

Hinweise: Die meisten Beispiele bestehen aus Modellierungsbeispielen und dazu passenden Theoriefragen.

- Modellierungsbeispiele: Bilden Sie den Sachverhalt, der in der Angabe geschildert wird, möglichst genau ab. Sollte etwas in der Angabe nicht erwähnt sein, treffen Sie sinnvolle Annahmen.
- Theoriefragen: Nehmen Sie sich bei der Beantwortung die Modellierungsaufgaben der jeweiligen Aufgabe zu Hilfe.

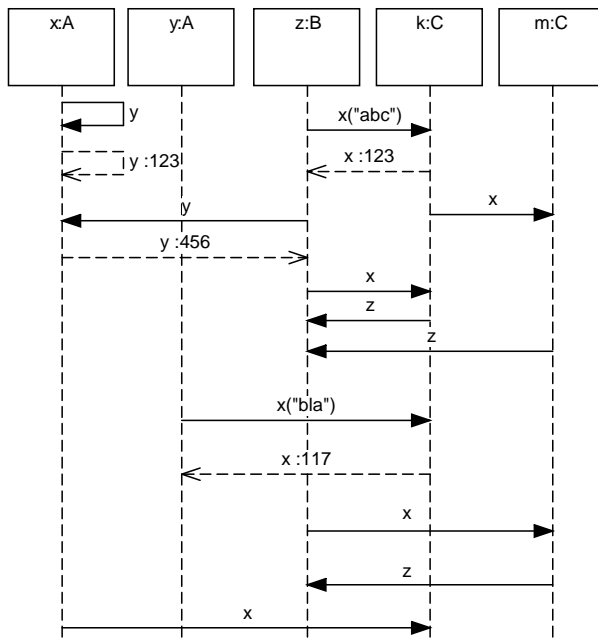
Aufgabe 1: Klassendiagramm aus Sequenzdiagramm

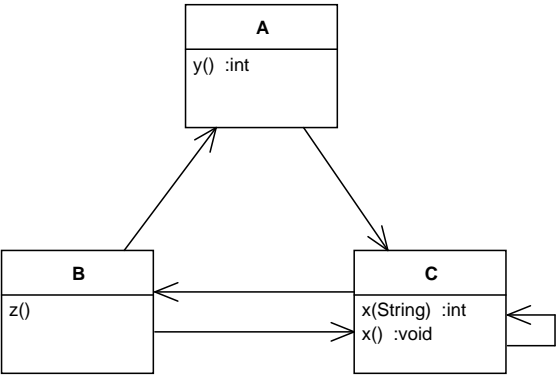
(a) Gegeben ist das nachfolgende Sequenzdiagramm. Modellieren Sie das Klassendiagramm, auf dem das gegebene Sequenzdiagramm basiert.

- Klassen
- Operationsdefinitionen mit Typangaben, soweit ersichtlich
- Beziehungen zwischen Klassen in Form von navigierbaren Assoziationen: Zeichnen Sie nur Navigationsrichtungen ein, die aus dem gegebenen Sequenzdiagramm ersichtlich sind.

Zusatzfragen:

(b) Wie ist ein Sequenzdiagramm prinzipiell aufgebaut? Welche Elemente kann es enthalten?





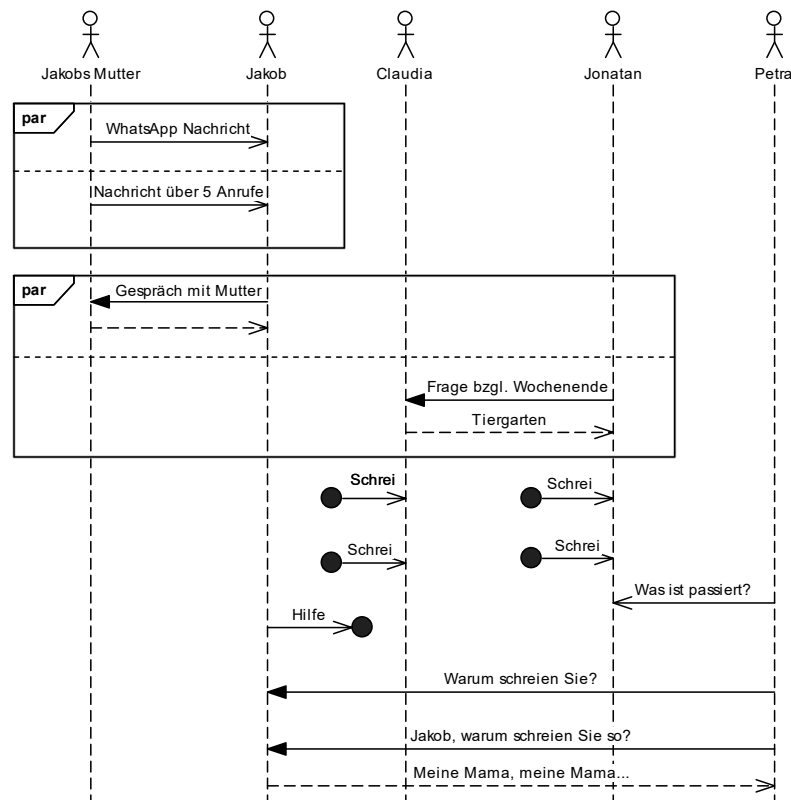
Aufgabe 2: Synchroner/Asynchroner Kommunikation

Beschreiben Sie die im folgenden Text vorkommenden Kommunikationsabläufe mittels Sequenzdiagramm. Achten Sie bei diesem Beispiel besonders darauf, ob die beschriebenen Kommunikationsabläufe synchron oder asynchron sind. Überlegen Sie außerdem, welche kombinierten Fragmente Sie hier einsetzen können.

- (Teil a) Krankenpfleger Jonatan sitzt mit seiner Kollegin Claudia im Dienstzimmer und er fragt sie, was sie letztes Wochenende unternommen hat. Claudia erzählt ihm von einem Ausflug in den Tiergarten. Plötzlich hören sie einen lauten Schrei und schrecken beide auf. Sie laufen auf den Gang und hören abermals einen Schrei. Sie können das Geräusch lokalisieren und laufen beide zu dem Zimmer, aus dem das Geräusch zu hören ist. Während sie laufen kommt auch Petra eine Ärztin hinzu und fragt Jonatan aufgeregt, was passiert ist. Ohne zu antworten stürmen die drei in das Zimmer.
- (Teil b) Dort sehen sie den Patienten Jakob, der gerade sein Telefon in der Hand hält, sie mit großen Augen anstarrt und „Hilfe“ schreit. Jonatan und Claudia versuchen den Patienten zu beruhigen und Petra fragt ihn, warum er schreit. Jakob nimmt sie gar nicht wahr und antwortet auch nicht. Petra versucht es noch einmal und sagt zu ihm: „Jakob, warum schreien Sie denn so?“ Darauf hin schaut Jakob sie mit großen Augen an und stammelt, „Meine Mama, meine Mama...“ Es stellt sich heraus, dass er von seiner Mutter am Handy am Vortag eine Nachricht über 5 versäumte Anrufe und eine WhatsApp Nachricht bekommen hat. Er hat sie heute zurückgerufen und während des Gesprächs mit ihr hat er sich so aufgeregt, dass er zu schreien und toben begonnen hat.

Zusatzfragen:

- (c) Was sind die Unterschiede zwischen synchronen und asynchronen Nachrichten?
- (d) Welche weiteren Arten von Nachrichten gibt es und wie sehen diese aus?



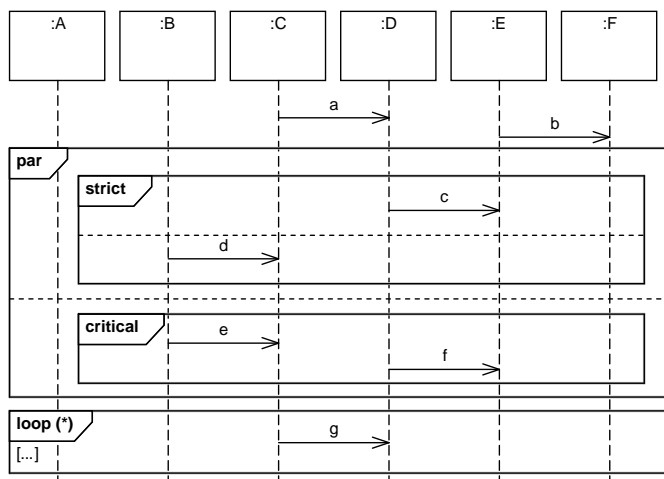
Aufgabe 3: Berechnung von Traces

- (a) Beschreiben Sie alle möglichen Ereignisfolgen des nachfolgenden Diagramms.
 (b) Welche der folgenden Traces sind möglich? Warum/warum nicht?

$a \rightarrow b \rightarrow c \rightarrow d \rightarrow e \rightarrow f \rightarrow g$	<input checked="" type="checkbox"/> ja
$c \rightarrow d \rightarrow e \rightarrow f \rightarrow g$	<input checked="" type="checkbox"/> nein
$a \rightarrow b \rightarrow c \rightarrow e \rightarrow f \rightarrow d \rightarrow g \rightarrow g$	<input checked="" type="checkbox"/> ja
$b \rightarrow a \rightarrow f \rightarrow e \rightarrow c \rightarrow d$	<input checked="" type="checkbox"/> ja
$b \rightarrow a \rightarrow d \rightarrow c \rightarrow e \rightarrow f \rightarrow g$	<input checked="" type="checkbox"/> nein
$a \rightarrow b \rightarrow c \rightarrow e \rightarrow d \rightarrow f \rightarrow g$	<input checked="" type="checkbox"/> nein
$a \rightarrow b \rightarrow c \rightarrow d \rightarrow f \rightarrow e$	<input checked="" type="checkbox"/> ja
$a \rightarrow b \rightarrow f \rightarrow e \rightarrow c \rightarrow d \rightarrow g \rightarrow g \rightarrow g$	<input checked="" type="checkbox"/> ja

Zusatzfragen:

- (c) Erklären Sie die Funktionsweise der Operatoren seq und strict.
 (d) Erklären Sie die Funktionsweise der Operatoren par und critical.
 (e) Erklären Sie die Funktionsweise des loop-Operators.



Die Nachrichten finden nach folgendem Schema statt:

- **a** und **b** in beliebiger Reihenfolge, da sie sich keine Lebenslinie teilen.
 Mögliche Sequenzen:
 - $a \rightarrow b$
 - $b \rightarrow a$
- **par**: Die Nachrichten der zwei **par**-Abschnitte können beliebig kombiniert werden, solange die folgenden Einschränkungen berücksichtigt werden:
 - Erster **par**-Abschnitt: **strict**; die Reihenfolge der Operanden muss auf jeden Fall eingehalten werden, **c** muss also vor **d** kommen.
 - Zweiter **par**-Abschnitt: **critical**; die Reihenfolge von **e** und **f** ist egal (da sie sich keine Lebenslinie teilen), dazwischen dürfen aber keine anderen Nachrichten kommen.

Mögliche Sequenzen zum Beispiel:

$c \rightarrow d \rightarrow e \rightarrow f$

$c \rightarrow f \rightarrow e \rightarrow d$

NICHT möglich zum Beispiel:

$d \rightarrow c \rightarrow e \rightarrow f$

$e \rightarrow c \rightarrow d \rightarrow f$

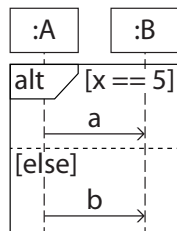
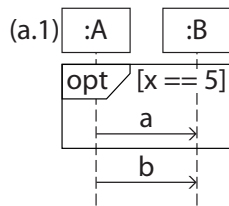
- loop: Wie oft g stattfindet ist abhängig von der Bedingung.

Aufgabe 4: Kombinierte Fragmente

- (a) Gegeben sind jeweils zwei Ausschnitte eines Sequenzdiagramms. Kreuzen Sie an, ob die beiden Ausschnitte jeweils „äquivalent“ oder „nicht äquivalent“ sind. Begründen Sie warum.

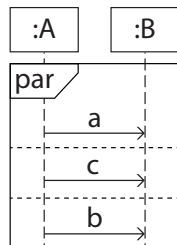
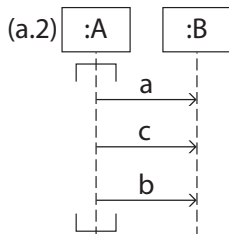
Zusatzfragen:

- (b) Erklären Sie die Funktionsweise der Operatoren alt und opt.
 (c) Erklären Sie die Funktionsweise der Coregion.
 (d) Erklären Sie die Funktionsweise des break-Operators.



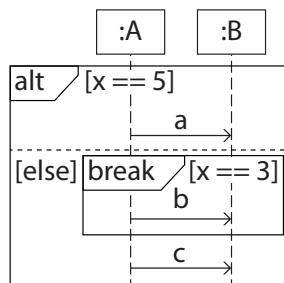
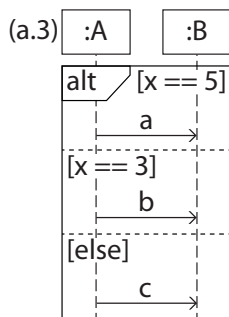
äquivalent

nicht äquivalent



äquivalent

nicht äquivalent



äquivalent

nicht äquivalent

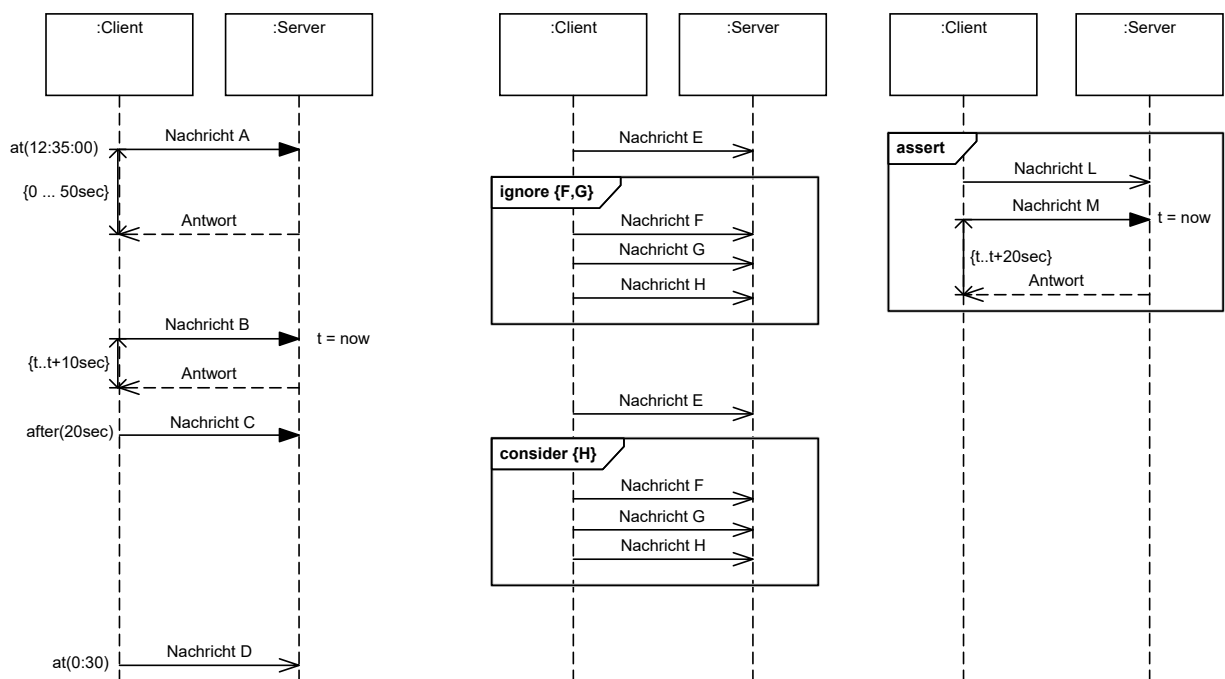
Aufgabe 5: Ausschnitte

Wie können Sie die folgenden Sachverhalte in einem Sequenzdiagramm abbilden? Modellieren Sie die geschilderten Sachverhalte.

- Der Client schickt um 12:35:00 eine Nachricht A zum Server und erwartet innerhalb von 50 Sekunden eine Antwort.
- Der Client schickt eine Nachricht B zum Server. Der Server antwortet innerhalb von 10 Sekunden. Der Client schickt 20 Sekunden nach Erhalt der Antwort eine Nachricht C.
- Der Client schickt Nachricht D um 0:30 zum Server und erwartet keine Antwort.
- Der Client schickt die asynchronen Nachrichten E, F, G und H zum Server. Die Nachrichten F und G sind für die Programmiererin nicht relevant, die Nachricht H hingegen ist sehr wichtig.
- Der Client schickt eine asynchrone Nachricht L, danach eine synchrone Nachricht M. Der Modellierer will sicher gehen, dass die Nachricht M auf jeden Fall nach Nachricht L geschickt wird und die Antwort vom Server auf die Nachricht M innerhalb von 20 Sekunden nach Erhalt der Nachricht M eintrifft.

Zusatzfragen:

- Welche Konzepte werden benötigt, um die jeweilige Aufgabenstellung zu lösen?
- Was versteht man im Kontext des Sequenzdiagramms unter Zustandsinvarianten?



Aufgabe 6: Darstellung von Programmabläufen mittels Sequenzdiagramm

Stellen Sie die Abläufe von folgendem Programm mittels Sequenzdiagramm dar. Modellieren Sie auch allfällige Antwortnachrichten.

Sie können davon ausgehen, dass alle nicht explizit deklarierten Variablen bereits deklariert und initialisiert sind. „...“ markiert vernachlässigte Codeteile, die nicht modelliert werden müssen.

```
1 class Main {
2   ScreenWindow sWindow = new ScreenWindow( );
3   FileHandler sFileHandler = new FileHandler( );
4   int nErrorCode = sFileHandler.openFile("data.txt");
5
6   if (nErrorCode == -1) {
7     sWindow.print("File does not exist!");
8   }
9   else if (nErrorCode == -2) {
10    sWindow.print("File could not be opened!");
11  }
12
13  if (nErrorCode < 0) {
14    sWindow.print("Program aborted!");
15    exit;
16  }
17
18  File sFile = sFileHandler.getFile("data.txt");
19  ...
20  boolean bManip = false;
21  int nTry = 0;
22
23  while (!bManip && nTry < 3) {
24    bManip = sFileHandler.manipulateContent(sFile);
25    nTry++;
26  }
27
28  if (!bManip) {
29    sWindow.print("File could not be manipulated!");
30    boolean nErrorCode_1 = sFileHandler.closeFile(sFile);
31    exit;
32  }
33  ...
34  boolean nErrorCode_2 = sFileHandler.closeFile(sFile);
35  ...
36 }
37
38 class FileHandler {
39
40  int openFile (String sFileName) {
41    int result;
42    ...
43    return result;
44  }
45
46  File getFile(String sFileName) {
47    File sFile = new File(sFileName);
48    ...
49    return sFile;
50  }
```

```

51
52 boolean closeFile(File sFile) {
53     boolean error;
54     sFile.close();
55     ...
56     return error;
57 }
58
59 boolean manipulateContent (File sFile) {
60     boolean error;
61     ...
62     String [] sLine = sFile.getLine();
63
64     for (int i=0; i<sLine.length;i++) {
65         if (sLine[i] == 3)
66             sFile.add("Blubs");
67         else if(sLine[i] == 5)
68             sFile.add("Blab");
69         else
70             sFile.add("BlubsBlab");
71     }
72     ...
73     return error;
74 }
75 }
76
77 class File {
78
79     void add(String text) { ... }
80
81     String [] getLine() {
82         String [] line
83         ...
84         return line;
85     }
86
87     void close() {...}
88 }
89
90 class ScreenWindow {
91     ...
92     public void print(String m) {...}
93     ...
94 }

```