UNIKASSEL VERSITÄT

SOFTWARE ENGINEERING
UNIVERSITÄT KASSEL

Programmiermethodik

Übung 13

Wintersemester 2012 / 13

Fachgebiet Software Engineering

Tobias George

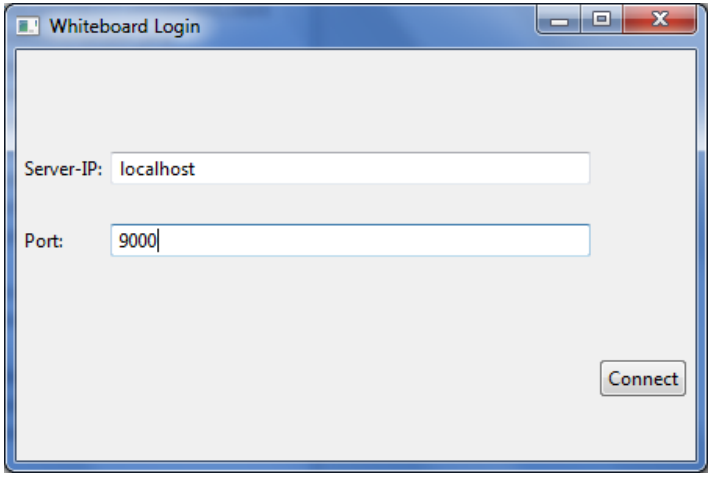
george@uni-kassel.de

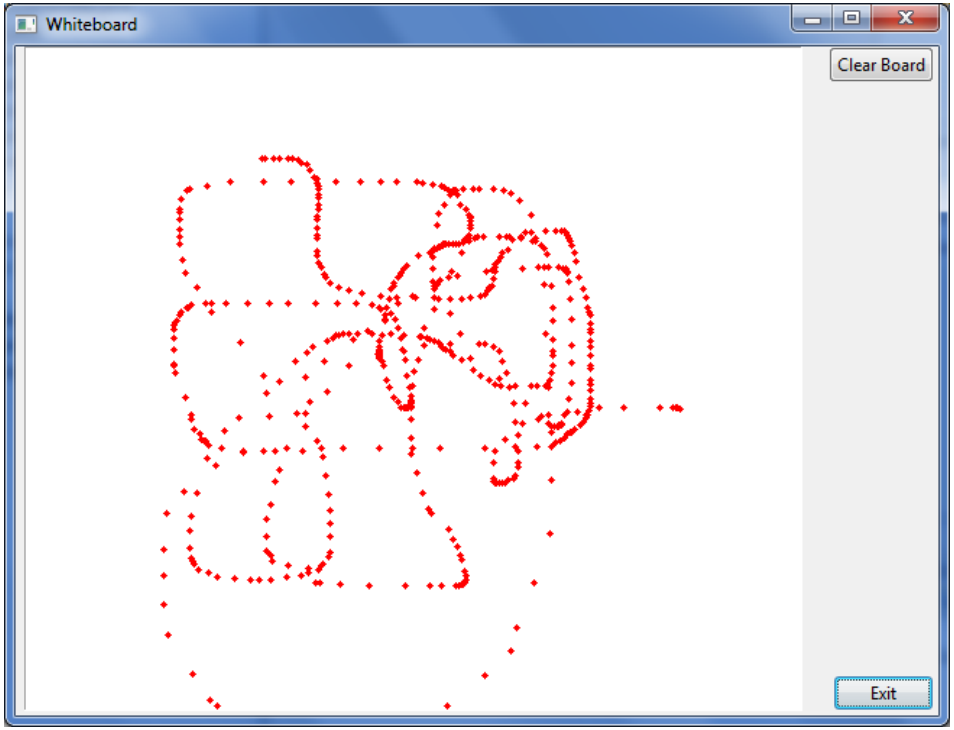
Agenda

- Besprechung HA 11
- Organisatorisches
- Organisatorisches zur Klausur
- Praktische Übung Vererbung und Polymorphie
- Zeit für eure Fragen

Vorstellung Musterlösung HA 11

- **Multiuser Whiteboard**



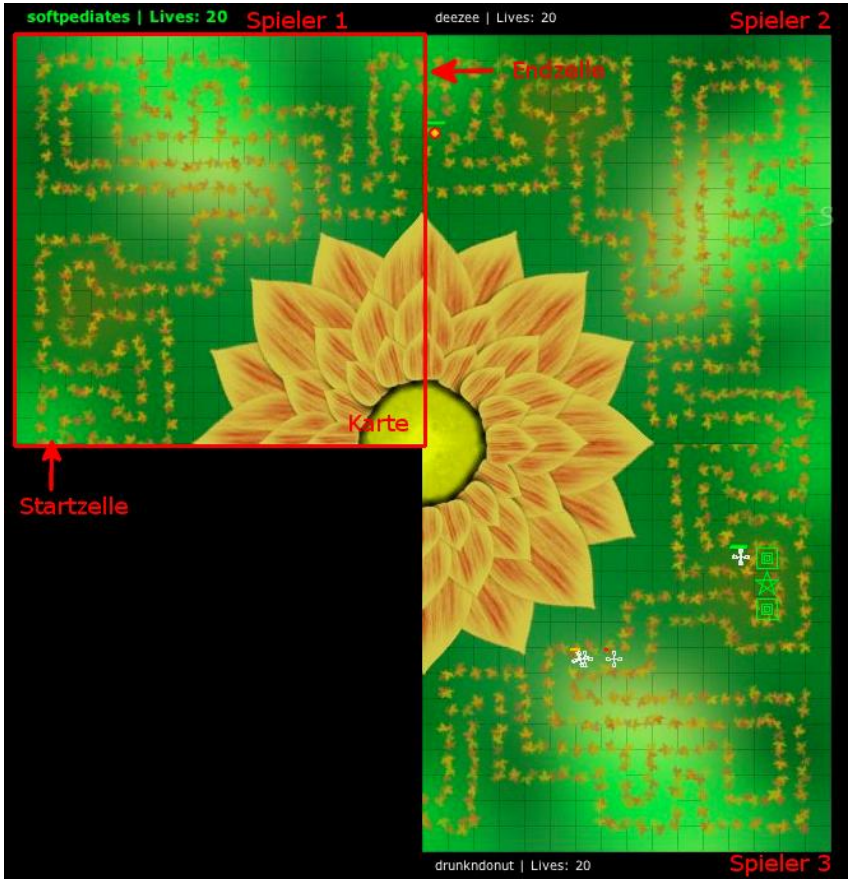


Organisatorisches I

- **Übungsbewertung:**
  - Es werden die acht besten Abgaben bewertet.
  - Nicht abgegeben entspricht 0%, fließt aber nicht in die Wertung ein.
  - Mit dem Zusatzblatt kann man das schlechteste (oder ein nicht abgegebenes) Blatt ausgleichen
  - Es wird nur abgeschnitten, NICHT gerundet! Beispiel: 89.99% = 2 Notensprünge.

Organisatorisches II

- **Software Engineering 1 im nächsten Semester**
  - Teamarbeit („zufällige“ Einteilung, Paare dürfen gewählt werden)
  - Erstellung eines Spiele Clients
    - Letztes Semester: CreepSmash



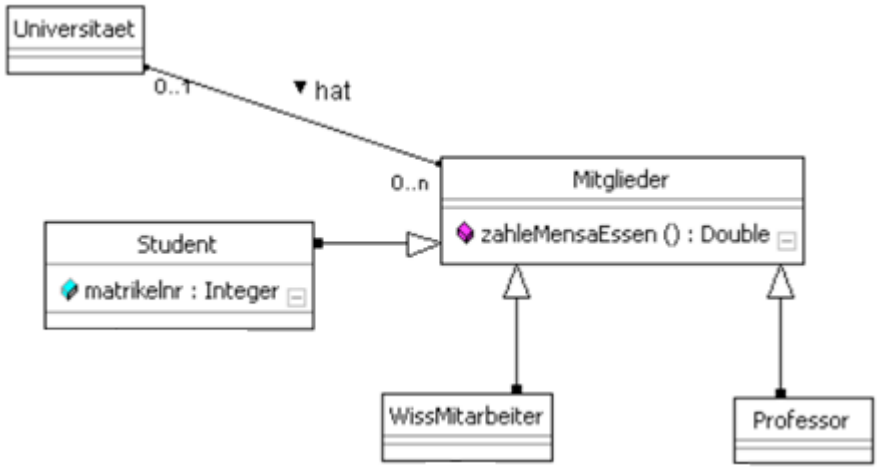
Organisatorisches zur Klausur

- **Datum, Uhrzeit: 26.03.2012, 9:00-11:00 Uhr, ca. 90 min.**
- **Ort:**
  - Raum R 0425 (Hörsaal bei der Bibliothek)
  - bzw. zusätzlich je nach Teilnehmerzahl R 0446

- **Konflikte mit anderen Klausuren??**
  - Klausuren aus Grundstudium wh. nicht rivalisierend
  - Wahlpflicht Klausuren müssen sich nach GS-Klausuren richten
  - Bei Gleichwertigkeit: Die teilnehmerärmere Klausur gibt nach

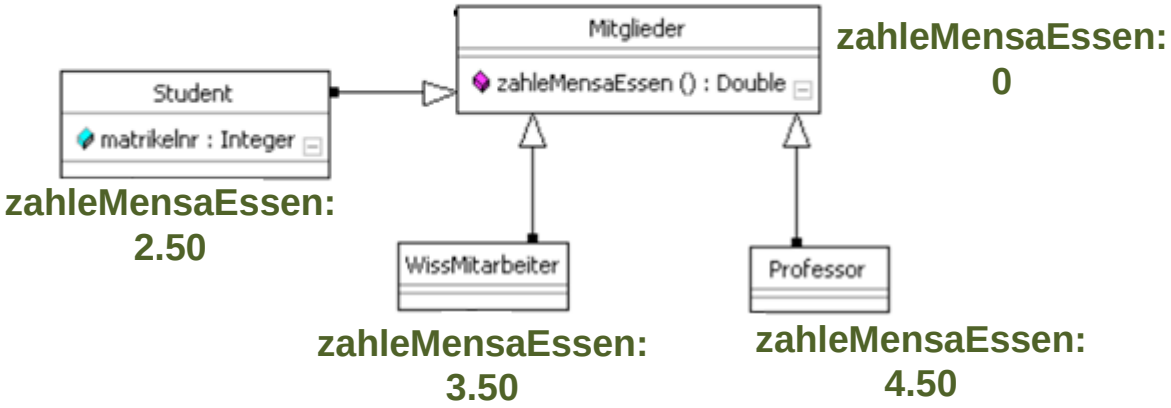
Praktische Übung I

- **Klassendiagramm mit Vererbung und Polymorphie**



Praktische Übung II

- **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 III

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

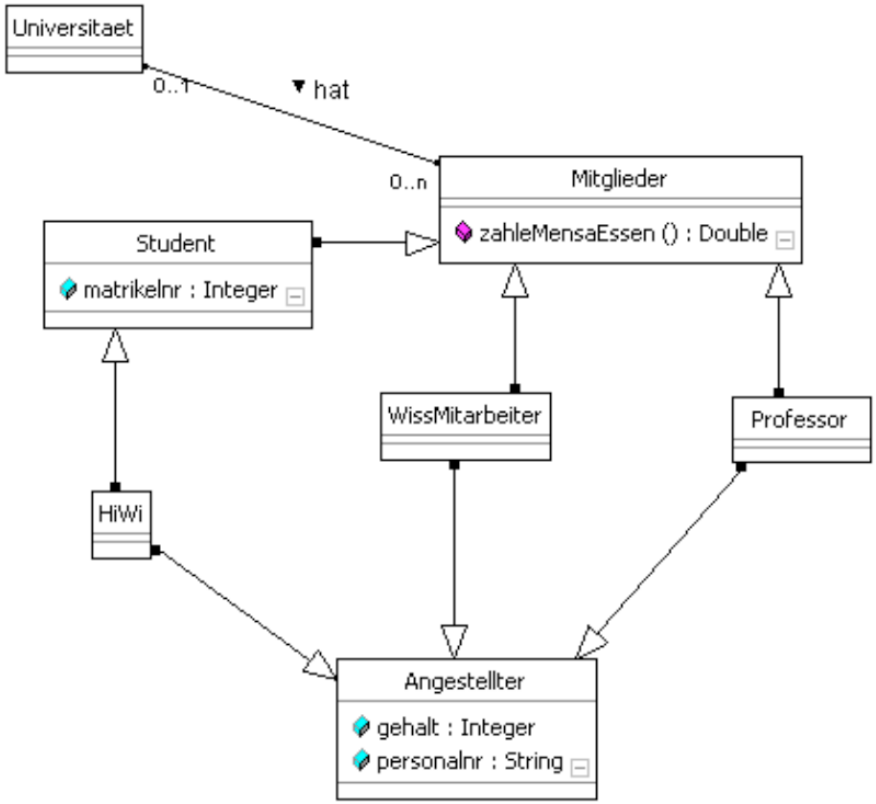
```
Mitglied m3 = new WissMitarbeiter();
System.out.println(m3.zahleMensaEssen());
```
⇨ **2.50**

```
Mitglied m4 = new Student();
System.out.println(m4.zahleMensaEssen());
```
⇨ **1.50**

- **Zeigt mir die Konsolenausgabe!**
- **Zeit: 20 min.**

Praktische Übung IV

- **Klassendiagramm mit Mehrfachvererbung und Polymorphie**



Praktische Übung V

- **Erstellt die Klassen Hiwi und Angestellter**
- **Probiert:**

```
public class Hiwi extends Student, Angestellter{

}
```

  Welchen Compilefehler ergibt dies?
- **Konvertiert die Klasse Angesteller zu einem Interface und:**
  - 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 abgeleiteten Klassen. Fügt dort auch die passenden Attribute hinzu.**

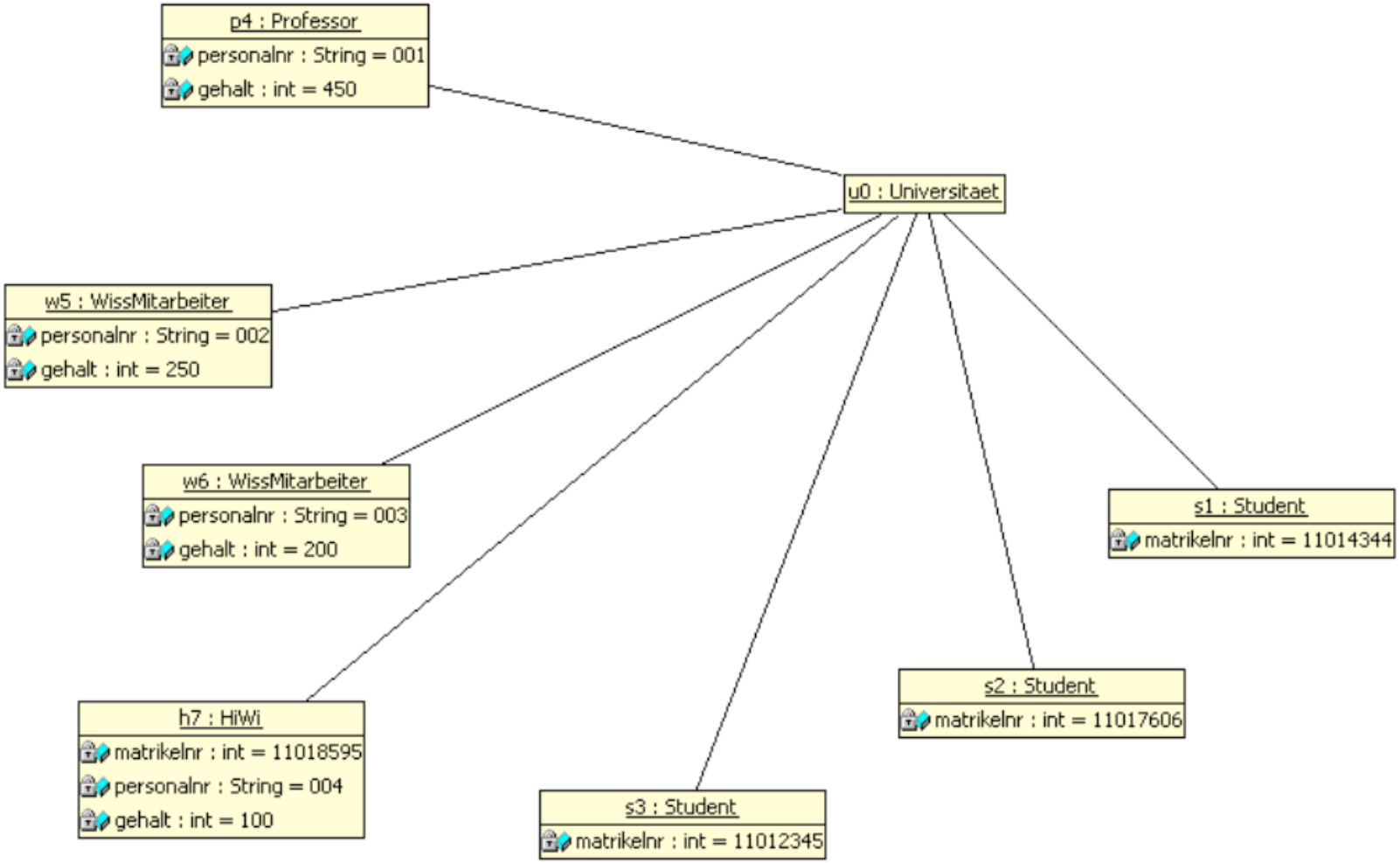
Praktische Übung VI

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

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

- Zeigt mir den Quellcode und das eDOBS-Diagramm!
- Zeit: 20 min.

Praktische Übung VII



Zeit für eure Fragen

Ende

Danke für die Aufmerksamkeit
und
eine schöne
vorlesungsfreie Zeit