

Programmieren in C++

SS 2010

Vorlesung 1, Mittwoch 21. April 2010
(Organisatorisches + Ein erstes Programm)

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

Blick über die Vorlesung heute

- Organisatorisches
 - Ablauf, Übungsblätter, Übungsgruppen, Regeln
 - Was mir wichtig ist / Was Sie hier lernen sollen
 - Zum Stil der Vorlesung
- Ein erstes Programm mit allem Drum und Dran
 - Berechnen der Quersumme
 - `Makefile` dazu schreiben
 - `Unit Test` dazu schreiben
 - Trennen in `.cpp` und `.h` file, `kompilieren` und `linken`
 - Ergebnis ins `SVN` einchecken
 - Schauen was das `Build System` sagt
 - Für's 1. Übungsblatt dann dasselbe für ein ähnliches Problem ...

- Die Vorlesung
 - fängt immer um 16.15 Uhr an und geht ca. 1 Stunde
- Die Übungen ...
 - sind der wichtigste Teil der Veranstaltung
 - Sie bekommen jede Woche ein Übungsblatt
 - Das können Sie machen wo Sie wollen
 - Aber Sie müssen es selber machen!
 - Ich mache Ihnen jeweils vor was Sie brauchen
 - Wir fangen gleich heute damit an
- Die Note ...
 - ergibt sich aus den Übungsblättern (10 Punkte / Blatt)
 - und am Ende gibt es ein kleines Projekt über mehrere Wochen

■ Die Übungsgruppen

- Das sind Zeiten zu denen im angegebenen Raum ein Tutor sitzt, den Sie fragen können
- Müssen Sie aber nicht, wenn da keiner kommt, lassen wir das einfach

■ Zusatzveranstaltung am Montag

- Ich gehe davon aus, dass das Niveau bez. Vorwissen sehr unterschiedlich ist
- Deswegen bei Bedarf Zusatzveranstaltung zu bestimmten Themen, z.B.
 - effizientes Benutzen von einem Editor
 - Linux Crash Course
 - informatischer Background (für die ESE Studierenden)

Was mir wichtig ist

- Fragen Sie wenn etwas nicht klar ist
 - in der Vorlesung
 - über den Wiki
 - Ihren Tutor
- Vorlesung
 - Ich würde Ihnen raten zu kommen
 - Aber sitzen Sie da nicht einfach nur passiv rum
 - Passen Sie die eine Stunde auf, stellen Sie Fragen wenn etwas nicht klar ist, und Sie werden mit dem Übungsblatt nur die Hälfte der Arbeit haben
 - Die Vorlesungen werden aufgenommen, das wird aber eher als Referenz nützlich (wenn man drin war)

Was Sie hier lernen sollen

- Programmieren in C++ nach den Regeln der Kunst
 - im Umfang von 500 – 1000 Zeilen
 - Umsetzen von einfachen **Algorithmen** in ein Programm
 - **Objektorientiertes** Programmieren
 - gutes **Klassendesign**, gute **Struktur**, gute **Namen**
 - gutes **Dokumentieren**
 - Trennung in **.cpp** und **.h** Dateien
 - Verständnis von **Compiler** und **Linker**
 - Benutzung eines **Build Systems** (gmake)
 - **Unit Tests** und **Performance Tests**
 - Benutzen eines **Versionsverwaltungssystems** (SVN)
 - Verwendung eines **Stylesheets** (cpplint.py)

Warum?

■ Es gibt Programme ...

- ... die lösen das gegebene Problem ... manchmal ... irgendwie
- Zeitaufwand beim Erstellen: 1 h Programmieren, 10 h Fehlersuche
- keiner außer dem Autor versteht das Programm
- in 1 Monat versteht es auch der Autor nicht mehr
- jegliche Änderung / Erweiterung unmöglich, je größer das Programm desto unmöglicher
- wenn man es nicht besser lernt schreibt man solche Programme auch noch in 10 Jahren, und dann lernst man es nimmer

■ Und dann gibt es Programme ...

- ... die sieht man nach 6 Monaten oder länger wieder und freut sich, dass man auf Anhieb alles versteht.
- dann macht Programmieren Spaß, sonst nicht so

Zum Stil der Vorlesung

- Ich werde das meiste anhand von Beispielen erklären
 - Für die Details gibt es genügend Referenzmanuale
 - Siehe Literaturliste am Ende
 - Und Sie kennen ja Google und Co
 - Ich werde schauen dass ich vor allem immer das erkläre was Sie auch gerade brauchen (für das nächste Übungsblatt)
- Fragen, Fragen, Fragen
 - Etwas ausprobieren, aber nicht zu lange, und dann fragen
 - Aber gut fragen (nicht: "Hilfe, mein Programm stürzt ab")
 - Viele Fragen werden für viele relevant sein, dann bitte über den Wiki

Unser Programm für heute

■ Berechnen der Quersumme

- Zum Beispiel Quersumme von $1234 = 1+2+3+4 = 10$
- Wir machen das nach folgendem Algorithmus
 - $1234 / 10 = 123$ (Teilen ohne Rest)
 - $1234 - 10 * 123 = 4$ (die letzte Ziffer)
 - $(1234 - 4) / 10 = 123$ (damit machen wir weiter)
 - so bekommen wir der Reihe nach alle Ziffern
 - bis nichts mehr von der Zahl übrig ist
- Dafür schreiben wir jetzt ein Programm nach den Regeln der Kunst
- Für das erste Übungsblatt machen Sie dasselbe für die Berechnung der Summe der Teiler einer Zahl

- C++

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

- GNU Make

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

- SVN

- <http://subversion.apache.org/>

- Google Test

- <http://code.google.com/p/googletest/>

