

# OPR Praktikum

## Übung: Grundlagen

### Motivation

Das OPR-Fach setzt Kenntnisse in C voraus. Mit dieser Übung sollen einige Grundlagen wiederholt werden, die Ihnen die Arbeit mit C++ erleichtern:

- Verständnis des Übersetzungsvorgangs in C und C++
- Benutzung der Kommandozeile
- Zeiger

### Voraussetzungen

Informatik 1 und Informatik 2

### Vorbereitung

1. Stellen Sie sicher, dass Sie den C Übersetzungsvorgang verstanden haben!
  - Was ist der Präprozessor?
  - Was ist der Compiler?
  - Was ist der Assembler?
  - Was ist der Linker?
2. Stellen Sie sicher, dass Sie auf den Rechnern im Praktikumsraum oder auf Ihrem mitgebrachten Rechner sowohl auf der Kommandozeile als auch in einer integrierten Entwicklungsumgebung übersetzen können.
3. Stellen Sie sicher, dass Sie auf den Rechnern im Praktikumsraum oder auf Ihrem mitgebrachten Rechner mit einem Debugger arbeiten können.

### Fehler

#### Präprozessor

- Erstellen Sie ein Programm, bei dem der Präprozessor auf einen Fehler stößt.
- Erklären Sie den Fehler.
- Beheben Sie den Fehler.

#### Compiler

- Erstellen Sie ein Programm, bei dem der Compiler auf einen Fehler stößt.
- Erklären Sie den Fehler.
- Beheben Sie den Fehler.

**Assembler:** Stellen Sie durch Setzen eines geeigneten Compiler-Schalters den vom Compiler erzeugten Assembler-Code dar.

#### Linker

- Erstellen Sie ein Programm, bei dem der Linker auf einen Fehler stößt.
- Erklären Sie den Fehler.
- Beheben Sie den Fehler.

#### Laufzeit

- Erstellen Sie ein Programm, das zu einem Laufzeitfehler führt.
- Erklären Sie den Fehler.
- Beheben Sie den Fehler.

# Aufgabe: Zeiger und Debugger

Das Programm in matrix.cpp ist gegeben. Bei Ausführung führt dieses Programm zu folgender Ausgabe:

```
11 12 13
21 22 23
```

Erweiterung: Es soll eine **3x3 Matrix** ausgegeben werden. Eine Ausführung des Programms mit folgenden Parameterkombinationen nx, ny, nz muss zu folgender Ausgabe führen:

Die Parameterkombination nx = 3, ny = 3, nz = 3 führt zu nachfolgender Ausgabe:

```
111 112 113
121 122 123
131 132 133
211 212 213
221 222 223
231 232 233
311 312 313
321 322 323
331 332 333
```

Die Parameterkombination nx = 1, ny = 2, nz = 3 führt zu nachfolgender Ausgabe:

```
111
121
211
221
311
321
```

2. Kapseln Sie die Anteile des Programms in die folgenden drei Funktionen:

- int\*\*\* build\_matrix(int nx, int ny, int nz)
- void print\_matrix(int\*\*\* matrix, int nx, int ny, int nz)
- void free\_matrix(int\*\*\* matrix, int ny, int nz)

3. Stellen Sie sicher, dass das Prüfskript check.cpp durchläuft. Hierzu muss eine matrix.h Datei erzeugt werden. Außerdem darf matrix.cpp nur noch die Funktionen, aber keine main-Funktion mehr enthalten. **Die Datei check.cpp darf nicht verändert werden.**

**Zeigen Sie das erfolgreiche Prüfskript mit Ausgabe bei der Abnahme:**

```
111 112 113
121 122 123
131 132 133
211 212 213
221 222 223
231 232 233
311 312 313
321 322 323
331 332 333
111
121
211
221
311
321
```