

# Programmierkurs (IPK)

Ole Klein

IWR  
Heidelberg University

27.10.2017

# Hintergrund

*Wer sind wir?*

Dozent: Dr. Stefan Lang

Kurzfristig eingesprungen, da der Dozent erkrankt ist:

- Dr. Steffen Müthing,  
<https://conan.iwr.uni-heidelberg.de/people/steffen/>
- Dr. Ole Klein,  
<https://conan.iwr.uni-heidelberg.de/people/oklein/>

Beide Postdocs in der Arbeitsgruppe von Prof. Dr. Peter Bastian  
(High Performance Computing)

Bitte bei Rückfragen, Problemen etc. immer an uns *beide* schreiben!

# Hintergrund (II)

Wird sind recht kurzfristig eingesprungen (weniger als eine Woche Vorlauf).

Unsere Mittel sind begrenzt:

- Zwei Postdocs, die eigentlich bereits in anderen Bereichen eingeplant sind
- Zwei Tutoren
- 125 angemeldete Teilnehmer

Komplikationen im Ablauf lassen sich u.U. nicht ganz vermeiden;  
bitte wenden Sie sich ggf. an einen unserer Tutoren oder direkt an uns.

# Survey

*Wer seid ihr?*

Um den Kurs möglichst angemessen gestalten zu können, müssen wir wissen, auf welches Publikum wir uns einstellen sollen.

Daher bitte kurz selbst einschätzen, welcher Gruppe Sie sich zuordnen würden:

- Informatiker/innen
- Mathematiker/innen
- Physiker/innen
- andere

Und:

- Keinerlei Programmierkenntnisse
- Ein wenig Programmiererfahrung
- Grundkenntnisse in C++

(Kriterien: coming up next)

# Ein wenig Programmiererfahrung

Ist das folgende für Sie verständlich?

```
1 #include<iostream>
3 int main()
4 {
5     int i;
6     std::cout << "Gib_eine_Zahl_ein:" << std::endl;
7     std::cin >> i;
9     while (i >= 0)
10    {
11        if (i == 8)
12            std::cout << "Du_hast_8_eingegeben ,gute_Wahl!" << std::endl;
13        else
14            std::cout << "Du_hast_" << i << "_eingegeben" << std::endl;
15
16        std::cout << "Gib_eine_weitere_Zahl_ein:" << std::endl;
17        std::cin >> i;
18    }
19    std::cout << "Die_Zahl_war_negativ ,Programmende..." << std::endl;
21 }
```

- Variablen, Schleifen, Verzweigungen, Ein-/Ausgabe

# Grundkenntnisse in C++

Ist das folgende für Sie verständlich?

```
1 template<typename Value>
  class Matrix
3   {
    private:
5     std::vector<std::vector<Value>> > entries;

7     public:
      Matrix(unsigned int size);
9     Matrix(unsigned int size, const Value& defaultEntry);

11    void add(const Matrix& otherMatrix);
    template<typename Vector>
13    Vector mult(const Vector& vector) const;
  };
```

- Funktionen
- Klassen und Objekte
- Vererbung
- Eventuell Schablonen (Templates)

# Survey

```
for (combi : combinations)  
    performSurvey(combi);
```

# Scheinkriterien

Kriterien für das Bestehen des Moduls:

- Die Prüfung des Moduls ist eine schriftliche Abschlussklausur
- Am Anfang ist *keine* Abgabe von Lösungen erforderlich
- Stattdessen: Vorführung des fertigen Programms / der gelösten Aufgabe in den Übungen
- Um an der Klausur teilnehmen zu dürfen, müssen 50% der Aufgaben *zufriedenstellend* vorgeführt worden sein
- Daher: Regelmäßige Teilnahme an den Übungen erforderlich
- Später: Einzelne Aufgaben zur Abgabe, ebenfalls 50% benötigt

Teamarbeit in Kleingruppen ist ausdrücklich gerne gesehen, aber das Vorführen findet immer einzeln statt!



# Scheinkriterien (II)

Falls es genügend Programmiererfahrene gibt, eventuell:

- Regelmäßiges Vorführen kann durch aktives Anleiten anderer ersetzt werden
- Ggf. neben Ersatzaufgaben, die im Niveau angepasst sind
- Unterrichten schärft auch das eigene Verständnis und vertieft Wissen
- Schult Kommunikationsvermögen und sicheres Auftreten
- In diesem Fall: Anwesenheitspflicht (!), nachweisbare aktive Teilnahme
- Wird von Tutoren überprüft und geleitet

Falls das nicht möglich ist:

- Der Kurs ist deutlich überbucht (125 Teilnehmer / 100 Plätze)
- Plätze werden bevorzugt an Informatiker vergeben (Pflichtkurs!), der Rest muss leider auf andere Semester verwiesen werden

# Zielsetzung des Kurses

Vermittlung der folgenden Kompetenzen:

- Selbständiger Entwurf von Lösungen für Programmieraufgaben
- Sicherer Umgang mit Programmierwerkzeugen (Editor, Compiler, IDE, ...)
- Kenntnis der grundlegenden Konzepte der Programmierung
- Grundkenntnisse in C++
- Grundkenntnisse im Umgang mit Linux / Unix
- Systematische Tests und Fehlersuche beim Programmieren

Konzept der Veranstaltung:

- Möglichst wenig Frontalunterricht
- Inhalte werden durch Ausprobieren erarbeitet (learning by doing)
- Kurze Vorträge zu ausgewählten Themen

# Und danach?

Für alle, die es nicht eh schon hören (?),  
Einführung in die Praktische Informatik (IPI):

- Behandelt die selbe Thematik, aber eher von der theoretischen Seite

Object-Oriented Programming for Scientific Computing (IOPSC):

- Modernes C++, umfassender Überblick, tiefergehende Konzepte
- [https://conan.iwr.uni-heidelberg.de/teaching/oopfsc\\_ss2017/](https://conan.iwr.uni-heidelberg.de/teaching/oopfsc_ss2017/)
- Möglicherweise im Sommer erneut angeboten

Einführung in die Numerik (MA7):

- Anwendung der hier erlernten Techniken auf mathematische Fragestellungen

# Einführungsveranstaltung

Nach dem heutigen Termin Vorlesungen

- nur nach vorheriger Ankündigung
- zu ausgewählten Themen und zur Vertiefung

Inhalt wird durch Ausprobieren in der Übung vermittelt!

Heute:

- Einführung in Unix
- Einführung in C++ (Folien vom letzten Jahr)
- Kurzes Beispielprogramm in C++ (von vorhin)