

## KGÜ 1 Einführung in die Sprache C

**Inhalte:** Wiederholung der Programmiersprache C, Funktionen der Standardbibliothek, Datenstrukturen, Felder, Debuggen

**Lernziele:** Grundzüge von C, Zweck und Anwendung verschiedener Funktionen der Standardbibliothek, Speicherverwaltung, Anwenden eines Debuggers

### Aufgabe 1: Verwendung von Datenstrukturen

Die Aufgaben 1.a) bis 1.c) beschäftigen sich mit Datensätzen (structs), den verschiedenen Datenspeicherbereichen und mit speziellen Funktionen, die in späteren Übungen immer wieder genutzt werden. Die grundlegende Benutzung des Debuggers *Totalview* wird im RWTH Cluster Howto aus dem Lernraum erläutert.

- a) Definieren Sie eine Datenstruktur „student“ und legen Sie einige dieser Datenstrukturen an, wobei Sie lokale, globale und Heap-Variablen nutzen. Dabei soll ein Student charakterisiert sein durch seinen Nachnamen (max. 30 Zeichen), seinen Vornamen (max. 20 Zeichen) und seine Matrikelnummer (als Ganzzahl). Dies lässt sich in eine Datenstruktur (hier ein *struct*) abbilden.
- b) Schreiben Sie eine Funktion, welche die Daten eines Studenten ausgibt. Testen Sie diese Funktion, indem Sie die Daten der oben definierten Studenten damit ausgeben lassen.
- c) Schreiben Sie eine Funktion, welche die Daten eines Studenten vom Nutzer erfragt und in eine neu anzulegenden Datenstruktur „student“ ablegt. Testen Sie diese Funktion, indem Sie ein Testprogramm (*main*) schreiben, das einen Studenten einliest und ihn zur Kontrolle wieder ausgibt.

### Aufgabe 2: Die Verwendung von mehrdimensionalen Feldern

Die Aufgaben 2.a) bis 2.b) schließlich beschäftigen sich mit verschiedenen Datenspeichermodellen für mehrdimensionale Felder und wieder mit den verschiedenen Datenspeicherbereichen.

- a) Gegeben sei das folgende C-Programm (s. Arrays1.c):

```

#include <stdio.h>

char global[][5] = {"Hans", "Tim", "Pit"};

int main(int argc, char* argv[])
{
    int i = 0;
    char lokal[][8] = {"Lena", "Ina", "Iris"};
    char meinName[10] = "Andreas";
    char* ptr = "Andreas";

    if (meinName == ptr)
        printf("1. Vgl. stimmt!\n");
    else
        printf("1. Vgl. fehlgeschlagen!\n");

    if (*meinName == *ptr)
        printf("2. Vgl. stimmt! - Nur was wird hier verglichen?\n");
    else
        printf("2. Vgl. ebenfalls fehlgeschlagen!\n");

    return 0;
}

```

1. Zeichnen Sie ein Speicherabbild nachdem alle Variabledefinitionen erfolgt sind.
  2. Welche Ausgaben erzeugt das Programm und warum?
  3. Überprüfen Sie Ihre Lösungen mit Hilfe des Debuggers. Bestimmen Sie hierfür mit Hilfe des Debuggers die Werte der folgenden Ausdrücke:
    - &ptr
    - &i
- b) Nun sollen die Vornamen beim Start des Programms als Argument übergeben werden. Lassen Sie alle Vornamen auf dem Bildschirm ausgeben. Zeichnen Sie ein Speicherabbild für den Programmaufruf `./start Andreas Pit Franz.` eingeben.