

# Exercise Sheet 8

Birgit Pohl 574353 (MO. 9-11)  
Kevin Trogant 572451 (Mo. 15-17)  
Ronja Enseleit 572404 (Mo. 15-17)  
Dustin Hebecker 571271 (MO. 9-11)



Group: \*Iron Man\*

9. Januar 2017

## Aufgabe 1

a)

#	C	E	F	D	B	A
1	C	E	F	D		
2	C	E	F	D	B	
3	C	E	F	D	B	
4	C	E	F	D	B	A
5	C	E	F	D	B	A
6	C	E	F	D	B	A

Tabelle 1: Integrierte Komponente, Platzhalter

b)

Es werden hier und in den meisten Fällen  $N - 1$  Platzhalter und ein Treiber benötigt, da bei der Top-Down-Strategie höher liegende Komponenten zuerst abgearbeitet werden. Gibt es mehr als eine Top-Level-Komponente oder loops, kann es Abweichungen von diesen Regeln geben. Also in diesem Fall 6 Platzhalter und ein Treiber, wobei die Platzhalter 11 mal Verwendung finden.

c)

Alle Komponenten können in beliebiger Reihenfolge integriert werden vorausgesetzt, dass mindestens eine Komponente die Oberhalb liegt bereits getestet wurde, woraus sich verschiedene Varianten ergeben. Im Beispiel hier können z.B. E, F und D beliebig in der Reihenfolge vertauscht werden. Eine optimale Variante bindet D als letztes ein um die Anzahl der zu verwendenden Platzhalter zu jedem Zeitpunkt so gering wie möglich zu halten. Also in diesem Fall den Platzhalter von A so spät wie möglich zu verwenden.

## Aufgabe 2

Bei der Top-Down-Strategie wird der Aufbau des Programms früh sichtbar und die Steuerfunktionalität früh geprüft. Dies geht zu Last der Prüfung der Integration mit dem System und der Hardware. Es werden viele Platzhalter verwendet bei mit der Zeit steigendem Aufwand. Bei der Bottom-Up-Strategie steigt der Aufwand mit der Zeit und anstelle von Platzhaltern werden viele Treiber benötigt. Integration mit System und Hardware werden früh geprüft, aber heit zu Last der Steuerfunktionalität und der Vorzeigbarkeit. Die Outside-In-Integration verbindet beide vorgehensweisen. der Zeitaufwand ist relativ constant und es werden die Vorteile aus Top-Down- und Bottom-Up-Integration genutzt. Jedoch werden nun Treiber und Platzhalter benötigt. Als default Strategie sollte die Outside-In-Integration verwendet werden, welche im Schnitt die meisten Vorteile mit sich bringt. Es sollte dennoch von Fall zu Fall entschieden werden ob nicht eine der anderen Strategien erfolgreicher ist. Werden z.B. viele Informationen nach unten transportiert ist es vermutlich einfacher mit einer Top-Bottom-Strategie zu Arbeiten, da die Implementierung der Dummies hier weniger Aufwand macht als die der Treiber. Werden auf der anderen Seite viele Informationen nach oben weiter gegeben tritt der umgekehrte Fall ein und eine Bottom-Up-Strategie ist effektiver. Bei einem Gleichstand kann im Allgemeinen angenommen werden, dass Treiber einfacher zu Implementieren sind als Platzhalter.