



# Software-Engineering für langlebige Systeme

## PA

- HA2
- Gelerntes aus dem Rechner
- Transitionssysteme

## HA - Bemerkungen

- Bearbeitungspunkt Aufgabe
  - Wer nur die Hälfte der Aufgabe bearbeitet kann nur die Hälfte der Punkte erwarten
- Heimübungslösungen sollen strukturiert sein

## HA 2

- Ziel:
  - Erkennen von Problemen im Code
  - Wartung
- Korrektur:
  - Vergleich mit meiner Kopie von HA1-Wertung
  - Ansicht des Codes

## HA2

- Ziel:
  - Erleben der Softwareerosion am eigenem Leib
  - Code kritisch lesen und sehen was falsch läuft

## Transitionssysteme / Vor- und Nachbedingung/ Graphtransformationen

- Erklären Sie kurz die Datenstruktur
- Modellieren Sie die Datenstruktur

	Transitionssystem	Vor – und Nachbedingung	Graphtransformation
Stack	Gruppe 1	Gruppe 2	Gruppe 3
Queue (evtl. beschränkt: 3 Elemente)	Gruppe 4	Gruppe 5	Gruppe 6

## Hello World

Schreiben Sie ein C-Programm welches die Ausgabe:

Hello World

erzeugt.

## Strings und Zeiger

- Erweitere das Hallo-World-Programm um einen Parameter, der einen Namen erwartet und diesen grüßt:
- 
- `> helloWorld Thomas`
- `Hello Thomas`
- `> helloWorld Jan`
- `Hello Jan`



## Mehrere Namen

> helloWorld Thomas

Hello Thomas

> helloWorld Jan Thomas

Hello Jan

Hello Thomas

>

## Verkettete Liste (Queue)

- Definieren Sie die Struktur für eine verkettete Liste
- Erstellen Sie eine Operation zum Hinzufügen eines Elementes am Anfang der Liste
- Erstellen Sie eine Operation zum Entfernen des letzten Elementes