

**Ohne Gewähr!****Aufgabe E-1:**

In UML-Klassendiagrammen sind Mehrfachvererbungen erlaubt  
 Das statische Modell in der Analyse unterscheidet sich meistens vom späteren statischen Modell im Entwurf  
 UML ist ein Modellierungsprozess, bei dem die Erstellung von Diagrammen im Vordergrund steht  
 Nicht-funktionale Anforderungen werden in Anwendungsfällen nicht optimal repräsentiert  
 Die Signatur einer Methode ist ein Teil ihrer Schnittstellenbeschreibung  
 Klient-Dienstgeber-Architekturen (client server architectures) sind immer auch Schichtenarchitekturen  
 Der Programmierer eines Moduls eines Softwareprodukts ist ein Beteiligter (stakeholder) des Produkts  
 Anwendungsfälle lassen sich vollständig mittels UML-Anwendungsfalldiagrammen beschreiben  
 Extreme Programming ist ein Beispiel für ein inkrementelles Prozessmodell  
 Sequenzdiagramme in UML zeigen Interaktionen zwischen Klassen  
 Entwurfsmuster werden in der Regel als Klassenbibliotheken ausgeliefert  
 Ein wasserfallartiger Entwicklungsprozess ist gut geeignet für hardwarenahe Systeme  
 Kohäsion und Kopplung eines Moduls verhalten sich reziprok zueinander  
 Ein Prozessmodell ist eine Abstraktion von Prozessen  
 Bei agilen Softwareprozessen werden Bereichsklassen und Lösungsklassen in der Regel nicht auseinander gehalten  
 Durchsichten (reviews) können zum Beispiel mittels Black-Box- oder mittels White-Box-Techniken durchgeführt werden

r	f
X	
X	
	X
X	
X	
	X
X	
	X
	X
	X
X	
	X
X	
	X
	X

**Aufgabe E-2:**

Finden Sie zunächst Akteure, d.h. alles, was mit dem System interagiert, aber selbst nicht Teil des Systems ist. Identifizieren Sie danach für jeden Akteur, was er mit dem System tun will und bilden Sie daraus einen Anwendungsfall. Jede genannte funktionale Anforderung muss damit abgedeckt sein. Schreiben Sie für jeden Anwendungsfall ein Szenario bestehend aus Vorbedingung, einer nummerierten Abfolge der Tätigkeiten und einer Nachbedingung. Ein solches Szenario können Sie auch grafisch in Form eines Sequenzdiagramms darstellen. Daraus ergeben die Objekte, die interagieren. Definieren Sie zu den so ermittelten Objekten allgemeinere Klassen. In der verbalen Anforderungsbeschreibung sind dies in der Regel Satzglieder vom Typ Substantiv. Fügen Sie dies grafisch zu einem Klassendiagramm zusammen. Modellieren Sie auch Assoziationen zwischen den Klassen. Ergänzen Sie zusätzlich die Operationen, die sich aus den Szenarien ergeben. Wenn möglich, beschreiben Sie danach grafisch in einem Zustandsdiagramm den dazugehörigen Lebenszyklus.

**Aufgabe E-3:**

1.

- \* "Anmeldung vor neues Thema anlegen" versus "Jeder muss Möglichkeit erhalten .. ein neues Thema zu erzeugen"
- \* "Darf neuen Beitrag oder neues Thema hinzufügen, sonst nichts" versus "Auch Löschen möglich".

2. Thema erzeugen, Beitrag hinzufügen, Anmelden, Alte Version Wiederherstellen, Löschen.

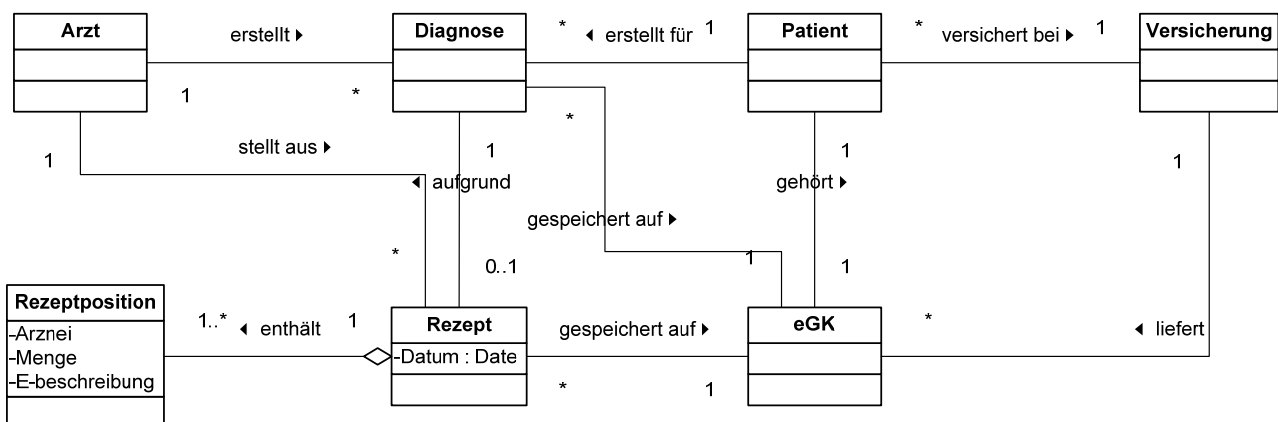
3. "leicht verständlich", "unkomplizierte Textsyntax", „Versionsverwaltung benutzen“

4.

Thema K  
Name A  
Version K (A ist auch denkbar)  
Leser S  
Versionsverwaltung E  
Textsyntax S

#### Aufgabe E-4:

1.



2.

Q nein  
S ja (0..1 Rezept aufgrund von 1..2 Diagnosen)  
V ja (\* Patient versichert bei 1..\* Versicherung)

#### Aufgabe E-5:

1. Am 26. April um 14:00 Uhr spricht Prof. Dr. Günther von Lojewski, Leiter des Journalisten-Kollegs der Freien Universität und Bruder des bekannten Fernsehjournalisten Wolf von Lojewski über „Die Rolle des Journalismus in jungen Demokratien“.

Diese Auftaktveranstaltung findet im Hörsaal 1a in der Habelschwerdter Allee 45 (Silberlaube) statt und gehört zu der Ringvorlesung „Journalismus: Die vierte Gewalt“, die während der Vorlesungszeit des Sommersemesters (also bis 15. Juli) beinahe wöchentlich am selben Ort zur selben Zeit mit zahlreichen höchst aktuellen Themen stattfinden wird.

2 od. 3

2. 2 Events und 1 Participant

#### Aufgabe E-6:

1. Die Knöpfe in den Ebenen und die Knöpfe im Fahrstuhl sind paarweise identisch, denn es ist egal, ob jemand von innen in eine Ebene möchte oder jemand auf einer Ebene den Fahrstuhl möchte.
2. Modellierung Z: 16 - Vier Zustandsklassen aufgrund Ebene/Richtung (Ebene 2 als „2 nach oben“ und „2 nach unten“) mit je vier Knopfzuständen. Die Knöpfe der „eigenen“ Ebene sind nicht von Belang.
3. Modellierung E: 4 - Aus Z fallen die vier Zustandsklassen aufgrund Ebene/Richtung weg, da sie nun in den Eingaben kodiert sind.
4. Zustandsdiagramme „explodieren“ hier mit steigendem N, daher eher ungeeignet. Eine Spezifikation müsste das Regelmäßige und Wiederkehrende der einzelnen Stockwerke nutzen, denn jedes neue Stockwerk verbindet sich auf immer die gleiche Weise mit den schon vorhandenen. (Modularität, Parametrisierung)

#### Aufgabe E-7:

##### 1. Kommando-Muster. Vorteile:

Beeinflussung der Kommandos ist zentral möglich (enable/disable, new()).  
Entkopplung von Befehlsgeber (Menü / Knopf) und Funktionalität.

##### 2. Kompositum-Muster. Vorteile:

Jedes Teil lässt sich über dieselbe Schnittstelle bedienen (paint()).  
Man braucht nicht zu wissen, dass eigentlich viele Objekte betroffen sind.  
Komposita können beliebig tief geschachtelt werden.

##### 3.

```
class GUI-Container implements GUI-Element {
    public Vector<GUI-Element> elements = new Vector<GUI-Element>();
    public void paint(){
        for (GUI-Element e : elements)
            e.paint();
    }
    // möglich, aber nicht nötig: add und remove für Elemente
}
```

#### Aufgabe E-8:

1.a. context Konto::einzahlen(betrag:double)  
post: kontostand = kontostand@pre + betrag

1.b. context Konto::auszahlen(betrag:double)  
pre: kontostand + dispolimit >= betrag

1.c. context Person  
inv: beschäftigungsstelle->notEmpty() implies  
bankverbindung->exists(b | b.dispolimit >= 500.0)

1.d. context Filiale  
inv: kunden->forall(k | k.bankverbindung->size > 0)

*Achtung: Da steht nirgends, dass es ein Konto der Filiale sein muss! Das wäre dann ...k.bankverbindung->exists(b | b.filiale = self))*

1.e. context Filiale  
inv: mitarbeiter->forall(m | self.kunden->exists(k | k = m))  
Oder: ... self.kunden->includes(m)

1.f. context Filiale  
inv: konten->forall(k | self.konten->forall(k2 |  
(k.blz = k2.blz) and (k.kontoNr = k2.kontoNr implies k = k2)))

2. Eine Person hat Konto bei genau einer Filiale, aber beliebig viele (also 0 möglich als auch bei verschiedenen anderen Filialen) Bankverbindungen.  
Vermutlich soll Person::filiale die Multiplizität 0..\* haben.

#### Aufgabe E-9:

1. Testfall-Notation: (Benutzername, Passwort, Ergebnis) mit  
Benutzername aus {Be, Bn}

Be = existierender Benutzername

Bn = nicht existierender Benutzername

Passwort aus {Pr, Pf, Pb}

Pr = zu Be passendes Passwort

Pf = zu Be nicht passendes Passwort

Pb = beliebiges, existierendes Passwort ungleich Pr

Ergebnis aus {Zulassung, Ablehnung}

2.

A (Be, Pr, Zulassung)

B (Be, Pf, Ablehnung)

C (Bn, Pb, Ablehnung)

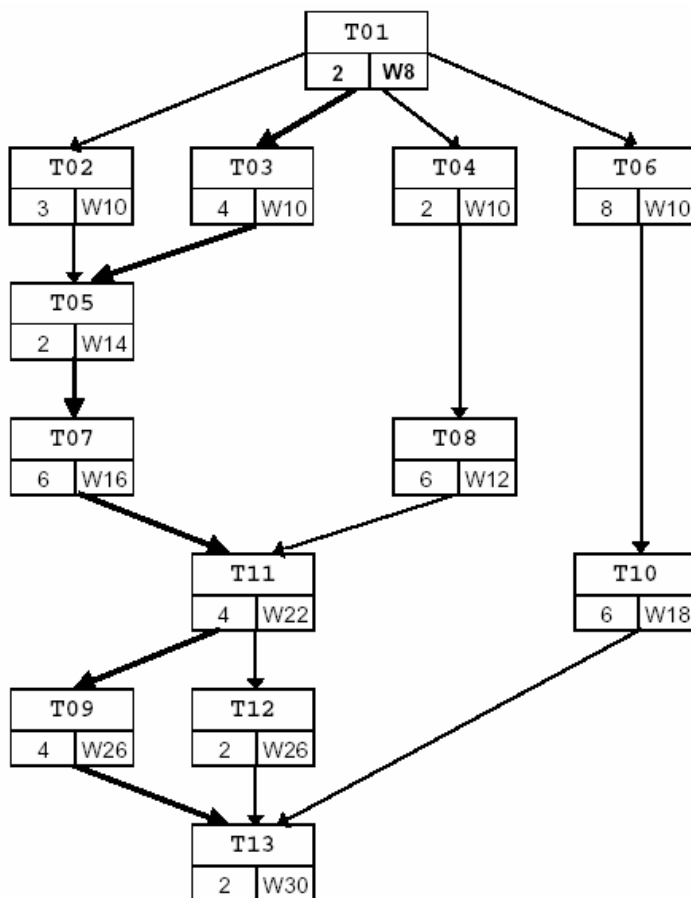
D (Be, Pb, Ablehnung)

Man könnte auch auf leere Benutzernamen und Passwörter testen.

3. A und B

#### Aufgabe E-10:

1.



2. 6 Wochen Pufferzeit haben die Aufgaben T06 und T10

3. T07

#### **Aufgabe E-11:**

1. Wenn man etwas in Auftrag gibt, wird man niemals unter einem Arbeitstag abgerechnet bekommen, also 8 Stunden, jede kostet etwa 50 Euro, ergibt 400 Euro. Der eigentliche Programmieraufwand (inkl. Qualitätskontrolle!) wird vielleicht bei 4 oder 5 Stunden liegen. Es bedarf aber einiges an Vor- und Nachaufwand. Manuell: Die Datei hat 400 Datensätze mit je 20 Ziffern, also 8000 Zeichen. Mit etwas Übung schafft man 100 Zeichen pro Minute, also dauert es 80 Minuten = 1,3 Stunden. Eine Kontrolle wird doppelt so viel Zeit kosten, also 3 Stunden. Die Qualifikation für eine solche Arbeit ist deutlich geringer und auch im Haus erledigbar, sagen wir 40 Euro, ergibt 120 Euro, also weniger als 20% der Kosten.
2. Wenn eine solche Konvertierung mehrmals nötig ist, rentiert sich eine programmierte Lösung schnell. Zudem ist anzunehmen, dass eine solche weniger unentdeckte Fehler verursacht.

#### **Aufgabe E-12:**

Hier nur ungeordnete Stichworte und Ideen: Agiler Prozess, Testautomatisierung, gute Dokumentation und guter Stil des Codes, funktionierendes Konfigurationsmanagement, dokumentierter Entwurf. Es dürften noch einige mehr vorhanden sein.