

# „Konservatives“ Software-Engineering

## Aufgaben

## Aufgabe 1 (CSV-Datei)

Ein Filmverwaltungssystem importiert Daten aus gegebenen CSV-Dateien:

### Filme.csv:

```
1;Titanic;1997;Drama;194;12;(USA);(1,2,7,12,13,25);(3)
2;Catch Me If You Can;1997;Komödie;135;6;(USA,CAN);(2,5,16,17,18);(4)
3; . . .
```

### Personen.csv:

```
1;Winslet;Kate;05.10.1975;GB;(1,7,12,23,76,84)
2;DiCaprio;Leonardo;11.11.1974;USA;(1,2,22,23,50)
3;Cameron;James;16.08.1954;CAN;(1,89,103)
4;Spielberg;Steven;18.12.1946;USA;(2,34,45,56,58,59,67,87,89,102)
5;Walken;Christopher;31.03.1943;USA;(2,44,35,66,68,74)
6; . . .
```

Analysieren Sie die Dateien mithilfe einer **Data-Dictionary-Beschreibung** oder einem **Syntaxdiagramm**.

## Aufgabe 2 (Telefonnummern)

In einem Eingabefenster einer grafischen Benutzeroberfläche sollen Telefonnummern in unterschiedlichen Formaten eingegeben werden können. Diese Eingaben sind auf syntaktische Richtigkeit zu überprüfen. Hierfür ist eine Beschreibung der Syntax aller zulässigen Nummernarten erforderlich.

Verwenden Sie eine **Data-Dictionary-Beschreibung** oder ein **Syntaxdiagramm** zur einheitlichen Definition der folgenden Telefonnummern (Das Zeichen "#" stellt ein Leerzeichen dar):

07247#825743

++49#7247#825743

++49/7247/825743

0721/934567

++44/34567334556

## Aufgabe 2B

Beschreiben Sie die Struktur einer allgemeinen XML-Datei mit Hilfe von **Data Dictionary** oder **Syntaxdiagramm**.

- `<?xml version="1.0" encoding="UTF-8" standalone="yes"?>`
- `<verzeichnis>`
- `<titel>Wikipedia Staedteverzeichnis</titel>`
- `<eintrag>`
- `<stichwort>Genf Att1=XXX Att2=YYY</stichwort>`
- `<eintragstext>Genf ist der Sitz von ...</eintragstext>`
- `</eintrag>`
- `<!-- Kommentar-Text -->`
- `<eintrag>`
- `<stichwort>Bonn</stichwort>`
- `<eintragstext>Bonn ist eine Stadt, die ...</eintragstext>`
- `</eintrag>`
- `</verzeichnis>`

## Aufgabe 3 (Geometriedatei)

Durch ein Softwareprodukt sollen viele Grafikelemente importiert und exportiert werden. Dabei soll eine lesbare Textdatei erstellt bzw. gelesen werden. Für die Grafikelemente **Punkt**, **Linie** und **Kreis** sieht diese Datei folgendermaßen aus:

```
/START_DESC;  
/== Dies ist ein Kommentar ==!/   
#1 = PUNKT( 1.0, 1.0 );  
#2 = PUNKT( 2.0, 2.0 );  
#3 = LINIE( #1, #2 );  
#4 = PUNKT( 2.0, 2.0 );  
#5 = KREIS( #1, #2, #4 );  
/END_DESC;
```

Dabei stellen die mit „#“ kombinierten Zahlen Referenzen auf Elemente dar. Die Datei beginnt mit einer Startkennung und endet mit einer Ende-Kennung. Verwenden Sie eine **Data-Dictionary-Beschreibung** oder ein **Syntaxdiagramm** zur Definition dieser Datei.

## Aufgabe 3 (Geometriedatei, Forts.)

Zur Vereinfachung:

- Leerzeichen können beliebig zwischen den Elementen in der Datei stehen, sie müssen bei der Definition nicht berücksichtigt werden!
- Der Datentyp FLOAT muss nicht näher beschrieben werden.
- Die Geometrieelemente können mit Namen definiert werden ( z.B.: .. + "PUNKT" + ...)

```
/START_DESC;  
/== Dies ist ein Kommentar ==!/   
#1 = PUNKT( 1.0, 1.0 );  
#2 = PUNKT( 2.0, 2.0 );  
#3 = LINIE( #1, #2 );  
#4 = PUNKT( 2.0, 2.0 );  
#5 = KREIS( #1, #2, #4 );  
/END_DESC;
```

## Aufgabe 4 (Datenkonvertierung)

Es soll ein Datenkonvertierungsprogramm entwickelt werden, das den Inhalt einer Datei filtert und die gefilterten Daten in eine neue Datei schreibt.

### Aufbau der Eingabedatei:

FLOAT1   INT1   INT2   FLOAT2   FLOAT3   (durch je ein Leerzeichen getrennt)

### Aufbau der neuen Datei:

FLOAT2   INT1   FLOAT1

- Entwickeln Sie ein **Datenflussdiagramm** (Kontextdiagramm, D0)
- Entwerfen Sie den Vorgang der Filterung mit Hilfe von **PAP**, **Pseudocode** oder **Nassi-Shneiderman**.

## Aufgabe 5 (Ampelschaltung)

Gegeben sei folgende vereinfachte Problembeschreibung einer Ampelschaltung:

- Wenn die Ampel grün ist, dann darf man fahren.
- Wenn die Ampel rot ist, dann muss man anhalten.
- Wenn die Ampel gelb ist und der Bremsweg ausreicht, dann muss man anhalten, reicht er nicht aus, muss man fahren.

Setzen Sie diese Problembeschreibung um in

1. einen **PAP**, ein **Nassi-Shneiderman-Diagramm** oder in **Pseudocode**
2. eine **Entscheidungstabelle**



## Aufgabe 6 (Erweiterte Ampelschaltung)

Erweitern Sie Aufgabe 5 (Ampelschaltung) um das Verhalten eines Rechtsabbiegers an einer Ampelkreuzung.

Folgender Sachverhalt muss beachtet werden:

- Es kann ein Rechtsabbieger-Pfeil existieren, der an der Hauptampel installiert ist
- Es muss beim Rechtsabbiegen (bei roter Ampel) angehalten werden
- Es muss auf den von links kommenden Verkehr geachtet werden

Freiwillige Zusatzaufgabe für die Schnellen:

- Hält der Fahrer nicht an und wird erwischt, muss er Bußgeld zahlen

## **Aufgabe 7** (Klausuranmeldung)

Eine Anwendung zur Unterstützung der Klausuranmeldungen für Studenten soll erstellt werden.

Dazu sollen zunächst zur Mitte des Vorsemesters die für das nächste Semester geplanten Klausuren in das System übernommen werden. Diese Eingaben (incl. Raum der Klausur, Länge der Klausur, Klausurtag etc.) werden vom Vorsitzenden des Prüfungsausschusses durchgeführt.

Wenn alle Klausuren erfasst worden sind, erzeugt das System zwei Prüfungspläne:

1. Der Prüfungsplan für die Professoren und Mitarbeiter beinhaltet die Aufsichtsführenden der jeweiligen Klausur und den jeweiligen Erst- und Zweitprüfer. Er wird allen Professoren und Mitarbeitern zugestellt.
2. Den Prüfungsplan für die Studenten ohne diese Informationen. Dieser Prüfungsplan wird den Studenten zugestellt.

## Aufgabe 7 (Klausuranmeldung, Forts.)

Studenten können sich sodann an speziellen Terminals für ihre Klausuren anmelden.

Das System überprüft, ob die Studenten für die Klausur zugelassen sind und ob es die Klausur überhaupt gibt. Zur Überprüfung der Vorleistung wird eine bereits existierende Vorleistungsdatei genutzt.

Ist der Student für die Klausur zugelassen, erhält er eine Bestätigung und wird in die Teilnehmerdatei für die Klausur übernommen. Ansonsten erhält er eine Ablehnung mit Begründung.

Nach Ablauf des Anmeldezeitraums erhalten die jeweiligen Erstprüfer eine Liste mit den für ihre Klausur zugelassenen Studenten.

Entwickeln Sie ein **Datenflussdiagramm** für diesen Vorgang.

## **Aufgabe 8 (Messwertdatei)**

Zur Erstellung von Statistiken sollen Messwertdateien so aufbereitet werden, dass sie von einem Statistikprogramm verarbeitet werden können.

Eine Messwertdatei wird geöffnet, die Daten werden konvertiert, in eine neue Datei abgespeichert und die Dateien wieder geschlossen.

Dabei gilt:

1. die Messwerte sind eine Folge von Real-Werten
2. einzelne Texte (als Kommentare) können beliebig in der Liste stehen
3. die Messwerte sollen sortiert und ohne Texte in eine neue Datei abgespeichert werden

## Aufgabe 8 (Messwertdatei, Forts.)

Aussehen einer Messwertdatei:

```
„Kommentar 1“  
3.444  
1.23456  
99.333  
...  
2.33333  
5.5533  
„Kommentar 2“  
1.23  
2.34  
...
```

Erstellen Sie ein **Datenflussdiagramm** (Kontextdiagramm, D0 sowie D1) für dieses Konvertierungsprogramm.

## Aufgabe 9 (Fußballtipprunde)

Im Internet werden mehrere Fußballtipprunden angeboten. Dabei kann jeder Teilnehmer seine Tipps für den aktuellen Spieltag abgeben, nach dem Spieltag werden die einzelnen Tipps ausgewertet. Für diese Auswertung soll ein Programm entworfen werden. Folgende Sachverhalte sind relevant:

1. Für ein korrektes Spielergebnis gibt es 3 Punkte, für das richtige Torverhältnis (dieselbe Tordifferenz) gibt es 2 und für die richtige Tendenz (Sieg richtig) gibt es 1 Punkt (exklusive Bewertung, d.h. max. 3 P).
2. Für den **aktuellen Spieltag** wird die Tages-Gesamtpunktzahl eines Teilnehmers ermittelt.
3. Nach dem **letzten Spiel des aktuellen Spieltags** soll über alle bisherigen Spieltage ermittelt werden:
  - die bisherige Durchschnittspunktzahl
  - die Position in der Durchschnittsliste aller Teilnehmer
  - die Gesamtpunktzahl, die ein Teilnehmer bisher erreicht hat.

## Aufgabe 9 (Fußballtipprunde, Forts.)

### Zur Vereinfachung:

Es können immer alle Spieltage bis zum aktuellen Spieltag ausgewertet werden, d.h. es erfolgt keine Zwischenspeicherung

### Beispiele:

Tipp	Ergebnis	Punkte
2:1	2:1	3
2:1	1:0	2 (richtige Tordifferenz)
1:1	2:2	2 (richtige Tordifferenz)
2:1	2:0	1 (richtige Tendenz)

Beschreiben Sie diese Problembeschreibung mit Hilfe eines **PAP**, einem **Nassi-Shneiderman-Diagramm** oder **Pseudocode**

## Aufgabe 10 (Geldwechselautomat)

Ein alter, vereinfachter Geldwechselautomat gibt nach der Eingabe eines Geldbetrags Geldstücke aus nach dem *Prinzip der minimalen Anzahl bei maximaler Diversifizierung*.

Die maximal wechselbare Geldmenge liegt bei DM 10, die minimal wechselbare Geldmenge liegt bei DM 2, ansonsten wird eine Meldung angezeigt.

Dabei können Münzen im Wert von jeweils DM 1 bis DM 5 ausgegeben werden. Es wird vorausgesetzt, dass sich genügend Wechselgeld im Automaten befindet.

1. Entwerfen Sie den Vorgang der Ausgabe des Geldwechselautomats mit Hilfe von **PAP**, **Pseudocode** oder einem **Nassi-Shneiderman-Diagramm**
2. Stellen Sie diesen Sachverhalt als begrenzte und als erweiterte Entscheidungstabelle dar
3. Was muss sich ändern, wenn Sie den Automaten auf EURO umstellen wollen?



## Aufgabe 11 (Würfelspiel)

Entwerfen Sie eine Auswertungsfunktion die ein bekanntes einfaches Würfelspiel mit 5 Würfeln. Folgende Sachverhalte sind relevant:

1. Nach max. dreifachem Würfeln können 4 verschiedene bewertete Hauptresultate erzielt werden (3er-, 4er-, 5er-Pasch und Full House).
2. 3er, 4er- oder 5er-Pasch ergeben 20, 50 bzw. 100 Punkte.
3. Haben Paare von 2 **und** 3 Würfeln jeweils dieselbe Augenzahl (Full House), erhält man 60 Punkte.
4. Diese Punkte werden in eine Liste mit je einer Zeile pro Resultat eingetragen, sofern der Eintrag an der entsprechenden Zeile noch nicht vorgenommen wurde.
5. Wurde keines der oben erwähnten bewerteten 4 Hauptresultate erreicht, kommt der nächste Mitspieler an die Reihe und die gerade gewürfelten Punkte werden auf einem separaten Feld addiert.

## Aufgabe 11 (Würfelspiel, Forts.)

6. Hat ein Spieler alle 4 bewertete Hauptresultate erreicht (Liste gefüllt), werden die Punkte zusammengezählt und der Gewinner des Spiels ermittelt.

Aussehen der Ergebnisliste:

Resultat	Spieler 1	Spieler 2		Spieler N
3er-Pasch	20			20
4er-Pasch		50		50
5er-Pasch		100		
Full House				60
Sonstige	21 16 12	19 14		13
Summe				

## Aufgabe 13 (Postschalter, DFD)

Gegeben ist die folgende verbale Beschreibung eines Vorgangs an einem Postschalter (Geld von einem Postspargbuch abheben):

1. Der Kunde teilt dem Schalterbeamten mit, dass er von seinem Postspargbuch gerne Geld abheben möchte.
2. Der Schalterbeamte gibt dem Kunden von einem Stapel einen Rückzahlungsschein und bittet, diesen auszufüllen. Den ausgefüllten Rückzahlungsschein gibt der Kunde dann mit seinem Postspargbuch ab.
3. Der Kunde muss stets noch seine Postspargbuchkarte vorlegen. Beträgt die Abhebung mehr als 500,- €, dann muss er noch zusätzlich seinen Personalausweis vorlegen.
4. Der Schalterbeamte prüft alle notwendigen Unterlagen (genügend Geld auf dem Postspargbuch, stimmen Postspargbuchnummern aus Postspargbuch und auf der Ausweiskarte überein, ist der Personalausweis in Ordnung).

## Aufgabe 13 (Postschalter, DFD, Forts.)

1. Der Schalterbeamte ergänzt den Rückzahlungsschein um seine Unterschrift, den Tagesstempel und gegebenenfalls um die Nummer des Personalausweises und heftet ihn in einem Ordner ab.
2. Er trägt die Abhebung im Postsparbuch ein, gibt dem Kunden seine Unterlagen zurück und überreicht ihm das Geld, das er der Schalterkasse entnimmt.

Erstellen Sie ein **Datenflussdiagramm** (Kontextdiagramm, D0) für diesen Vorgang.