Prof. Dr. Franz Schweiggert — Abteilung Angewandte Informationsverarbeitung — 07.12.2005
Norbert Heidenbluth — Blatt 8

Allgemeine Informatik I für Mathematiker/Wirtschaftsmathematiker (WS 2005/2006)

Abgabetermin: 14. Dezember 2005

Auch in diesem Übungsblatt geht es wieder um Algorithmen und deren Implementierung in Java. In Aufgabe 17a) bekommen Sie einen Algorithmus vorgegeben (der sogar wirklich eine reale Anwendung hat, allerdings nicht in Zusammenhang mit Reisepässen!), in Aufgabe 17b) ist dieser Algorithmus dann selbständig zu invertieren. Aufgabe 18 gibt Gelegenheit, sich zu einem gegebenen Problem den passenden Algorithmus selbst auszudenken.

Aufgabe 17: Fälschungssichere Reisepässe? (6 Punkte)

Den neuen biometrischen Reisepaß finden Sie zwar ganz toll, aber irgendwie kommen Sie nicht darüber hinweg, daß sich während dessen Entwicklung niemand um Ihre sensationelle Idee gekümmert hat, wie man diesen Paß noch fälschungssicherer machen kann. Sie haben sich nämlich einen Algorithmus überlegt, der aus den Initialen des Vor- und Nachnamens eine Zahl (Prüfzahl) berechnet. Wenn man diese Zahl nun mit in die Seriennummer des Passes einarbeitet, hilft es, Personen zu finden, die mit falschem Namen unterwegs sind (weil diese ja Ihren streng geheimen Algorithmus nicht kennen und daher eine unpassende Seriennummer im Paß führen).

Nun hat sich wie gesagt niemand für Ihre Idee interessiert (vielleicht, weil auch schon 1.000 Leute vor Ihnen auf diese Idee gekommen waren). Als harte(r) Naturwissenschaftler(in) geben Sie allerdings nicht auf und programmieren Ihren Algorithmus und dessen Rückrichtung nun in Java – in der Hoffnung, daß dies wenigstens Ihren Tutor begeistert und er Ihnen hierfür ein paar Punkte einträgt.

Daraus ergibt sich dann auch gleich die (zweiteilige) Aufgabe 17:

Teil a) Berechnung der Prüfzahl aus den Initialen (2 Punkte)

Die Berechnung der Prüfzahl aus den vorliegenden Namen (Vorname / Nachname) geschieht nach dem folgenden Algorithmus:

1. Nimm den ASCII-Wert des ersten Buchstabens des Nachnamens und multipliziere ihn mit 100!
2. Addiere hierzu den ASCII-Wert des **ersten Buchstabens des** Vornamens!
3. Addiere den Wert 2468 und verdopple die erhaltene Summe!

Das Ergebnis dieser Berechnung ist die Prüfsumme!

Schreiben Sie ein Java-Programm, das sowohl den Nachnamen als auch den Vornamen einliest und gemäß vorstehendem Algorithmus die Prüfsumme berechnet (und ausgibt)!

Beispiel:

```
theseus$ java CalcCheckVal
Bitte den Nachnamen eingeben:
Wurst
Bitte den Vornamen eingeben:
Hans
Der berechnete CheckVal ist: 22480.
```

Teil b): Berechnung der Initialen aus der Prüfsumme (4 Punkte)

Teil a) war ja nur zum Aufwärmen und sollte kein Problem dargestellt haben. Nun aber soll die Berechnung invertiert, d.h. aus einer gegebenen Prüfsumme sollen die Initialen berechnet werden.

Schreiben Sie ein weiteres Java-Programm, das eine Prüfsumme einliest, daraus die Initialen berechnet und diese ausgibt!

Tip: Machen Sie sich zunächst über die (eindeutige!) Umkehrbarkeit des Algorithmus Gedanken. Dies betrifft insbesondere die Fälle, in denen der Vorname mit einem (großen) Umlaut beginnt!

Beispiel:

```
theseus$ java CalcInitials
Bitte geben Sie den Check-Value ein:
22480
Initial des Vornamens: H --- Initial des Nachnamens: W
```

Aufgabe 18: Zahlen-Konverter (4 Punkte)

Die Aufgabe 9 vom 5. Übungsblatt wäre doch bestimmt viel einfacher gewesen, wenn Sie ein Programm gehabt hätten, mit dem Sie Ihre Ergebnisse hätten kontrollieren können.

Okay, der ein oder andere Taschenrechner kann das auch, aber viel spannender ist es, ein solches Konvertierungs-Programm einmal selber zu schreiben. Das motiviert doch die folgende Aufgabe:

"Schreiben Sie ein Programm, das von der Standardeingabe eine Dezimalzahl und eine Basis (zwischen 2 und 16) einliest, die Dezimalzahl in das Zahlenformat zur eingegebenen Basis umrechnet und das Ergebnis auf der Standardausgabe ausgibt!"

Beispiel:

```
theseus$ java Converter
Welche Basis sollen wir verwenden? 2
Geben Sie nun die zu konvertierende Dezimal-Zahl ein: 120
Ergebnis der Umrechnung: 1111000

theseus$ java Converter
Welche Basis sollen wir verwenden? 16
Geben Sie nun die zu konvertierende Dezimal-Zahl ein: 511
Ergebnis der Umrechnung: 1FF

theseus$ java Converter
Welche Basis sollen wir verwenden? 99
Die eingegebene Basis ist ungueltig!
```

Hinweise zu beiden Aufgaben

- Die Quellcode muß lesbar (d.h. insbesondere eingerückt) sein.
- Verwenden Sie sprechende Variablennamen!
- Hilfreich für die Aufgabe 17 ist sicherlich ein Blick in die ASCII-Tabelle! (Damit Ihre Lösung auch auf unseren Rechnern läuft, sollten Sie die für die Codierung der Umlaute die ISO-8859-Erweiterung verwenden.)
- Sie können Ihr Programm aus Aufgabe 17 mit den folgenden Werten auf seine Korrektheit hin testen:
  - Die Prüfsumme des Namens "Hein Blöd" ist 18280.
  - Aus der Prüfsumme 18976 ergibt sich **eindeutig(!)** als Initial des Vornamens das Ü und als Initial des Nachnamens das D.
- Hinweise zum Erhalt des ASCII-Wertes zu einem gegebenen Zeichen (und umgekehrt) bekommen Sie in den Übungen.
- Bauen Sie in Ihre Programme Fehlerbehandlungen ein, d.h.:
  - bei Aufgabe 17:
    - Sofern der Nachname und/oder der Vorname nicht korrekt eingelesen werden konnten, soll das Programm mit einer entsprechenden Fehlermeldung abbrechen.
    - Es sollen nur solche Namen (Vor- und Nachname!) akzeptiert werden, die mit einem Großbuchstaben ("A" – "Z") oder einem großen Umlaut (Ä, Ö, Ü) beginnen. Trifft dies nicht zu, soll das Programm ebenfalls mit einer entsprechenden Fehlermeldung abbrechen.
  - bei Aufgabe 18:
    - Wenn eine Basis < 2 oder > 16 eingegeben wird, sollte Ihr Programm mit einer Fehlermeldung abbrechen.

Viel Erfolg!