Programmieren in C++
(ESE) SS 2013

Vorlesung 1, Dienstag 24. April 2012
(Ein erstes Programm + das ganze Drumherum)

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

UNI FREIBURG

Blick über die Vorlesung heute

UNI FREIBURG

- Organisatorisches
  - Ablauf Vorlesungen / Übungen / Projekt am Ende
  - Punkte / Note / Aufwand / ECTS-Punkte
  - Voraussetzungen / Lernziele / Stil der Vorlesung
- Ein erstes Programm mit (fast) allem Drum und Dran
  - Berechnung ob ein gegebenes Jahr ein Schaltjahr ist
  - Kompilieren, Ausführen, Stylecheck
  - Makefile dazu schreiben
  - Unit Test dazu schreiben
  - Dateien ins SVN hochladen
  - Für das 1. Übungsblatt sollen Sie dann dasselbe machen für: die Berechnung ob eine gegebene Zahl prim ist

Ablauf: Vorlesungen

UNI FREIBURG

- Die Vorlesungen
  - ... sind Aufzeichnungen vom letzten Jahr

    Programmieren in C++, SS 2012
  - Insgesamt 12 Vorlesungstermine
  - Auf dem Wiki finden Sie alle Kursmaterialien
    - Aufzeichnungen, Folien, Dateien aus der VL, Übungsblätter, Hinweise zur Punktevergabe, Musterlösungen

Ablauf: Übungen / Projekt

UNI FREIBURG

- Die Übungen
  - ... sind der wichtigste Teil der Veranstaltung
  - Sie bekommen jede Woche ein Übungsblatt
  - Das können Sie machen wo und wann Sie wollen
  - Aber Sie müssen es **selber** machen!
  - Ich mache Ihnen jeweils vor, was Sie brauchen
- Das Projekt
  - ... am Ende der Vorlesung ist eine etwas größere Programmieraufgabe für eine etwas umfangreichere Aufgabe mit etwas weniger Vorgaben als bei den Übungen
  - Macht ca. 1/3 der Arbeit für die VL aus, wobei die letzten beiden Übungsblätter schon zum Projekt gehören

Ablauf: Übungsgruppen / Tutoren

UNI FREIBURG

- Die Übungsgruppen
  - ... gibt es in diesem Sinne nicht; insbesondere können Sie die "offiziellen" Übungstermine (vom QIS) ignorieren
  - Es gibt aber Termine, zu denen Sie kommen können wenn Sie Fragen / Probleme haben, siehe Wiki
- Die Tutoren
  - ... für diese Veranstaltung sind

    Janosch Deurer und Lukas Vögtle
  - Sie werden automatisch einem Tutor / einer Tutorin zugewiesen, siehe unser Kursverwaltungssytem Daphne
  - Die Tutoren korrigieren Ihre Abgaben, geben Ihnen Feedback, und sind Ihr Ansprechpartner für Fragen

Punkte

UNI FREIBURG

- Sie bekommen Punkte
  - Es gibt 20 Punkte pro Übungsblatt, das sind 200 Punkte für die Übungsblätter 1 bis 10
  - Es gibt 100 Punkte für das Projekt, davon 40 für die Übungsblätter 11 und 12, die schon zum Projekt gehören
  - Macht insgesamt 300 Punkte
  - Für das Ausfüllen des Evaluationsbogens am Ende gibt es 20 Punkte, mit denen Sie die Punktezahl des schlechtesten der Übungsblatter 1 – 10 ersetzen können

Note / Bestehen

UNI FREIBURG

- Die Note
  - ... ergibt sich linear aus der Gesamtpunktzahl am Ende

    150 – 164: 4.0; 165 – 179: 3.7; 180 – 194: 3.3
    195 – 209: 3.0; 210 – 224: 2.7; 225 – 239: 2.3
    240 – 254: 2.0; 255 – 269: 1.7; 270 – 284: 1.3
    285 – 300: 1.0
  - **Außerdem:** Abgabe des Projektes ist Voraussetzung zum Bestehen, auch wenn man sonst genug Punkte hätte!
  - **Außerdem 2:** Sie müssen sich einmal mit Ihrem Tutor / Ihrer Tutorin treffen, dazu mehr in einer der späteren Vorlesungen

Aufwand / ECTS-Punkte

UNI FREIBURG

- ECTS Punkte = Aufwand
  - Informatik / MST / ESE: 4 ECTS Punkte
  - Das entspricht 4 x 30 = 120 Stunden Arbeit
  - Davon 80 Stunden für die ersten 10 Vorlesungen + Übungen
    - also etwa 8 Stunden Arbeit / Woche
    - also etwa 6 Stunden pro Übungsblatt
  - Bleiben 40 Stunden für das Projekt und die dazu gehörigen letzten beiden Vorlesungen + Übungen

Was Sie (hoffentlich) schon können

UNI FREIBURG

- Wie man "im Prinzip" programmiert
  - Eine einfache Problemstellung in ein einfaches Programm umsetzen, zum Beispiel
    - die Zahlen von 1 bis 10 ausgeben
    - berechnen, ob eine gegebene Zahl prim ist
    - alle Primzahlen ≤ 1000 berechnen und ausgeben
    - eine Websuchmaschine mit natürlicher Sprachverarbeitung
  - Verständnis einiger Grundkonzepte
    - Variablen, Funktionen, Schleifen, Ein- und Ausgabe
- Wenn Ihnen das alles gar nichts sagt
  - ... können Sie trotzdem mitmachen, es wird aber dann mehr Arbeit für Sie, als es den ECTS Punkten entspricht

Was Sie hier lernen sollen

UNI FREIBURG

- Programmieren in C++ nach den Regeln der Kunst
  - ... im Umfang von 500 – 1000 Zeilen
  - Umsetzen von einfachen Lösungsideen in C++ Programme
  - Grundlegende Programmkonstrukte in C++
  - Objektorientiertes Programmierem
  - Gutes Design, gute Namen, gute Dokumentation
  - Verständnis von Compiler und Linker
  - Benutzung eines Build Systems (make)
  - Unit Tests und Performance Tests
  - Benutzen eines Versionsverwaltungssystems (SVN)
  - Verwendung eines Stylesheets (cpplint.py)

Warum, ich kann doch schon programmieren ...

UNI FREIBURG

- Es gibt Programme ...
  - ... die lösen das gegebene Problem ... irgendwie ... manchmal
  - Zeitaufwand beim Erstellen: 1 h Programmieren, 10 h Fehlersuche
  - Keiner außer dem Autor versteht das Programm
  - In einem 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 lernt man es nicht mehr
- 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

UNI FREIBURG

- Ich werde das meiste exemplarisch vormachen
  - Für die Details gibt es genügend Referenzmanuale
    - insbesondere in der Linux-Shell: man 3 <Funktion>
  - Siehe auch die Referenzen auf der letzten Folie
  - Und Sie kennen ja Google und Co
  - Ich werde vor allem immer das erklären, was Sie auch gerade brauchen (für das nächste Übungsblatt)
- Fragen, Fragen, Fragen
  - Etwas ausprobieren, aber nicht zu lange, und dann fragen!
  - Die meisten Fragen interessieren auch andere, von daher vorzugsweise über das Forum (oder gleich in der Vorlesung)
  - Zum Forum später noch mehr ...

Unsere Entwicklungsumgebung

UNI FREIBURG

- Wir machen hier alles ganz "low-level"
  - Linux, Kommandozeile, Texteditor, Makefile
  - Was das konkret heißt, sehen Sie gleich
  - So lernt man am besten was "under the hood" passiert
  - Aufwändige Entwicklungsumgebungen (Eclipse, NetBeans, Visual Whatever) sind was für später, wenn man den "low level" verstanden hat
  - Wir werden das Thema im Lauf der Vorlesung noch öfter diskutieren

Unser Programm für heute

UNI FREIBURG

- Wir schauen uns ein sehr einfaches Problem an
  - Ist ein gegebenes Jahr ein Schaltjahr oder nicht?
  - Beispiele: 1900: Nein; 1984: Ja; 2000: Ja; 2022: Nein
  - Es kommt heute weniger auf das Problem an, sondern es geht vor allem um das ganze Drumherum
- Wir machen das jetzt zusammen live
  - Erstmal das ganze Programm in einer main Funktion
  - Kompilieren und Ausführen (von der Kommadozeile)
  - Dann Stylecheck und ins SVN hochladen
  - Dann separate Funktion für die Schaltjahrberechnung
  - Dann ein Unit Test für diese Funktion
  - Schließlich ein (einfaches) Makefile für das Ganze

Make / Makefiles

UNI FREIBURG

- Ein sehr mächtiger Mechanismus
  - Wir benutzen es erstmal nur als eine Art Abkürzung für Befehle, die wir immer wieder verwenden
    - z.B. zum Kompilieren, Linten, Testen
  - Die Syntax im Makefile dafür ist einfach wie folgt

```
<target>:
        <Befehl 1>
        <Befehl 2>
        ...
```

Achtung: jede der Befehlszeilen muss mit einem TAB anfangen!

  - Wenn man dann (in dem Verzeichnis, in dem das Makefile steht) make <target> ausführt, werden einfach die entsprechenden Befehle ausgeführt
  - In den nächsten Vorlesungen mehr zu make ...

Stylesheet / Lint

UNI FREIBURG

- Stylesheets sind wichtig
  - Nicht nur der Compiler muss Ihren Code verstehen, sondern auch andere Menschen
    - z.B. Ihr Tutor, ein Teamkollege (für spätere Projekte), oder Sie selber in drei Monaten ...
  - Deshalb wichtig, sich an bestimmte Konventionen zu halten
    - manche Konventionen sind das Ergebnis langjähriger Programmiererfahrung, z.B. explicit (kommt später)
    - andere Konventionen sind einfach nur Standards um der Konsistenz willen, z.B. Einrückungstiefe
  - In Ihrer SVN Arbeitskopie liegt ein Skript cpplint.py, dass den korrekten Stil überprüft → soll ohne Fehler durchlaufen !

Google Test

UNI FREIBURG

- Google Test ist ein Unit Test Framework für C++
  - Unit Tests testen die Korrektheit einzelner Funktionen, typischerweise anhand von Spezialfällen
  - Ein guter Unit Test testet insbesondere
    - Randfälle, bei denen potenziell etwas schief gehen kann
    - mindestens einen typischen allgemeinen Fall
  - Im Rahmen dieser Vorlesung werden wir fast immer sehr einfache Unit Tests schreiben; sie sind trotzdem nützlich
    - ... weil man auf diese Weile schon sehr viele Fehler findet
    - ... weil die Fehlersuche so viel mehr Spaß macht
    - ... weil einen das Schreiben von Unit Tests zum Nachdenken darüber bringt, was die Funktion eigentlich genau berechnen soll

SVN

UNI FREIBURG

- SVN = Subversion
  - SVN ist ein sogenanntes Versionskontrollsystem
  - Es gibt ein sogenanntes Repositorium, das ist einfach ein Verzeichnisbaum mit Dateien drin, die liegen bei uns am Lehrstuhl auf einem Rechner
  - Jeder, der sich (via Daphne) bei uns registriert, hat ein Unterverzeichnis dort → URL siehe Ihre Daphne-Seite
  - Sie bekommen eine Kopie dieses Verzeichnisses mit

    svn checkout <URL> --username=<Ihr RZ Username>
  - In Ihrer Arbeitskopie können Sie dann Sachen ändern, Unterordner und Dateien hinzufügen, etc.
    - svn add <file name> fügt eine Datei erstmals hinzu
    - svn commit <file name> lädt die Änderungen zu uns hoch

Daphne

UNI FREIBURG

- Unser Kursverwaltungssystem
  - Wird von Axel Lehmann bei uns am Lehrstuhl entwickelt / gepflegt
  - Macht mir, den Tutoren, und hoffentlich auch Ihnen das Leben leichter
  - **Registrieren Sie sich bitte** nach der Vorlesung dort
    - dadurch bekommen Sie auch ein Unterverzeichnis (heißt so wie ihr RZ Username) in unserem SVN
    - Bei Problemen Mail an daphne@lists.informatik.uni-freiburg.de (das geht dann an Jens und Axel); die Adresse steht auch auf Daphne und dem Wiki

Forum

UNI FREIBURG

- Es gibt ein Forum für Fragen aller Art
  - Machen Sie bitte regen Gebrauch davon; Link auf dem Wiki
  - Haben Sie **keine Hemmungen**, Fragen zu stellen
    - Selbst wenn Sie denken, die Frage ist blöd ... mit großer Wahrscheinlichkeit haben ein paar andere genau dieselbe blöde Frage
  - Geben Sie sich aber gleichzeitig Mühe, Ihre Fragen möglichst genau und konkret zu stellen, zum Beispiel (nicht)

    "Mein Programm stürzt ab, warum?"

    "Wenn ich mein Programm mit <genauer Aufruf> aufrufe kommt die Fehlermeldung <genaue Fehlermeldung>. Hier die entsprechenden Zeilen aus meinem Programm <Programmzeilen auf die sich die Fehlermeldung bezieht>"

Literatur / Links

- C++
  - http://www.cplusplus.com/doc/tutorial/
- Make
  - http://www.gnu.org/software/make/manual/
- SVN
  - http://subversion.apache.org/
- Google Test
  - http://code.google.com/p/googletest/

+ auf dem Wiki stehen kurze Einführungen dazu