

## Software Engineering

# Praktikumsversuch 1 Gruppe C - Hausaufgaben -

**Abgabe der Lösungen:** Freitag, 14.11.2014, 9 Uhr, in Ilias im Kurs zu dieser Veranstaltung -> Praktikum -> Gruppe C -> Ihr Team -> Blatt 1

**Wichtig:** Da sich die Anwesenheitsaufgaben auf die Lösungen dieses Blattes beziehen, müssen Sie alle Dokumente und Dateien, die Sie bei der Lösung der Hausaufgaben erstellt haben, zum Anwesenheitstermin mitbringen! Ebenso sollten Sie dieses Aufgabenblatt zusammen mit dem Lastenheft zum Anwesenheitstermin mitbringen.

**Ziel:** Spezifikation von Daten, Klassendiagramm implementieren

Sie erhalten als Anlage das Lastenheft für das System UniIS. In diesem sind Anforderungen an die Systemfunktionen und die Systemdaten enthalten.

## H 1.1 Klassendiagramm mit Entitätsklassen erstellen

Für das System UniIS sollen sie einige Klassen identifizieren. Wir beschränken und zunächst auf Entitätsklassen. Entitätsklassen dienen zum langfristigen Speichern der Daten des Systems. Es müssen also alle Informationen, die dem System dauerhaft zur Verfügung stehen sollen, durch entsprechende Klassen und ihren Beziehungen modelliert werden. Welche konkreten Informationen zum Betrieb des Systems dauerhaft benötigt werden, ist den Anwendungsfällen im Lastenheft zu entnehmen.

**Ihre Aufgabe:** Erzeugen Sie ein UML-Klassendiagramm mit allen Entitätsklassen für das System UniIS.

Die Entitätsklassen sollen alle erforderlichen Attribute und Assoziationen inkl. Multiplizitäten enthalten.

Sie müssen keine Operationen an den Entitätsklassen definieren.

Erstellen Sie die Entitätsklassen in einem geeigneten Unterpaket von `se.uniis`.

Sie können hierzu Microsoft Visio verwenden (verwenden Sie folgende Shapes:

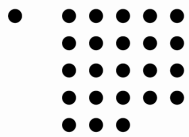
Software->UML-Modelldiagramm->UML-Statische Struktur). Diese Software ist von der Campus IT verfügbar (ELMS für Dreamspark Premium).

## H 1.2 Klassendiagramm in Java implementieren

Implementieren Sie alle Klassen des Klassendiagramms aus Aufgabe H 1.1 in Java.

Sie müssen hierzu NetBeans in der vorgeschriebenen Version verwenden!

Erstellen Sie Konstruktoren, Getter, Setter und die Einfüge-Operationen für mehrwertige Assoziationen.



## Lastenheft UnilS

### Inhaltsverzeichnis

1. Produkteinsatz.....	2
2. Produktfunktionen .....	2
3. Produktdaten .....	4

## 1. Produkteinsatz

Es soll ein Informationssystem (mit Namen UnilS) für eine Hochschule erstellt werden.

## 2. Produktfunktionen

/LF10/ Muss

**Anwendungsfall:** zu Kurs anmelden

**Akteure:** Student

**Beschreibung:** Ein Student soll sich zu einem Kurs anmelden können. Hierzu soll der Student anhand der Matrikelnummer und der Kurs anhand der Kursnummer gesucht werden. Es soll dann im System gespeichert werden, dass der Student sich zu diesem Kurs angemeldet hat. Weitere Informationen wie Datum o.ä. muss im System nicht gespeichert werden. Der Student erhält eine Anmeldebescheinigung mit seiner Matrikelnummer, der Kursnummer und dem Anmeldedatum.

/LF20/ Muss

**Anwendungsfall:** Prüfungsergebnis registrieren

**Akteure:** Professor

**Beschreibung:** Ein Professor soll für einen Studenten eine abgelegte Prüfung zu einem Kurs registrieren können. Hierzu soll die Matrikelnummer und die Kursnummer vom Professor eingegeben werden. Es soll dann der entsprechende Kurs und der zugehörige Student gesucht werden. Es ist zu prüfen, ob der Student für den Kurs angemeldet war. Falls er nicht angemeldet ist, muss der Anwendungsfall abgebrochen werden. Ist er zu dem Kurs angemeldet, so soll eine Prüfungsmeldung im System gespeichert werden. An der Prüfungsmeldung sind das Datum und die Note zu speichern. Weiterhin wird für jede Meldung eine eindeutige Meldungsnummer vergeben. Es wird für den Studenten ein Prüfungsschein mit der Kursnummer, dem Datum, der Note und der Meldungsnummer erstellt. Abschließend wird die bestehende Anmeldung des Studenten zu diesem Kurs entfernt.

/LF30/ Muss

**Anwendungsfall:** Immatrikulationsbescheinigung drucken

**Akteure:** Sachbearbeiter

**Beschreibung:** Ein Sachbearbeiter soll für einen Studenten eine Immatrikulationsbescheinigung erstellen können. Diese soll den Namen des Studenten und seine Matrikelnummer enthalten.

/LF60/ Muss

**Anwendungsfall:** Neuen Studenten registrieren

**Akteure:** Sachbearbeiter, Student

**Beschreibung:** Ein Sachbearbeiter soll einen neuen Studenten registrieren können. Es werden der Name und die Anschrift aufgenommen und eine Matrikelnummer vergeben.

/LF70/ Muss

**Anwendungsfall:** Anmeldungen suchen

**Akteure:** Sachbearbeiter

**Beschreibung:** Der Sachbearbeiter soll nach allen im System registrierten Anmeldungen suchen können.

/LF80/ Muss

**Anwendungsfall:** Anmeldungen eines Studenten ermitteln

**Akteure:** Sachbearbeiter

**Beschreibung:** Der Sachbearbeiter soll für einen durch seine Matrikelnummer identifizierten Studenten alle seine Anmeldungen ermitteln können.

/LF90/ Muss

**Anwendungsfall:** Anmeldungen zu einem Kurs ermitteln

**Akteure:** Professor

**Beschreibung:** Ein Professor soll für einen Kurs alle angemeldeten Studenten ermitteln können.

/LF100/ Muss

**Anwendungsfall:** Prüfungen eines Studenten ermitteln

**Akteure:** Sachbearbeiter

**Beschreibung:** Der Sachbearbeiter soll für einen durch seine Matrikelnummer identifizierten Studenten alle seine abgelegten Prüfungen ermitteln können.

/LF110/ Muss

**Anwendungsfall:** Prüfungen zu einem Kurs ermitteln

**Akteure:** Professor

**Beschreibung:** Ein Professor soll für einen Kurs alle Studenten ermitteln können, die in diesem Kurs eine Prüfung abgelegt haben.

/LF130/ Muss

**Anwendungsfall:** Neuen Kurs erfassen

**Akteure:** Professor

**Beschreibung:** Ein Professor soll einen neuen Kurs im System erfassen können. Er soll hierzu den Titel des Kurses, den Namen des Dozenten und die Kursnummer eingeben können.

### **3. Produktdaten**

/LD10/ Studentendaten (50.000)

/LD20/ Kursdaten (9.000)

/LD30/ Prüfungsmeldungen (3.000.000)

/LD40/ Anmeldungen (1.000.000)