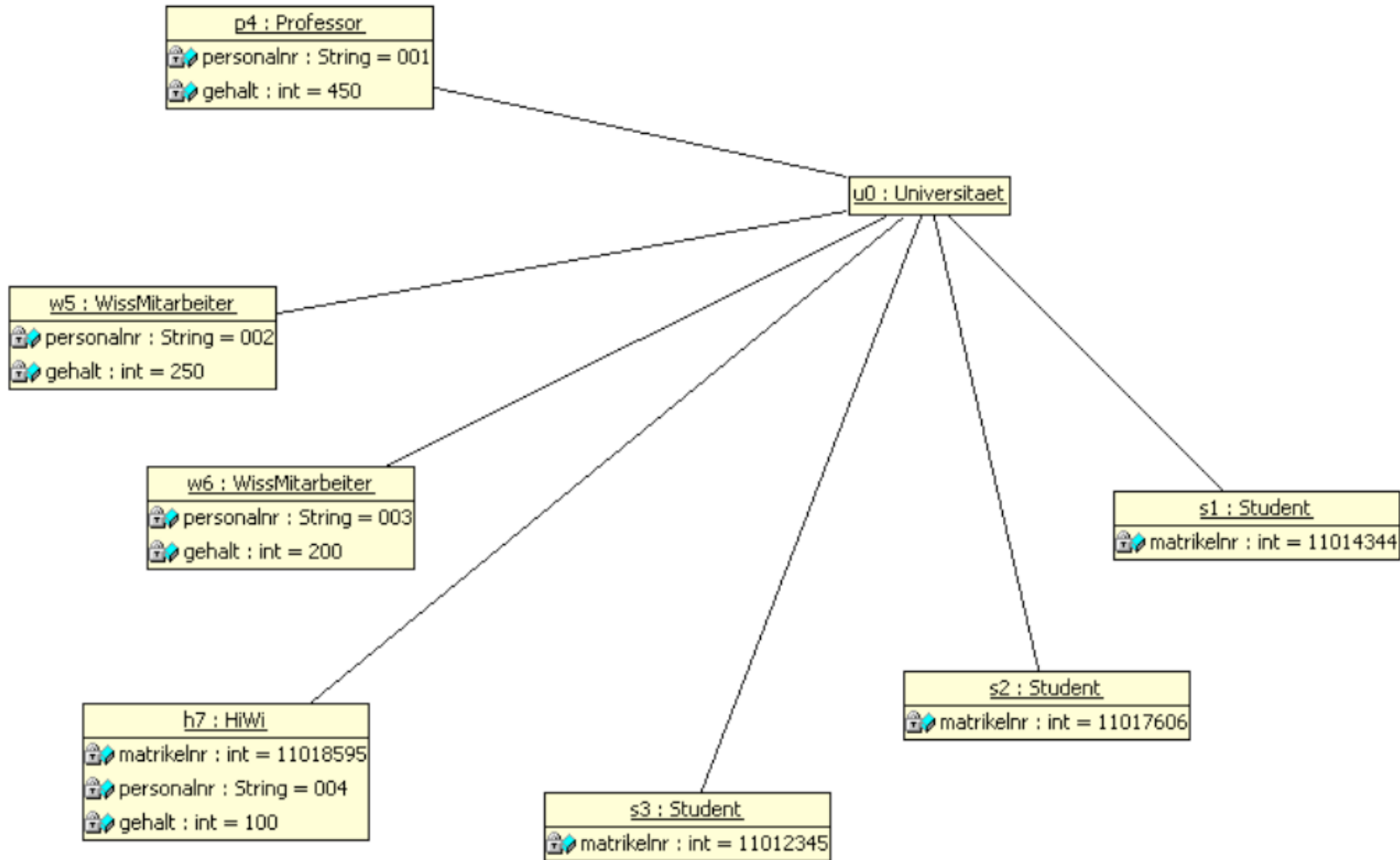
Praktische Übung III

- Implementiert eine neue Klasse mit einer *main()*-Methode, welche die Objektstruktur, wie in nachfolgender Folie gezeigt, erzeugt. Versuche Variablen so generell wie möglich zu deklarieren. Lassen sich dann alle Attribute setzen? Verwende Casts wie:

```
Mitglied m = new Student();  
Student s = (Student) m;  
s.matrikelnr = 11012345;
```

- Zeige den Quellcode und das eDOBS-Diagramm dem Betreuer.

Praktische Übung IV



Ende

Schönes WE!