

## Programmieren I Zusammenfassung

**Wichtig:** diese Zusammenfassung ist entworfen, um die wichtigen, sich wiederholenden Ideen in der Klausur zu zeigen, es gibt keine Garantie, dass die Fragen nur von diesen Ideen kommen müssen!

Klausur Fragen lassen sich am meistens in einer der folgenden Gruppen verteilen:

- 1) Stringbearbeitung, hier sollte man ziemlich viele Aufgaben erwarten (zu einem anderen String kopieren, chars vergleichen, search and replace, nach einem Teilstring suchen, String mit anderem String an einer Stelle erweitern, Verschlüsselung/ Entschlüsselung (am meistens mit Caesar-Verschlüsselung),...usw.) (Beispiel WS2014-erste Gruppe):  
"Schreiben Sie eine Funktion, die überprüft, ob in einem String 4 aufeinanderfolgende aufsteigende Zahlen auftreten.  
Beispielsweise "as53679e"."
- 2) **Bäumebearbeiten** (sehr wichtig, kommt jedes Jahr!), gutes Beispiel dazu ist (WS2014-zweite Gruppe)  
"Aufgabe 2 (Java): Schreiben Sie eine Methode, die zurückgibt, wie häufig es Child- und Parent-Nodes gibt, bei denen sich das Vorzeichen unterscheidet."
- 3) Liste- oder Arraybearbeitung: am meisten kommt fast immer **sortieren** vor, Elementweisebearbeitung ist auch sehr wichtig (2 Listen, man muss sie z.B. miteinander Element für Element vergleichen dann eine neue Liste zurückbringen, die das größte Element an jeder Stelle enthält), auch wäre es nützlich, wenn man eine einfache Suchfunktion im Kopf hat (nur eine for-Schleife mit if bedingung da drin, kein Fancy stuff !)  
gutes Beispiel dazu ist (WS2014-2Gruppe): "Schreiben Sie eine Funktion, die das mittlere Element einer verketteten Liste zurückgibt. Wenn die Liste eine gerade Anzahl an Elementen hat, soll eine Null zurückgegeben werden."
- 4) Pointers und Speicher Verwaltung: nicht so schwierige Sachen, wenn überhaupt dann nur basisnahe Fragen werde ich erwarten.
- 5) Nur um meine Liste vollständig zu machen, werde ich auch hier Funktionen, numerische Bearbeitung (Decimal zu Hex\Binär\Octal zu verändern, GGT (Größte gemeinsamer Teiler), KGV (Kleinstes gemeinsames Vielfach)..usw.)

Klausur Fragen sind meistens einfach zu beantworten, wenn man die geeignete Tools dafür schon im Kopf hat, Folgendes sind wichtige Tools zum Benutzen.

- 1) Im Anhang steht eine ziemlich einfache Baumklasse (Tree.java), man kann es einfach ein paar mal lesen vor der Klausur (auch ein paar Zeilen Code auswendig lernen schadet nicht ;) (wenn man natuerlich versteht, wie es laeuft) )
- 2) Man soll eine Suchfunktion immer im Kopf haben, es ist sehr einfach, einmal durch die gesamte Struktur durchgehen mit eine for- oder Whileschleife (kann ein bisschen Tricky mit nicht linearen Datenstrukturen wie Bäumen sein!) dann mit einer if-Bedingung ueberpruefen, ob dieses Element den gesuchten Wert hat, wenn es keine Elemente mehr gibt, return -1 oder irgendein Wert der zeigt, dass dieses Element nicht gefunden ist.

- 3) Für Listen-/ Arraysortieren Aufgaben, gibt es ziemlich viele Algorithmen zu bemerken, man kann auch irgendeinen Algorithmus lernen und für immer benutzen (Empfehlung Bubblesort, man findet es im Anhang unter Bubble.java, kann einfach uebersetzt werden fuer C, und **Listen**).
- 4) Stringsbearbeiten kann sehr einfach sein, wenn man die Stringfunktionen in **C** und Java lernt, hier sind ein paar persoenliche Empfehlungen (ich habe keine Ahnung von der neuen Hilfe-Bibliothek, es kann hilfreich sein, einen vor der Klausur eine Blick auf sie zu werfen).

<code>strcpy(s1, s2)</code>	Copies string s2 into string s1
<code>strcat(s1, s2)</code>	Concatenates string s2 onto the end of string s1 (Erweiterung)
<code>strlen(s1)</code>	Returns the length of string s1 (sehr nuetzlich fuer Schleifen, besonders fuer for-Schleifen)
<code>strcmp(s1, s2)</code>	Returns 0 if s1 and s2 are the same; less than 0 if s1<s2; greater than 0 if s1>s2
<code>strchr(s1, ch)</code>	Returns a pointer to the first occurrence of character ch in string s1
<code>strstr(s1, s2)</code>	Returns a pointer to the first occurrence of string s2 in string s1

- 5) Andere Bemerkungen:
  - a. Wenn man ein periodisches Verhältnis beobachten kann, ist es nützlich modulo operator zu benutzen(%), Beispiel ist 0,1,2,3,0,1,2,3,0,.. == x%4 (wobei x ein Variable in einer Schleife ist)
  - b. Wenn man überprüfen will, ob 2 Zahlen die gleichen Vorzeichen haben, muss man nur Produkt davon überprüfen, wenn es positiv ist, haben die beiden Zahlen gleiche Vorzeichen, wenn es negativ ist, dann sind sie verschieden (no need to 4-if-else checks)

Viel Erfolg euch allen!

**Written by: Mido Mughrabi**, for the Students in the Leibniz Universitaet Hannover.  
Third-party materials are owned by their Original owners.