

Dies ist keine „Musterlösung“, sondern eine gute von vielen möglichen Lösungen. Kommentare, die nicht Teil der Lösung sind, sind *kursiv* gesetzt.

### Aufgabe 3-1:

*Teil dieser Aufgabe war es auch sich kurz zu fassen.*

1. Ist es möglich ein Werkzeug zu haben, was weder Schraubenzieher, Hammer noch Bohrer ist?

Ja, Werkzeug ist nicht abstrakt. Der kleine IKEA-Nimbus wäre z.B. nur ein Werkzeug.

2. Ist es in dem Klassendiagramm prinzipiell erlaubt, dass ein Bohrer auf eine Schraube anzuwenden ist?

Die statische Struktur verbietet dies nicht. Hierfür wäre OCL nötig.

3. Sind die Klassen Holzstift, Scharnier und Hammer nicht überflüssig, da sie weder Methoden noch Attribute besitzen?

- Nein, denn sie erben von ihren Oberklassen natürlich deren Operationen/Attribute.
- Auch nein, wenn man die Frage im Bezug auf „keine zusätzlichen Methoden/Attribute“ interpretiert hätte. Da 1.) ein UML-Diagramm nie eine vollständig Darstellung eines Modells sein muss und 2.) da selbst die unterschiedliche Klassenzugehörigkeit Informationen trägt. Ob es eine gute Idee ist solche Unterklassen anzulegen ist eine andere Frage.

4. Wie sieht der Pseudocode für die Methode berechneGesamtdauer() aus?

Achtung: Eine Rekursion wegen möglicher Unterschritte ist nicht nötig, da alle Schritt stets mit ihrer Anleitung verknüpft sind.

Achtung2: Ob man die geschätzte Dauer noch mit Wiederholungen multiplizieren muss, kann man ohne weitere Kenntnis der genauen Bedeutung von geschätzteDauer nicht sagen.

```
berechneGesamtdauer() {  
    result = 0;  
    foreach (Schritt s : schritte) {  
        result += s.geschätzteDauer  
    }  
    return result  
}
```

5. Kann für einen Schritt ermittelt werden, wie viele Bauelemente verbaut wurden?

Nein, denn man weiß ja nicht, wie viele der in einem Bauelement-Objekt repräsentierten wirklichen Bauelemente verbaut wurden. Wenn man z.B. das Bauelement-Objekt Brett hätte mit Attribut Anzahl = 5, welches in drei verschiedenen Schritten verbaut wird, wüsste man nicht, in welchem Schritt wie viele einzelne Bretter verbaut wurden. Hierfür müsste man die Assoziation „zu verbauen in“ um eine Assoziationsklasse erweitern, die die Anzahl der in einem Schritt verbauten Bauteile (und damit auch Bauelemente) angibt.

6. Wie kann es sein, dass jeder Schritt nur genau ein Zwischenergebnis erzeugt, obwohl ein Schritt doch auch jeweils bis zu 3 Unterschritte mit jeweils einem Zwischenergebnis haben kann?

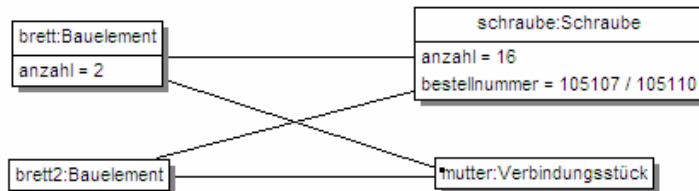
Jedes Zwischenergebnis kann aus mehreren Unterzwischenergebnissen bestehen.

7. Ist das letzte entstehende Zwischenergebnis das Möbelstück?

Nein, denn es könnte auch ein Zwischenergebnis geben, welches nicht Teil des letzten Zwischenergebnisses ist (z.B. wenn das Möbelstück zwei separate Teile hat).

8. Extra-Knifflig: Ist es möglich zu beschreiben, dass zwei Verbindungsstücke zusammen zwei Bauelemente verbinden?

Nein. Es ist nicht zu unterscheiden, ob zwei Verbindungsstücke alleine oder zusammen zwei Bauelemente verbinden. Beide male sieht das Objektdiagramm so aus:



## Aufgabe 3-2:

**1. Strukturdiagramme** stellen allgemeingültige (statische) Modelle eines Systems oder eine Momentaufnahme eines tatsächlichen Systemzustands dar. Mit ihnen lässt sich erklären, wie das System aufgebaut ist und wie es in einem Beispielzustand aussieht.

**Verhaltensdiagramme** stellen zeitliche (dynamische) Abläufe, Interaktionen zwischen Teilsystemen und Objekten, und den Lebenszyklus in den Vordergrund.

Die Grenzen zwischen den beiden Klassen ist nicht 100% scharf zu ziehen, was man sicher an den Beispielen Objektdiagramm und Anwendungsfalldiagramm deutlich merken kann. Das Objektdiagramm wird dabei eher als Strukturdiagramm und das Anwendungsfalldiagramm eher als Verhaltensdiagramm gesehen wird.

### 2.1. Welche Beschreibungselemente werden sowohl in Klassendiagrammen als auch in Sequenzdiagrammen benutzt?

Klassen (als Typangabe der Objekte, nicht als Graphik) und Operationen.

### 2.2. Was ist in der Vertikalen dargestellt?

Die Zeit oder die Lebenslinien der Objekte

### 2.3. Wie wird ein Aufruf einer Methode eines Objektes dargestellt?

Methodenaufrufe werden durch Pfeile von dem aufrufenden Objekt hin zum aufgerufenen Objekt dargestellt. Die Methode muss in der Klasse des aufgerufenen Objekts (d.h. des Objekts auf den der Pfeil zeigt) existieren.

### 2.4. Woran ist klar zu erkennen, dass Sequenzdiagramme immer nur einen von vielen möglichen Abläufen darstellen?

Das wird oft vergessen! „Oben“ stehen Objekte, keine Klassen. Da von konkreten oder vielleicht besser beispielhaften Objekten die Rede ist, kann also das Sequenzdiagramm auch nur einen beispielhaften Ablauf der Interaktion dieser Objekte von vielen möglichen darstellen. Es gibt zwar Variationsmöglichkeiten mittels opt und alt-Fragmente, aber ganz allgemein will das Sequenzdiagramm nicht sein, dann würde nämlich jede Anschauung verloren gehen. Ziel ist es häufig den Erfolgsfall (alles läuft wie erwartet) und ausgesuchte Fehlerfälle zu beschreiben.

Sequenzdiagramme sind damit also auch *nicht* geeignet, den vollen Quellcode eines Objekts zu erzeugen.

3. Siehe nächste Seite. Diese Lösung nicht eindeutig, da die statische Modellierung nur geraten werden kann und bei bekannter statischer Modellierung es natürlich viele mögliche Abläufe geben kann. In der vorliegenden Lösung kann z.B. das eRezept sich selbst verschlüsseln (enc). Das muss nicht so sein. In der Vorlesung hat Prof. Prechelt (zumindest letztes Jahr ;-)) gesagt, dass das authC2C() auch die Karte selbst (mit dem HBA) machen könnte, da sie einen Prozessor hat. Das HBA ist hier Teil des PVS geworden; es kann aber auch ein eigenständiges Objekt sein. Dann sieht's anders aus. Die Erzeugung vom Ticket und vom eRezept ist nicht mit aufgenommen, auch das könnte man machen. Ist euer Diagramm eher verständlich als die informelle Notation aus der eGK?

## Aufgabe 3-3:

Der Entwickler sagt:

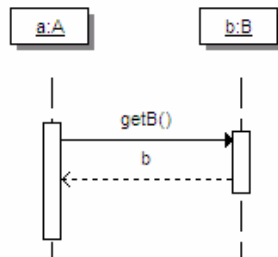
- \* Nur ich kann die speziell gebrauchte Funktionalität realisieren. Bei Fremdem weiß ich nicht, ob es wirklich das tut, was ich erwarte. („Does-Not-Apply-Here-Syndrom“)
  - \* Ich will mich nicht abhängig machen von dem Hersteller. Wenn etwas nicht funktioniert, kann ich nichts machen, bekomme aber den Ärger. Außerdem kann es sein, dass sich die Funktionalität in neuen Versionen ändert – das muss ich jedes Mal prüfen.
  - \* Ich muss erst diese andere Bibliothek finden. Das Suchen ist aber nicht so einfach. Außerdem muss man Evaluieren, welche der vielleicht vorhandenen Alternativen die geeignetste ist – das ist so „unkreativ“.
  - \* Einarbeiten in fremde Bibliotheken macht weniger Spaß als selbst programmieren. („Not-Invented-Here-Syndrom“)
- Hier könnte man gut fragen, wie diese Probleme beseitigt werden könnten.

Der Manager sagt hingegen:

- \* Eine Komponente kaufen ist häufig billiger und oft schneller als Selbstmachen.
- \* Aber Abhängigkeit ist in der Tat nicht schön, vor allem kann der Hersteller pleite gehen oder die Unterstützung für das Produkt einstellen.
- \* Wenn der fremde Hersteller einen guten Namen hat, wertet es unser Produkt auf. Und gute Geschäftsbeziehungen sind immer hilfreich.
- \* Die Wartung und Gewährleistung für dieses Modul liegt nicht mehr bei uns (Outsourcing).

Hier noch ein paar Hinweise zu Sequenzdiagrammen und typischen Fehlern, die ihr gemacht habt:

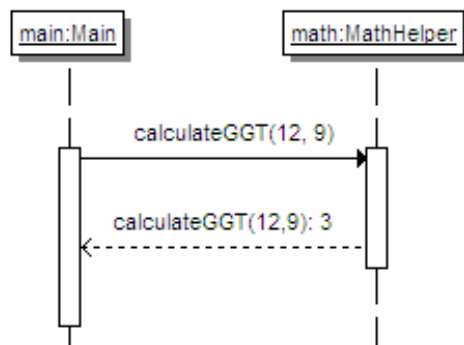
- In einem Sequenzdiagramm kann man nicht sehen, welches Objekt welches andere „kennt“ bzw. einen Verweis auf das andere Objekt hält (dies würde durch einen Link im Objektdiagramm dargestellt). Das hat euch etwas verwirrt und häufig habt gab es Aufrufe in euren Lösungen wie z.B. folgenden:



Natürlich funktioniert dies nicht, denn der Aufrufer musste das aufgerufene Objekt ja schon kennen, um diesen Aufruf überhaupt zu tätigen.

Geht lieber davon aus, dass sich Objekte im Sequenzdiagramm gegenseitig kennen (die einzige Ausnahme wäre, wenn das Sequenzdiagramm als Ziel hat zu zeigen, wie Objekte miteinander bekannt gemacht werden).

- Rückgabewerte sollten nicht auch Aussagekraft haben. Anstatt „Success“ lieber gar nichts hinschreiben.
- Viele von euch haben ein Combined-Fragment alt mit leerer unterer Hälfte verwendet. Besser ist es hierfür opt zu nehmen.
- Wir verwenden übrigens eine sehr lax Syntax für die Rückgabewerte. Richtig sollte dies nämlich so aussehen (Rückgabewert mit Doppelpunkt hinter dem Funktionsaufruf, Siehe UML SuperStructure 2.1.1 Seite 491):



# Transport eRezept über eGK

