

Übungen zu Algorithmen

Wintersemester 2003/2004

Blatt 2

Aufgabe 2.1 (0 Punkte)

Machen Sie sich mit den folgenden UNIX-Kommandos vertraut:

Kommando	Erklärung
man man	Ausgabe der Manualseite zum Kommando man (<i>manual</i>).
more 15prozent	Seitenweise Auflisten der Datei 15prozent.
ls -l	Auflisten aller Dateien und Kataloge in Langform (<i>list</i>).
mkdir Blatt1	Anlegen des Katalogs Blatt1 (<i>make directory</i>).
rmdir Blatt1	Löschen des (leeren) Katalogs Blatt1 (<i>remove directory</i>).
cd Blatt1	Wechsel in den Katalog Blatt1 (<i>change directory</i>).
cd ..	Wechsel in den übergeordneten Katalog.
cd	Wechsel in den Heimatkatalog.
pwd	Gibt den aktuellen Katalog aus (<i>print working directory</i>).
cp <Pfad/Dateiname> .	Kopiert eine Datei in das aktuelle Verzeichnis.
javac Collatz.java	Übersetzen der Programmdatei Collatz.java, Ergebnis ist die Datei Collatz.class.
java Collatz	Ausführen des Programms Collatz.

Sie sollten Ihrem Tutor außerdem die folgenden vi-Kommandos vorführen können:

Mode	Kommando	Erklärung
K-Mode	G	Der Cursor wird an das Ende der Datei bewegt (<i>go</i>).
K-Mode	J	Hängt die folgende an die aktuelle Zeile (<i>join</i>).
K-Mode	Y, yy	Speichert die aktuelle Zeile in einem Puffer (<i>yank</i>).
K-Mode	P, p	Gibt den Puffer vor/hinter der aktuellen Position aus (<i>put</i>).
K-Mode	R	Ab der aktuellen Cursorposition überschreiben (<i>replace</i>) (→ I-Mode).
K-Mode	rx	Das Zeichen unter dem Cursor durch das Zeichen <i>x</i> ersetzen (<i>replace</i>).
K-Mode	Ctrl d	Der Bildschirm bewegt sich um eine halbe Seite nach unten (<i>down</i>).
K-Mode	Ctrl u	Der Bildschirm bewegt sich um eine halbe Seite nach oben (<i>up</i>).
K-Mode	Ctrl f	Der Cursor bewegt sich um eine Seite weiter (<i>forward</i>).
K-Mode	Ctrl b	Der Cursor bewegt sich um eine Seite zurück (<i>backward</i>).
K-Mode	Ctrl l	Baut den Bildschirm neu auf.
K-Mode	Ctrl g	Gibt aktuelle Zeilennummer und Dateinamen aus.
K-Mode	ZZ	Den vi verlassen und Datei speichern, falls geändert.
Z-Mode	x	Den vi verlassen und Datei speichern, falls geändert.
Z-Mode	q!	Den vi verlassen ohne zu speichern.

Aufgabe 2.2 (0 Punkte)

Schreiben Sie mit dem *vi* eine Datei `Aufg2_2.txt`, die Sie zur Kommentierung zu Beginn in **alle** Ihre zukünftigen Java-Programmdateien kopieren und mit den jeweils aktuellen Daten füllen. `Aufg2_2.txt` hat folgendes Aussehen:

```
/* ***** AufgX_X.java ***** */
import AlgoTools.IO;

/**
 *   Programmbeschreibung:
 *
 *   @author Ralf Kunze   (rkunze@informatik.uni-osnabrueck.de)
 *   @author Olaf Mueller (olaf@informatik.uni-osnabrueck.de)
 *
 *   @version TT.MM.JJJJ
 */

public class {

    public static void main(String[] argv) {
```

Aufgabe 2.3 (30 Punkte)

Schreiben Sie ein Java-Programm `Aufg2_3.java`, das ganze Zahlen zwischen 0 und 99 „auspricht“, sprich das Zahlwort ausgibt.

Beispiele: Eingabe: 18, Ausgabe: achtzehn; Eingabe: 99, Ausgabe: neunundneunzig.

Das Programm soll genau dann terminieren, wenn eine Zahl eingegeben wird, die nicht im Intervall $[0..99]$ liegt.

Hinweis: Verwenden Sie eine `switch`-Anweisung mit weniger als 100 Fällen. Lassen Sie Ihr Programm die Zahlen ziffernweise „aussprechen“, soweit möglich.

Aufgabe 2.4 (30 Punkte)

Schreiben Sie ein Java-Programm `Aufg2_4.java`, das zunächst eine ganzzahlige Basis $b \in [-10; 10]$ einliest. Anschließend soll Ihr Programm in einer `do/while`-Schleife wiederholt eine positive, ganze Zahl ($a_i < 10$) einlesen, den Wert b^{a_i} berechnen und ausgeben. Durch die Eingabe von 0 für den Exponenten soll Ihr Programm mit der Ausgabe von b^0 terminieren.

Hinweis: $x^0 = 1 \forall x \in \mathbb{Z}$.

Aufgabe 2.5 (40 Punkte)

Schreiben Sie ein Java-Programm `Aufg2_5.java`, das eine positive ganze Zahl ($k < 26$) einliest und für alle Paare $(x, y) \in \{1, \dots, k\} \times \{1, \dots, k\}$ den größten gemeinsamen Teiler (GGT) ausgibt. Benutzen Sie zur Darstellung der GGTs eine Tabelle. Beispiel für $k = 3$:

	1	2	3
1:	1	1	1
2:	1	2	1
3:	1	1	3