

# Programmieren in C++

## SS 2010

Vorlesung 4, Mittwoch 12. Mai 2010  
(strings, arrays, pointers + mehr über make)

Prof. Dr. Hannah Bast  
Lehrstuhl für Algorithmen und Datenstrukturen  
Institut für Informatik  
Universität Freiburg

# Blick über die Vorlesung heute

---

## ■ Organisatorisches

- Neue Frage-Termine, jetzt auch am Montag!
- Korrekturen für die ersten beiden Übungsblätter
- Erfahrungen mit dem [3. Übungsblatt](#)

## ■ Strings, Arrays, Pointer

- Das gehört alles zusammen
- Wie immer werde ich Ihnen das am Beispiel erklären
- Und auch zeigen wie diese Konzepte mit der Darstellung auf Maschinenebene zusammenhängt
- Unser Spielwelt werden heute Listen von Zahlen sein
  - Ich schreibe einen Teil (der diesmal wieder funktioniert)
  - Sie erweitern das dann für das [4. Übungsblatt](#)
- Bei der Gelegenheit lernen wir auch was Neues über [make](#)

# Geänderte Termine im TF-Pool

---

## ■ Und zwar

- Freitag, 16 – 18 Uhr, SR 00 028 im Gebäude 82 wie gehabt
- Montag, 14 – 16 Uhr, Raum 00 030 im Gebäude 51 NEU
- Dienstag, 16 – 18 Uhr, SR 00 028 im Gebäude 82 wie gehabt

## ■ Nur keine Hemmungen

- wenn Sie Fragen haben

# Korrekturen für das 1 + 2 Übungsblatt

---

- Läuft auch über das SVN
  - Sie machen `svn update`
    - in dem Ordner der den Namen Ihres RZ-Accounts hat
  - Dann bekommen Sie pro Übungsblatt
    - eine Datei `feedback-tutor.txt`
    - da stehen allgemeine Kommentare zu dem, was Sie abgegeben haben drin
    - mit evtl. Verweisen auf Kommentare in Ihrem Code
    - sowie im übergeordneten Verzeichnis Ihre Gesamtpunktzahl pro Übungsblatt in `punkte.txt`

# Erfahrungen mit dem 3. Übungsblatt

---

- Zusammenfassung / Auszüge
  - Die Aufgabe hat vielen gefallen
  - Vom Programm her deutlich schwieriger als Ü 1 + 2
  - Man konnte viele kleine Fehler machen
  - Entsprechend für die, für die Ü 1 + 2 leicht waren nicht mehr so leicht + zeitaufwändiger
    - 4 – 5 Stunden war typisch, einige 10 Stunden und länger
    - kein signifikanter Unterschied zwischen ESE und Info
  - Das Drumherum geht jetzt schon ganz gut
  - Linter nervt immer noch
  - Funktionen waren vorgegeben aber nicht kommentiert
  - Programm aus der Vorlesung hat nicht kompiliert

# Einfache Variablen

---

- Repräsentation einer normalen Variable im Speicher

# Felder (Arrays) und Zeichenketten (Strings)

---

- Oft braucht man eine Folge von Variablen vom selben Typ, auf die man alle mit demselben Namen und einem Index zugreifen möchte
  - zum Beispiel eine Zeichenkette
  - zum Beispiel die Zahlenlisten vom [4. Übungsblatt](#)
  - das nennt man dann ein [Feld](#), engl. [Array](#)
  - eine Zeichenkette ist auch nur ein Feld
    - mit Elementen vom Typ [char](#) (1 Byte in C / C++)
    - null-terminated

# Felder (Arrays) und Zeichenketten (Strings)

---

# Zeiger (Pointer)

---

- Ein Zeiger zeigt auf eine bestimmte Stelle im Speicher
  - Einfach eine ganze Zahl = die Adresse der Speicherzelle
  - Aber ein Zeiger hat einen Typ, und dieser Typ sagt wie das Stück Speicher was da anfängt zu interpretieren ist
    - je nach Typ auch mehr als eine Speicherzelle
  - ein Feld ist in C / C++ auch nichts anderes als ein Zeiger

- Arrays

- <http://www.cplusplus.com/doc/tutorial/arrays>

- Strings

- <http://www.cplusplus.com/doc/tutorial/ntcs>

- Pointers

- <http://www.cplusplus.com/doc/tutorial/pointers>

- Make Variablen + automatische Variablen

- <http://www.gnu.org/software/make/manual/make.html#Reference>

- <http://www.gnu.org/software/autoconf/manual/make/Automatic-Variables.html>

