

Übungen zur Vorlesung Softwaretechnologie

- Wintersemester 2018/19-

Dr. Günter Kniesel

Übungsblatt 6 - Lösungen

Aufgabe 1. *Domain Object Modell*(16 Punkte)

Stellen Sie sich vor, Sie sollen für die Informatik-Fachschaft ein Programm zur Tutoriums-Wahl entwerfen, das sich Ihre Auftraggeber wie folgt vorstellen:

Das Programm besteht aus einem Eingabe-Fenster, nämlich der Veranstaltungs-Auswahl und zwei Ausgabe-Fenstern, der Stundenplan-Ansicht und einer Hilfeseite.

Zuerst wählen die Studierenden in der Veranstaltungs-Auswahl ihre Veranstaltungen aus. Veranstaltungen sind Vorlesungen oder Seminare. Die Veranstaltungen haben als Veranstaltungsdaten einen Titel, eine Uhrzeit und eine Raumangabe, Seminare haben zusätzlich eine Maximal-Teilnehmerzahl.

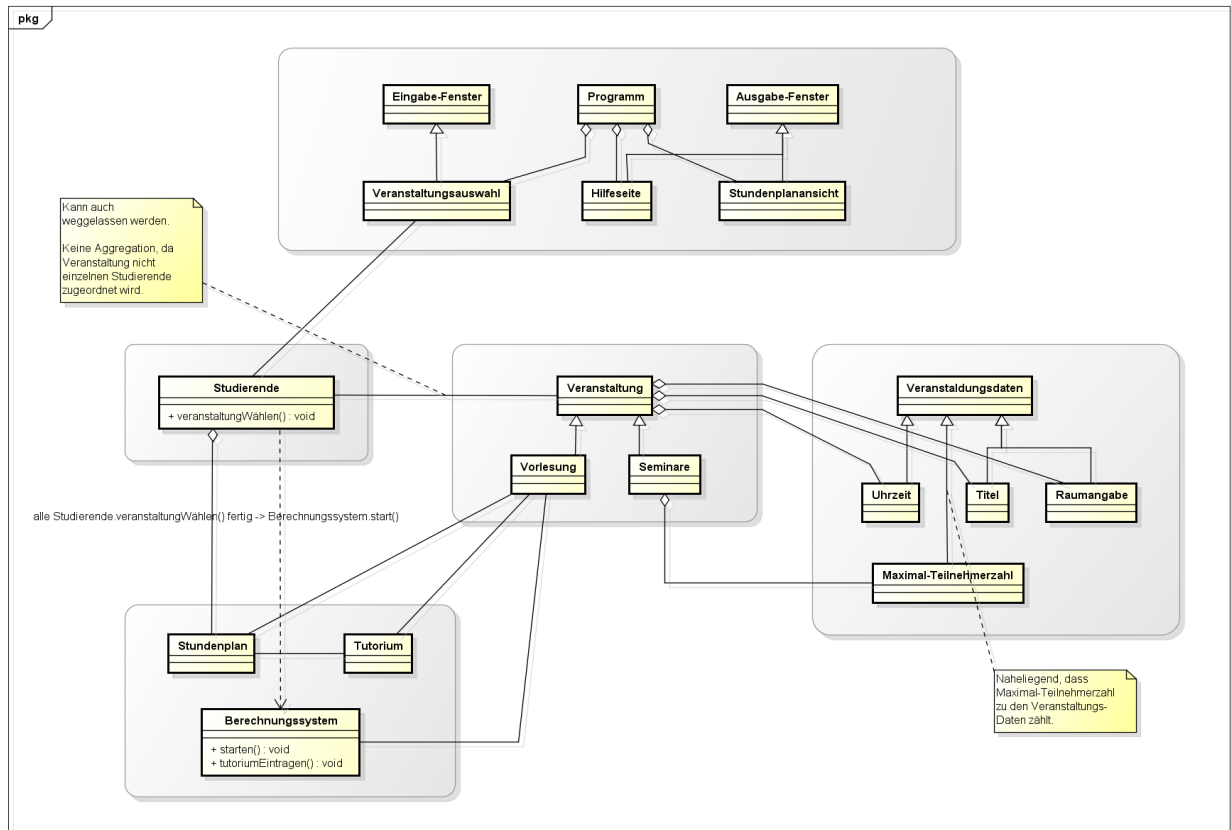
Nachdem alle Studierenden ihre Veranstaltungen ausgewählt haben, muss das Berechnungssystem starten, das in die Stundenpläne der Studierenden für jede Vorlesung ein zugehöriges Tutorium einträgt.

- a) (6 Punkte) Analysieren Sie obige Beschreibung mit Hilfe von Abbotts Technik und stellen sie die Ergebnisse tabellarisch dar. Hinweis: Geben Sie Ihre Lösung als PDF-Dokument ab.

Aussage	Wort	Wortart	Modellelement
Das Programm besteht aus einem Eingabe-Fenster, nämlich der Veranstaltungs-Auswahl und zwei Ausgabe-Fenstern, der Stundenplan-Ansicht und einer Hilfeseite.	Programm, Eingabe-Fenster, Veranstaltungs-Auswahl, Ausgabe-Fenster, Stundenplan-Ansicht, Hilfeseite	Nomen	Klassen
	... besteht aus ...	Verb (haben)	Aggregation
	Eingabe-Fenster [...] Veranstaltungsauswahl, [...] Ausgabe-Fenster [...] Stundenplan-Anzeige [...] Hilfeseite	(verstecktes) Verb (sein)	Vererbung
Zuerst wählen die Studierenden in der Veranstaltungs-Auswahl ihre Veranstaltungen aus.	Studierenden, Veranstaltungen	Nomen	Klassen
	wählen	Verb	Methode
	Die <i>Studierenden</i> ... [wählen] in der Veranstaltungsauswahl ... aus, in der <i>Veranstaltungs-Auswahl</i> ... <i>Veranstaltungen</i>		
Veranstaltungen sind Vorlesungen oder Seminare.	Vorlesungen, Seminare	Nomen	Klassen
	Sind	Verb (sein)	Vererbung

Die Veranstaltungen haben als Veranstaltungsdaten einen Titel, eine Uhrzeit und eine Raumangabe, Seminare haben zusätzlich eine Maximal-Teilnehmerzahl.	Veranstaltungsdaten, Titel, Uhrzeit, Raumangabe, Maximal-Teilnehmerzahl	Nomen	Klassen
	Haben ...	Verb (haben)	Aggregation
	... als ...	(verstecktes) Verb (sein)	Vererbung
Nachdem alle Studierenden ihre Veranstaltungen ausgewählt haben, muss das Berechnungssystem starten, das in die Stundenpläne der Studierenden für jede Vorlesung ein zugehöriges Tutorium einträgt	Berechnungssystem, Stundenpläne, Tutorium	Nomen	Klassen
	... muss das Berechnungssystem ...	Einschränkung	Constraint
	... starten ..., ... einträgt	Verben	Methoden
	<i>Studierende</i> [tragen] ... <i>Veranstaltungen</i> [ein], [<i>das Berechnungssystem trägt</i>] in die <i>Stundenpläne</i> ... ein, ... <i>Vorlesung</i> zugehöriges <i>Tutorium</i> ... <i>das Berechnungssystem</i> [trägt] ... <i>Tutorien</i> ein		

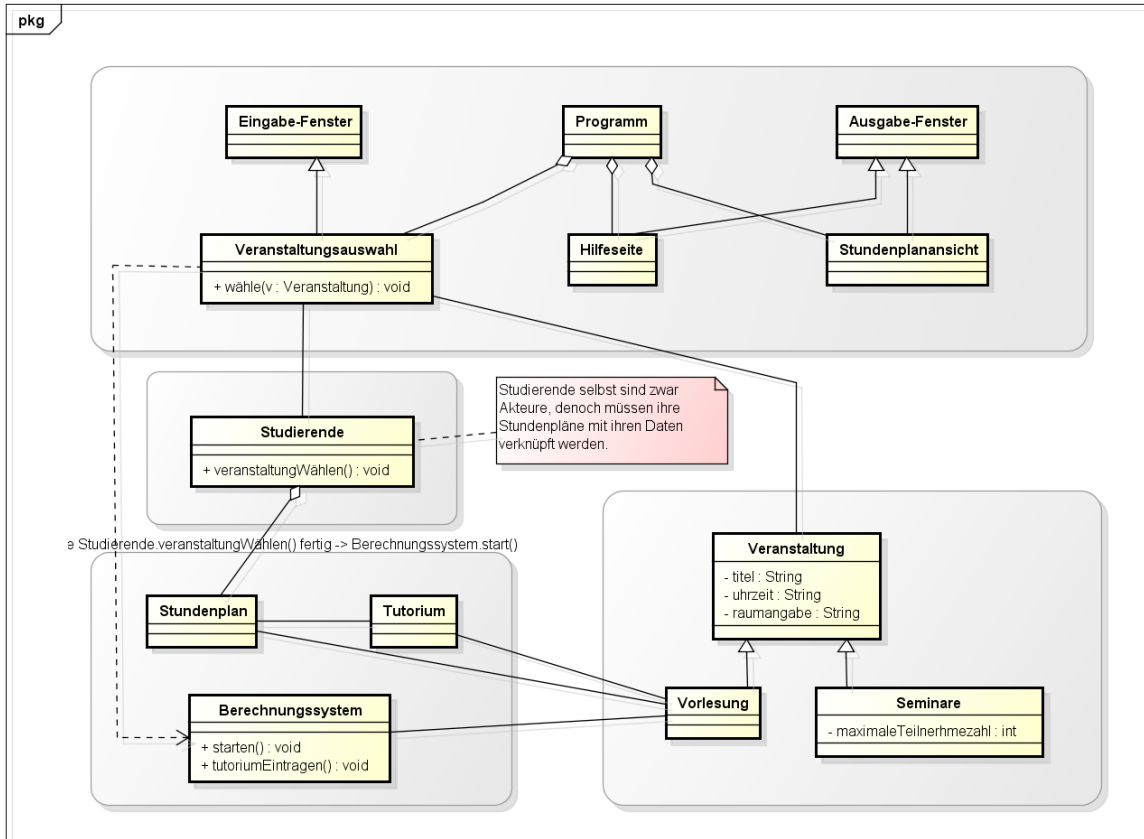
b) (6 Punkte) Setzen Sie die Ergebnisse der textuellen Analyse aus Aufgabe a) zunächst schematisch (1-zu-1) in ein Domain Object Model (UML-Klassendiagramm mit Attributen und Assoziationen zwischen den Klassen) um.



powered by astah

c) (4 Punkte) Überdenken Sie das Domain Object Model und fassen die Ergebnisse aus Aufgabe b) zusammen.

- Klassen zu Attributen konvertieren, wenn Klasse kein eigenes Verhalten besitzt
Klasse als einzige Information ihre Klassenzugehörigkeit liefert
- Akteure entfernen, da sie außerhalb des Systems stehen.
Achtung: Datenklassen zu den Akteuren einführen, falls Daten modelliert werden sollen.
- Abhängigkeit zwischen Veranstaltungsauswahl und Berechnungssystem ist abstrahiert.
Wir legen uns nicht fest, wie das Berechnungssystems konkret startet.



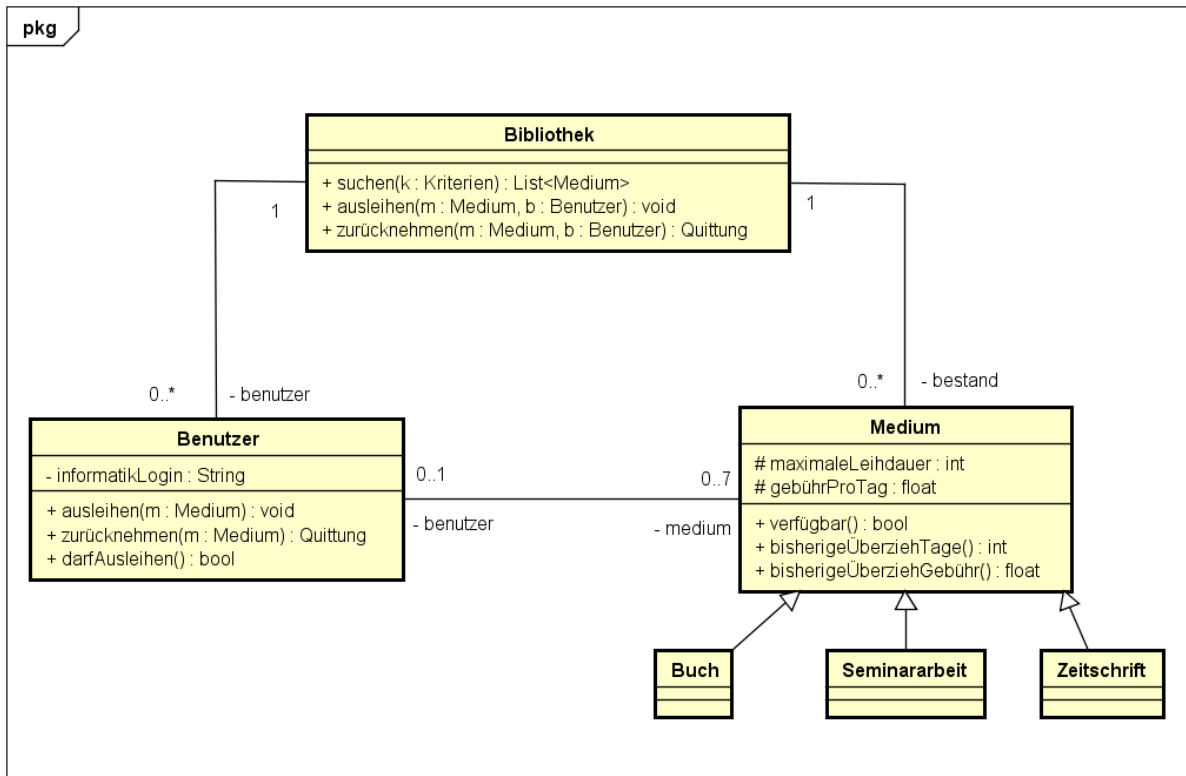
powered by astah

Aufgabe 2. Design by Contract (7 Punkte)

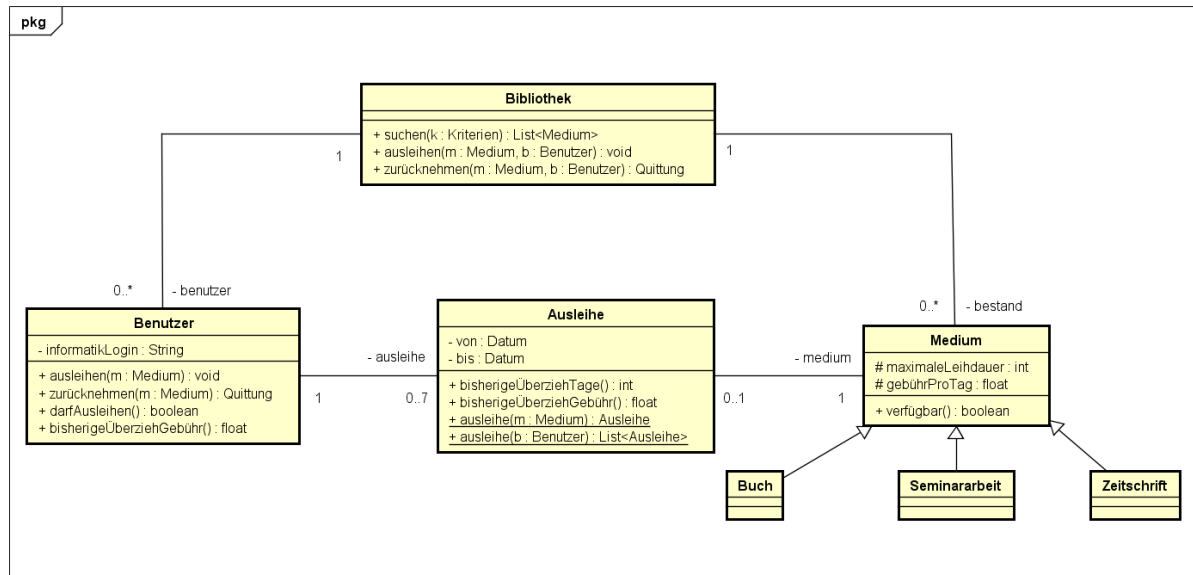
Verfeinern Sie Ihren Entwurf von *Blatt 5 Aufgabe 3* wie folgt:

- (4 Punkte) Erstellen Sie ein Klassendiagramm, das die Ergebnisse der CRC-Analyse widerspiegelt, aber komplett typisiert ist (Ergebnis- und Parametertypen zu jeder Operation). Achten Sie darauf, dass manche Beziehungen evtl. eigene Attribute und Operationen besitzen.

Lösungsvariante 1: Eins-zu-eins Übertragung des CRC-Modells, mit Überschreitungsdauer als Methoden im Medium



Lösungsvariante 2: Modellierung der Ausleihe als eigene Klasse und der Überschreitungsdauer als Methoden der Ausleihe, denn ein Medium ist nicht an und für sich überzogen, das macht nur Sinn im Kontext einer Ausleihe. Insbesondere kann die Überschreitungsdauer nur berechnet werden, wenn man das für die betrachtete Ausleihe spezifische ursprünglich geplante Rückgabedatum kennt.



powered by Astah

- a) Beschreiben Sie informell für drei Operationen mit unterschiedlichen Verantwortlichkeiten, jeweils ihre Vor- und Nachbedingungen, die sich aus Aufgabe 2 ableiten lassen. Versuchen Sie anschließend, diese so genau wie möglich zu formulieren, unter Verwendung der Object Constraint Language (OCL).

Lösungsvariante 1: (passend zur ersten Lösungsvariante von Aufgabe a)

```
/* Pre: „true“ (d.h. die Methode braucht keine besondere Vorbedingung)
 * Post: Die Überziehungsgebühr ist das Produkt von Überziehungstagen und Tagesgebühr
 */
```

```
context Medium::bisherigeÜberziehGebühr()
  pre: true
  post: result = self.gebührProTag * self.bisherigeÜberziehTage()
```

```
/* Pre: Das auszuleihende Medium m muss verfügbar sein und der Benutzer b muss die
 * Ausleihberechtigung besitzen
 * Post: Das Medium m kann nicht mehr ausgeliehen werden, es ist an den Benutzer b
 * ausgeliehen und das bedeutet, b ist Benutzer von m und m ist in der Menge der von b
 * ausgeliehenen Medien
 */
```

```
context Bibliothek::ausleihen(Medium m, Benutzer b)
  pre: m.verfügbar() && b.darfAusleihen()
```

```

post:  !(m.verfügbar())
          && m.getBenutzer() = b
          && b.getMedien().contains(m)

/* Pre: Das Medium muss vom entsprechenden Benutzer ausgeliehen sein
* Post: Das Medium m kann wieder ausgeliehen werden, es hat keinen Benutzer mehr,
* ist nicht in der Menge der vom entsprechenden Benutzer ausgeliehenen Medien und diese
* Menge ist um ein Element kleiner, als sie vor der Rückgabe war
*/
context Bibliothek::zurücknehmen(Medium m, Benutzer b)
  pre:  m.getBenutzer() = b && b.getMedien().contains(m)
  post:  m.verfügbar()
          && m.getBenutzer() = null
          && !(b.getMedien().contains(m))
          && b.getMedien().@post.size() = b.getMedien().@pre.size() - 1

```

Lösungsvariante 2 (passend zur zweiten Lösungsvariante von Aufgabe a)

```

/* Pre: „true“ (d.h. die Methode braucht keine besondere Vorbedingung)
* Post: Die Überziehungsgebühr pro Medium ist das Produkt von Überziehungstagen und
* Tagesgebühr des jeweiligen Medium. Die Überziehungsgebühr für den Benutzer ist die
* Summe der Überziehungsgebühren aller zu seinen Ausleihen gehörenden Medien.
*/

```

```

context Benutzer::bisherigeÜberziehGebühr()

```

- a. **pre:** true
- b. **post:** result = self.ausleihe->iterate(
 a : Ausleihe ; -- Laufvariable
 sum : Real = 0 -- Akkumulator mit Anfangswert 0
 | sum + a.medium.gebührProTag *
 a.medium.bisherigeÜberziehTage()
)

← Diese Lösung benutzt die generische iterate()-Funktion die OCL für Collection-Typen bietet. Sie hat zwei Parameter, eine Laufvariable, die über alle Elemente der Collection iteriert und einen Akkumulator. Hinter dem | steht ein Ausdruck der für jeden Wert der Laufvariablen ausgewertet wird und dessen Ergebnis zu dem vorherigen Wert des Akkumulators hinzuaddiert wird. Details finden Sie in der OCL Spezifikation <http://www.omg.org/cgi-bin/doc?ptc/13-08-13.pdf> oder der ihr entsprechenden Dokumentation der OCL Operationen in einer Implementierung wie z.B https://wiki.eclipse.org/Acceleo/OCL_Operations_Reference

```

// Kommentar siehe Lösungsvariante 1

```

```

context Bibliothek::ausleihen(Medium m, Benutzer b)
  a. pre:  m.verfügbar() && b.darfAusleihen()
  b. post:  !(m.verfügbar())
          && Ausleihe.ausleihe(m).benutzer = b
          && Ausleihe.ausleihe(m).medium.contains(m)

```


// Kommentar siehe Lösungsvariante 1

context Bibliothek::zurücknehmen(Medium m, Benutzer b)

a. pre: m.getBenutzer() = b && b.getMedien().contains(m)

b. post: m.verfügbar()

&& Ausleihe.ausleihe(m) = null

&& ! (Ausleihe.ausleihe(m).medium@pre.contains(m))

&& Ausleihe.ausleihe(b).size() = Ausleihe.ausleihe(b)@pre.size() - 1

Aufgabe 3. *Abhängigkeiten* (16,5 Punkte)

a) **(8 Punkte)** Überprüfen Sie die gezeigte Klassendiagramme auf

- **(1,5 Punkte)** Abstraktheit der Pakete
 - i. Paket *Nicht-Studentische Uni-Mitglieder* : 3/7
 - ii. Paket *Studentische Uni-Mitglieder* : 0/2
 - iii. Paket *Universität Organisation* : 1/4
- **(6,5 Punkte)** Instabilität der Typen
 - i. Uni-Mitglied = 0/2 = 0
 - Efferente: -
 - Afferente: Mitarbeiter, Student
 - ii. Mitarbeiter = 1/(1+2) = 1/3
 - Efferente: Uni-Mitglieder
 - Afferente: Dozent, Hilfskraft

iii. Dozent $= 3/(3+4) = 3/7$

- Efferente: Mitarbeiter, Student, Fachzeitschrift
- Afferente: Student, Hilfskraft, Professor, Doktor

iv. Hilfskraft $= 2/(2+2) = 1/2$

- Efferente: Mitarbeiter, Dozent
- Afferente: WHK, Tutor

v. Professor $= 2/(2+2) = 1/2$

- Efferente: Dozent, Student
- Afferente: Student, Fachzeitschrift

vi. Doktor $= 2/(2+0) = 1$

- Efferente: Dozent, Student
- Afferente: -

vii. WHK $= 1/(1+0) = 1$

- Efferente: Hilfskraft
- Afferente: -

viii. Student $= 5/(5+6) = 5/11$

- Efferente: Uni-Mitglied, Dozent, Professor, Fachschaft, Lernbetreuung
- Afferente: Tutor, Lernbetreuung, Fachschaft, Professor, Doktor, Dozent

ix. Tutor $= 2 / (2+0) = 1$

- Efferente: Hilfskraft, Student
- Afferente: -

x. Organisation $= 1 / (1+2) = 1/3$

- Efferente: String (!)
- Afferente: Fachschaft, Fachzeitschrift

xi. Fachschaft $= 3 / (3+2) = 3/5$

- Efferente: Organisation, Lernbetreuung, Student
- Afferente: Lernbetreuung, Student

xii. Fachzeitschrift $= 2 / (2+1) = 2/3$

- Efferente: Organisation, Professor
- Afferente: Professor

xiii. Lernbetreuung $= 3 / (3+2) = 3/5$

- Efferente: Fachschaft, Student, Hilfskraft
- Afferente: Fachschaft, Student

- b) **(3,5 Punkte)** Klassifizieren sie ausgehend von ihren Ergebnissen die Typen im Paket „Nicht-Studentische Uni-Mitglieder“ bezüglich ihrer Wiederverwendbarkeit. Dazu sollten die in der Vorlesung behandelten Kategorien *gut wiederverwendbar*, *schlecht wiederverwendbar*, *nutzlos* und *unwartbar* verwendet werden.



- c) **(3 Punkte)** Betrachten Sie nun wieder das Gesamtdiagramm. Welche Entwurfsprinzipien (von Folie 5-68) sehen Sie als erfüllt oder verletzt an? Begründen Sie jeweils ihre Antwort.

- *Dependency Inversion Principle*: Verletzt, da:
 - Abstrakte Klasse Dozent abhängig von konkretem Mitarbeiter, Student, Fachzeitschrift
 - Abstrakte Klasse Hilfskraft abhängig von konkretem Mitarbeiter
- *Stable Dependencies Principle*: Verletzt, da:
 - Dozent abhängig von instabilem Fachzeitschrift, Student
 - Student abhängig von instabiler Fachschaft

- *Stable Abstractions Principle: (bei typweiser Betrachtung): Verletzt, da:*
 - Mitarbeiter, Hilfskraft, Student, Professor, Dozent keine Proportionalität zwischen Abstraktheit und Stabilität
- *Acyclic Dependencies Principle: Verletzt, da:*
 - Lernbetreuung und Fachschaft (und Student) zyklisch voneinander abhängig
 - Student und Dozent zyklisch voneinander abhängig
 - Student und Professor zyklisch voneinander abhängig

d) **(2 Punkte)** Haben Sie gegebenenfalls Vorschläge, wie das Design verändert werden kann um auch die verletzten Prinzipien zu erfüllen?

(Beispiele)

- DIP: Die Klasse Mitarbeiter kann als abstrakte Klasse deklariert werden, somit hat die Klasse Hilfskraft keine Abhängigkeit von einer konkreteren Klasse
- SDP: Die Assoziation zwischen Student und Fachschaft müsste entfallen. Gegebenenfalls kann man auch die Stabilität der Fachschaft erhöhen (durch mehr afferente Abhängigkeiten/weniger efferente) oder die Stabilität der Klasse Student verringern (etwa durch mehr efferente Abhängigkeiten)
- SAP: Die Stabilität müsste durch weitere (bzw. entfallende) Abhängigkeiten gemäß der Abstraktheit angepasst werden
- ADP: Die Klasse Lernbetreuung erbt nicht mehr von Fachschaft, somit gibt es keine Kreisabhängigkeit zwischen der Fachschaft und der Lernbetreuung mehr

