

IN: (80') Aufgaben 1-12
II, WI: (60') Aufgaben 1-9

Klausur: Softwaretechnik / SWT1

Name:

Matr.Nr:

Fachrichtung:

1	2	3	4	5	6	7	8	9	10	11	12
4	8	15	4	3	6	4	3	8	3	7	5

Punkte: von 70/55

- 1.) Warum ist die **Komplexität** (z.B. Maß des Informationsaustausches von Modulen) von Softwaresystemen nichtlinear (Brooksches Gesetz) ? (4P)
- 2.) Prüfen Sie nachfolgende **Entscheidungstabelle** auf *Redundanzen* und auf *Widersprüche* (8P) und, falls vorhanden, geben Sie diese an (beteiligte Regeln mit Aktionen). Ist die ET vollständig (wenn nicht, fehlende Bedingungsanzeigerspalten angeben) ?

	Regel 1	Regel 2	Regel 3	Regel 4	Regel 5	Regel 6
Bedingung 1	J	N	J	J	N	J
Bedingung 2	-	J	-	J	-	N
Bedingung 3	J	N	N	N	J	-
Aktion 1	*	*				
Aktion 2			*		*	*
Aktion 3				*		

- 3.) Folgende Anforderungen wurden als Ziele des SW-Systems „**Versandhaus_Helle**“ ermittelt: (15P)

Auf Anfrage schicken **Lieferanten** dem Versandhaus Angebote.
Lieferanten erhalten vom Versandhaus Lieferaufträge und liefern Warenartikel.
Kunden bestellen Artikel und erhalten die Artikellieferung mit Rechnung.
 Monatlich erhält der **Geschäftsführer** des Versandhauses eine Übersicht über Kundenbestellungen und (von Lieferanten) gelieferte Warenartikel.
Kunden und **Lieferanten** können ihre Anschrift ändern.

Stellen Sie (in dieser Reihenfolge)

Ereignistabelle, Kontextdiagramm, Datenflussdiagramm (DFD) der essentiellen Ebene (0-Ebene) auf.

Jedem Ereignis in der Tabelle soll ein Prozess im DFD entsprechen.
 Anschließend leiten Sie aus dem DFD ein **Klassendiagramm** ab (UML-Notation);
 Methoden und Attribute brauchen hierbei nicht angegeben werden.

- 4.) Was bedeutet folgender DD-Eintrag? (4P)

$$A = \{ B \} + [C | D + (E) | 2\{G\}3] + @H$$

- 5.) a) Wodurch sind die sogenannten "**äußeren**" **Qualitätsmerkmale** von Software erklärt? (3P)
 b) Nennen Sie zwei Beispiele für sich **widersprechende** Software-Qualitätsmerkmale!
- 6.) Erläutern Sie das "**Stagewise Model**" der SWE (Vorgehen, Nachteile). (6P)
- 7.) Für die folgenden beiden Projekte ist ein geeignetes **Vorgehensmodell** zu wählen und die Wahl kurz zu begründen: (4P)
- Software zur Steuerung eines neu zu entwickelnden Antiblockier-Systems für PKWs. Team 3 Mitarbeiter. ca. 3000 LoC. Lebensdauer der Software 5 Jahre.
 - Online-Banking-System einer europaweit agierenden Bank. Team 500 Mitarbeiter. ca. 500.000 LoC. 30 Jahre Lebensdauer der Software.
- 8.) Kreuzen Sie an, ob die Aussage richtig (Ja) oder falsch (Nein) ist: (3P)
 (falsches Kreuz: ¼ Pkt Abzug, Minimum 0 Pkt)

Minimale Bedingungsüberdeckung enthält Zweigüberdeckung	Ja	Nein
Das CoCoMo – Modell beruht hauptsächlich auf LoC Angaben	Ja	Nein
Qualität kann man in Software „hineintesten“	Ja	Nein
Das CMM – Modell dient der Einordnung des Softwareprozesses	Ja	Nein
Klassendiagramme dienen der statischen OO-Analyse	Ja	Nein
Warnier-Orr-Diagramme lassen sich in Struktogramme umwandeln	Ja	Nein

- 9.) Für folgende Codesequenz ist ein **Bedingungsüberdeckungstest** durchzuführen. (8P)
 Notieren Sie tabellarisch mit konkreten Wertezuweisungen die dazu notwendigen Testfälle getrennt nach a), b), c) für
- minimale** Bedingungsüberdeckung (C_2)
 - minimale-mehrfache** Bedingungsüberdeckung (C_{2-3})
 - mehrfache** Bedingungsüberdeckung (C_3)

```
IF (a = 1) OR (b = 0) THEN Anw_1 ELSE Anw_2 ENDIF
```

- 10.) Erklären Sie die Ziele von **Verifikation**, **Validation** und **Testen** bei der Softwareentwicklung. (3P)
- 11.) Ein Programm verwalte Lagerbestände. Dazu werden *Warennummern* und *Stückzahlen* festgehalten: $1 \leq \text{Stückzahl} < 1000$
Warennummer: dreistellige Ganzzahl (ohne führende Nullen!)
 Gib **Äquivalenzklassen** an und notiere anschließend mit einer minimalen Anzahl von Eingabe-Kombinationen eine **Grenzwertanalyse**. (7P)
- 12.) Formen Sie folgendes **Jackson**-Diagramm in eine Pseudocode-Notation um (5P)

