Dr. M. Grabert
Johannes Mayer Abteilung Angewandte Informationsverarbeitung 8. November 2001
Blatt 1

Objektorientierte Programmierung mit C++ (WS 2001/2002)

Abgabetermin: 25. Oktober 2001

1 Ärger mit den Matrikelnummern (10 Punkte)

In dieser Woche geht's bei uns rund. Alle reißen sich um einen Account auf unseren Rechnern. Da ist natürlich einiges zu tun für unsere Tutoren, die die Daten der Studenten aufnehmen, welche einen Account wollen. Dabei können sie das *SLC* von Ingo Melzer verwenden. Damit sich möglichst wenig Fehler beim Eintippen der Daten einschleichen, werden diese auf Plausibilität hin überprüft. So gibt es zum Beispiel von der Uni-Verwaltung bisher schon zwei Algorithmen, mit denen man feststellen kann, ob eine Matrikelnummer korrekt ist – denkste! ;-)
Realität ist leider, dass es jetzt eine ganze Menge von Matrikelnummern gibt, die von diesen Algorithmen als falsch eingestuft werden. (Und die Verwaltung hat uns nichts von einem neuen Algorithmus zur Überprüfung gesagt. ;-)) So konnten also haufenweise Daten nicht sofort eingegeben werden, da die Matrikelnummern nicht angenommen wurden. Also hat Ingo Melzer kurzerhand die Überprüfung der Matrikelnummern ausgeschaltet, damit die Accounts angelegt werden konnten.
Wir haben jetzt also eine kleine *Datenbank* mit den Logins, Namen und Matrikelnummern von den Accounts, die neu angelegt wurden. Zumindest eine triviale Überprüfung – nämlich, ob es sich um eine 6- oder 7-stellige Zahl handelt – der Matrikelnummern möchten wir doch noch durchführen, um auf Nummer sicher zu gehen. Gesagt, getan: Ich krame alles zusammen, was ich noch über C weiß und schreibe ein paar kleine C-Progrämmchen, die die kleine Datenbank einlesen und alle Datensätze sortiert ausgeben, deren Matrikelnummer bei der Überprüfung als nicht korrekt gemeldet wird. Geschafft – denke ich!
Doch mein Chef hätte das Hauptprogramm lieber als C++-Programm. Knurr ... Aber halt, wozu gibt es denn die C++-Übungen – lassen wir das doch die Studenten für mich erledigen! Gute Idee! :-))
Sie bekommen also von mir die Programme *main.c*, *student.c*, *student.h* und *compare.c*, die ich geschrieben habe, zusammen mit dem *makefile* von mir.
Bis nächste Woche geben Sie Ihrem Tutor bitte die portierte Version. Der leitet sie dann an mich weiter. :-)
Benennen Sie zunächst das Programm *main.c* in *main.cc* um. (Denken Sie dabei auch an die Änderungen, die deshalb im makefile notwendig sind.) Lassen Sie das ganze jetzt doch 'mal

laufen, d. h. lassen Sie `make` 'mal die ausführbare Datei erzeugen. Ach, das tut nicht? Na, jetzt sind Sie dran! Ändern Sie *ausschließlich* die Datei *main.cc* – an den anderen Dateien dürfen Sie nur im Notfall eine Änderung vornehmen, d. h. wenn es keinen anderen Weg gibt! Und jetzt an die Arbeit – mein Chef will schließlich nächste Woche das C++-Programm von mir sehen! :-))

P. S.: Ich bin ja nicht so, eine *Datenbank* gebe ich Ihnen natürlich, damit Sie das Programm testen können. Das Programm liest von der Standardeingabe und schreibt auf die Standardausgabe. D. h. Sie müssen das compilierte Programm wie folgt aufrufen:

```
thales$ main < test.db
```

Dann wir die Datenbank aus der Datei *test1.db* gelesen und die Datensätze, deren Matrikelnummer als nicht korrekt eingestuft wird, werden sortiert nach der Matrikelnummer ausgegeben. (Rufen Sie das Programm wie folgt auf

```
thales$ main --login < test.db
```

dann erfolgt die Ausgabe sortiert nach dem Login.)

P. P. S.: Aus der Postscript-Datei `blatt1.ps` kann man mit folgender Kommandozeile eine Postscript-Datei `blatt1.2ps` erzeugen, bei der zwei Seiten der ursprünglichen Datei auf einer Seite untergebracht sind, um Druckkosten zu sparen :-)) :

```
thales$ psnup -2 blatt1.ps blatt1.2ps
```

Viel Spass und Erfolg!