Prof. Dr. Stefan Brass
Dipl.-Inform. Annett Thüring
Institut für Informatik
MLU Halle-Wittenberg

01. November 2010

Vorlesung "Objektorientierte Programmierung" — Beispiel für 1. Programmiertest —

Hinweise:

Beim echten Programmiertest (am 8./9. November) gelten folgende Regeln. Sie würden dann disqualifiziert, wenn Sie gegen die Regeln verstoßen (d.h. Sie hätten den Programmiertest nicht bestanden wegen Täuschungsversuch). In der Übung ist es jetzt natürlich noch nicht so streng: Sie sollen ja alle Möglichkeiten nutzen, noch etwas zu lernen. Andererseits wäre es wohl klug, zunächst einmal zu sehen, wie weit Sie alleine kommen.

- Bearbeitungsdauer: 60 Minuten
- Am Ende gibt es nur "bestanden" und "nicht bestanden", keine Punkte für partiell korrekte Lösungen. Möglicherweise werden am Ende noch einige Minuten Bearbeitungszeit angehängt, wenn mehrere Teilnehmer ganz knapp vor einer fertigen Lösung sind — aber wir haben nicht vor, nicht funktionierende Lösungen zu akzeptieren, selbst wenn nur eine Kleinigkeit fehlt.
- Sie dürfen bis zu 3 Blätter "Spickzettel"/"Quick Reference" verwenden (ggf. mit Vorder- und Rückseite beschriftet). Außerdem dürfen Sie ein Buch verwenden (nicht zu groß, es muß noch auf den Tisch passen ohne den Nachbarn zu stören — ein Aktenordner wäre nicht akzeptabel).
- Sie dürfen außerdem ein Blatt leeres Papier und einen Stift verwenden, um sich Notizen zu machen.
- Notebooks, PDAs, etc. dürfen nicht verwendet werden. Mobiltelefone bitte ausschalten (oder mit Aufsicht besprechen).
- Sie dürfen nur Terminalfenster für den Compileraufruf und Editor-Fenster offen haben. Selbstverständlich können Sie auch einen Debugger benutzen und make. Dagegen ist z.B. ein Web-Browser nicht erlaubt, und auch kein EMail-Programm. Ebensowenig dürfen Sie versuchen, Dateien außerhalb Ihres Homeverzeichnisses abzulegen.
- Selbstverständlich dürfen Sie auch Microsoft Visual Studio benutzen. Das abgegebene Programm muss aber unter Linux/g++ laufen.
- Tauschen Sie keinesfalls irgendwelche Dinge mit den Nachbarn aus. Notfalls rufen Sie eine Aufsichtsperson zur Kontrolle.
- Fragen Sie, wenn Ihnen die Aufgabe nicht klar ist!
- Wenn Sie an einer unverständlichen Fehlermeldung länger festhängen, können Sie probieren, zu fragen. Wir wollen aber nicht zu viele Tipps geben.

Aufgabe

Bekannt ist ja die Teilbarkeitsregel: "Eine Zahl ist durch 3 teilbar genau dann, wenn ihre Quersumme durch 3 teilbar ist". Die Quersumme ist die Summe der Dezimalziffern einer Zahl, z.B. hat 123 die Quersumme $1+2+3 = 6$. Sie sollen nun ein Programm schreiben, das für eine eingegebene Zahl die entsprechende Regel für die Teilbarkeit durch 7 überprüft. Diese Regel ist im allgemeinen falsch, z.B. hat 28 die Quersumme 10, und 28 ist durch 7 teilbar, aber 10 ist nicht durch 7 teilbar. Der umgekehrte Fall tritt auch auf, z.B. hat 61 die Quersumme 7: Hier ist die Quersumme teilbar, aber die Zahl nicht. Es gibt aber auch Fälle, wo es zufällig passt: Z.B. ist 133 durch 7 teilbar, und die Quersumme ist 7, und damit auch teilbar.

Ihr Programm soll zunächst folgenden Text ausgeben:

```
Bitte eine Zahl eingeben:
```

(am Ende mit einem Leerzeichen, aber ohne Zeilenumbruch). Dann soll es eine ganze Zahl einlesen. Fehlerhafte Eingaben brauchen Sie nicht zu behandeln: Sie können sich darauf verlassen, dass eine positive ganze Zahl eingegeben wird. Anschliessend sollen Sie die Zahl und ihre Quersumme, jeweils zusammen mit der Information, ob durch 7 teilbar, ausgegeben werden. Hat der Benutzer z.B. 28 eingegeben, sollen Sie folgenden Text ausgeben:

```
Zahl: 28 (teilbar)
Quersumme: 10 (nicht teilbar)
```

Wie bekannt, können Sie in C++ mit `n % m` den Rest bei Divison von `n` durch `m` bestimmen (`%` ist der Modulo-Operator). Damit können Sie die Teilbarkeit überprüfen.

Zur Berechnung der Quersumme können Sie jeweils die letzte Ziffer der Zahl als Divisionsrest bei Division durch 10 bestimmen, und dann die Zahl durch 10 teilen. Bei der ganzzahligen Division von positiven Zahlen wird in C++ immer abgerundet. Z.B. würden Sie bei der Eingabe 123 zuerst den Divisionsrest (Modulo 10) 3 erhalten (letzte Ziffer). Bei 123/10 ergibt sich 12. Nun bestimmen Sie wieder den Divisionsrest bei Divison durch 10 (Ziffer 2) und teilen wieder durch 10 (Ergebnis 1). Auf diese Art erhalten Sie nach und nach alle Ziffern der Zahl, und können die Summe dieser Ziffern bestimmen.