Programmieren in C++

SS 2010

Vorlesung 12, Mittwoch 14. Juli 2010

(Projekt, Debugging mit gdb)

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

UNI FREIBURG

Blick über die Vorlesung heute

UNI FREIBURG

- Organisatorisches
  - Erfahrungen mit dem 11. Übungsblatt
  - Offizielle Evaluation dieser Veranstaltung
  - Alles Mögliche andere ...
- Projekt
  - Details zum Ablauf des Projektes
  - Übungsblatt 12: der erste Schritt vom Projekt
    - die .h Dateien mit allen Methoden
    - die ...Main Datei
    - Dummy-Implemtierung in der C++ Datei, so dass man die ...Main schon aufrufen kann
- Debugging
  - Debugging mit gdb (= GNU debugger)

Erfahrungen mit dem 11. Übungsblatt

UNI FREIBURG

- Zusammenfassung / Auszüge
  - Vererbung wurde gut erklärt
  - Aber newMetricSpaceElement hat Probleme gemacht
  - Recht kniffelig, obwohl nicht viel zu programmieren
  - Wenn man Vererbung mal verstanden hat, einfach
  - Noch viele Probleme mit * und &
  - Lint-Einrückungs-Check noch nicht perfekt
  - Kein optimales Beispiel für Vererbung

Offizielle Evaluation

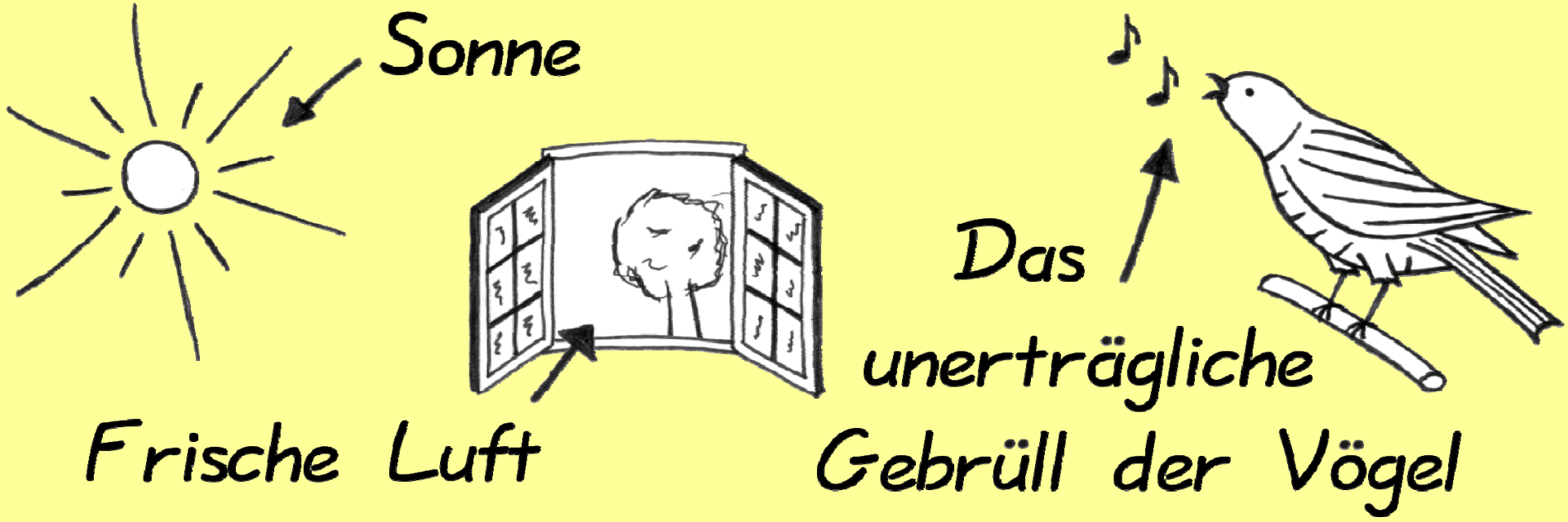
UNI FREIBURG

- Was müssen Sie tun
  - Evaluationsbogen auf dem Vorlesungswiki
  - Bitte ausdrucken und bis zum 21. Juli abgeben
    - jederzeit bei Jens Hoffmann oder bei mir (Geb. 51, 2. OG)
    - oder zu Beginn der nächsten (= letzten) Vorlesung
  - Es gibt dafür 10 Punkte (sollten Sie sich nicht entgehen lassen)
    - es reicht dafür ein Satz in den erfahrungen.txt, dass Sie den Bogen abgegeben haben (die Bögen selber sind anonym)
  - Nehmen Sie sich bitte Zeit dafür, nicht einfach husch-husch
    - Ihr aussagekräftiges Feedback ist uns sehr wichtig
  - Wenn Sie möchten, ist auch eine elektronische Abgabe ok
    - das PDF elektronisch ausfüllen ist allerdings nicht-trivial
    - Upload über das Forum, Abschnitt Feedback

Forum

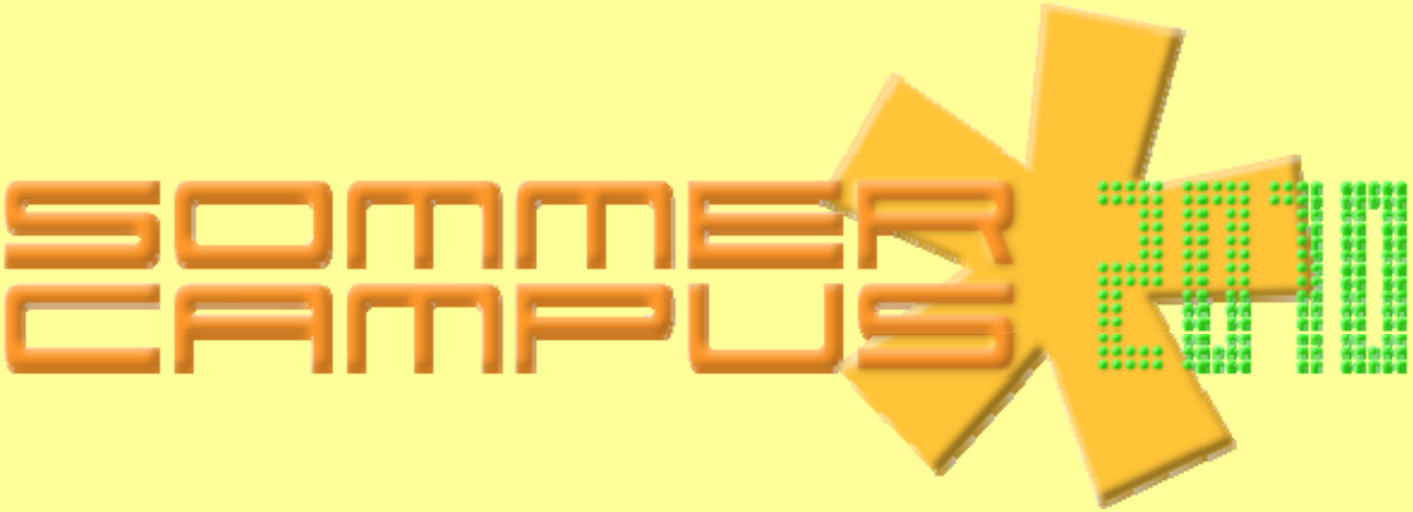
- Hat ja dann schnell sehr gut geklappt
  - War erst etwas gewöhnungsbedürftig (für mich)
  - Aber deutliche Verbesserung gegenüber der langen Liste auf dem Wiki
  - An dieser Stelle herzlichen Dank an
    - **Sebastian Sester** (ESE, 2. Semester)
    - **Jonathan Nagel** (Informatik. 2. Semester)

    für das Einrichten !!!



Das muss doch nicht sein!

Die Alternative:



27. Sept – 1. Okt 2010

Du möchtest im Organisationsteam mitarbeiten?
Du kannst einen Kurs anbieten?

Dann besuche unsere Website unter

http://sommercampus2010.informatik.uni-freiburg.de

Das Projekt

UNI FREIBURG

- Es gibt drei Projekte zur Auswahl
  - Ein Kalender
  - Ein (De-)Komprimierungsprogramm
  - Ein frei definiertes Projekt Ihrer Wahl
- Ablauf
  - Bis zur nächsten Vorlesung vollständiger Entwurf
    - die .h Dateien, die Main Funktion, und Dummy-Implementierungen, damit die Main schon mal läuft
    - dafür gibt es (bis zu) 10 Punkte
  - Den Rest dann nach der letzten Vorlesung
    - dafür gibt es (bis zu) 30 Punkte
  - Gruppenarbeit ist grundsätzlich **nicht** erlaubt

Projekt 1

- Kalender
  - Ausführlichere Version des Kalenders vom 3. Übungsblatt
    - mit Monatsnamen
    - in mindestens drei verschiedenen Sprachen (wählbar)
    - mit Kalender für ein ganzes Jahr
      - bei beliebig vorgebbarer Zeilenbreite
    - highlighting des aktuellen Tages
  - Separate Klasse für das Malen von mehreren "Textrechtecken" mit einer gegebenen Anzahl von Rechtecken pro Reihe

Projekt 2

- (De-)Komprimierung
  - Komprimierung bzw. Dekomprimierung einer beliebigen Datei (a la gzip oder zip)
  - Mindestens zwei verschiedene (De-)Komprimierungsalgorithmen
    - Welche ist Ihnen überlassen, Hauptsache nicht-trivial
    - Sie müssen sich selber umschauen, es gibt aber Dutzende von Verfahren, von relativ einfach bis ziemlich komplex
    - Realisierung über abstrakte Oberklasse, mit einer konkreten Unterklasse pro Verfahren

Projekt 3

- Ein Projekt Ihrer Wahl
  - Es sollte vom Umfang her vergleichbar sein mit den anderen beiden Projekten
  - Es sollte von der Methodik her ähnlich anspruchsvoll sein

    (insbesondere soll Ihr Tutor es auch korrigieren können)
  - Im Zweifelsfall bei Jens Hoffmann rückfragen (cc an mich)
  - Sonst machen Sie einfach bis nächste Woche einen Designvorschlag, so wie es Übungsblatt 12 auch für Projekt 1 bzw. 2 fordert
  - Ansonsten gelten für dieses Projekt genau die gleichen Bedingungen wie für Projekt 1 oder 2

Gesamtnote

- Bedingungen
  - Mindestens die Hälfte der Punkte in den Übungsblättern
  - Mindestens die Hälfte der Punkte für das Projekt
- Berechnung der Note
  - 60% der Gesamtpunktzahl reicht zum Bestehen
  - Je mehr Punkte desto besser die Note (ach)
  - Details dazu in der letzten Vorlesung am 21. Juli
- Um durchzufallen muss man also
  - Kaum Übungsblätter abgegeben haben; oder
  - Das Projekt nicht ernsthaft bearbeiten; oder
  - Geschummelt haben (Plagiat); oder
  - Sich dem Treffen mit dem Tutor entzogen haben

Treffen mit Ihrem Tutor

- Falls Sie bisher noch keine E-Mail bekommen haben
  - Checken Sie bitte, dass Ihre daten.txt richtig ausgefüllt ist
  - Insbesondere, dass Sie unter der E-Mail Adresse, die dort steht, auch E-Mails empfangen können

Debugging

- Methode 1 ("printf")
  - printf statements einbauen
    - an Stellen wo der Fehler vermutlich auftritt
    - von Variablen wo man denkt dass etwas falsch läuft
  - Haupt-Vorteil
    - einfach
  - Haupt-Nachteil
    - man muss jedesmal rekompilieren (das kann bei größeren Programmen lange dauern)
    - nicht interaktiv

Debugging

- Methode 2 ("gdb")
  - mit dem gdb = GNU debugger
  - der kann so Sachen wie
    - Anweisung für Anweisung durch das Programm gehen
    - sogenannte breakpoints im Programm setzen und zum nächsten breakpoint springen
    - Werte von allen möglichen Variablen ausgeben
  - das wollen wir jetzt mal anhand eines Beispiels machen
  - Haupt-Vorteil:
    - viel interaktiver als mit printf statements
  - Haupt-Nachteil:
    - man muss sich die gdb Kommandos merken

GDB

- Ein paar grundlegende Kommandos
  - gdb aufrufen, z.B. gdb ./NumberTest
  - Programm starten mit run <command line arguments>
  - breakpoint setzen, z.B. breakpoint Number.cpp:47
  - breakpoints löschen mit delete
  - Weiterlaufen lassen mit continue
  - Nächste Programmzeile ausführen step
  - Wert einer Variablen ausgeben, z.B. print c1
  - stack trace (nach seg fault) mit backtrace
  - Kommandoübersicht / Hilfe mit help oder help all
  - gbd verlassen mit quit
  - Wie in der bash command history mit Pfeil hoch / runter
  - Und suchen in der history mit Strg+R

Literatur / Links

- gdb = GNU debugger
  - http://sourceware.org/gdb/current/onlinedocs/gdb/
- Linux calender tool (cal)
  - http://linux.about.com/library/cmd/blcmdl1_cal.htm
  - Und einfach mal von der Kommandozeile aus aufrufen

    z.B. cal July 2010 oder cal 2011
- (De-)Komprimierungsalgorithmen + Beispielverfahren
  - http://en.wikipedia.org/wiki/Data_compression#Lossless_data_compression
  - http://en.wikipedia.org/wiki/Run-length_encoding
  - http://en.wikipedia.org/wiki/Huffman_coding
  - http://en.wikipedia.org/wiki/LZW