

Handout 5: Anweisungen zur Abgabe

Ausgabetermin: 24. April 1998

Form der Abgabe: eine Mail pro Gruppe und pro Blatt

Leider hat sich die Form, die wir zunächst vorschlugen — wir holen die Übungen im Home selbst ab — nicht sehr bewährt, denn es gab allerhand Ausnahmen wo die Übungen zu finden sind, die Rechte stimmten bei vielen nicht und man konnte es nicht lesen etcetc. Auch die Rundmail an alle bzgl. der Mailabgabe war nicht sehr erfolgreich. Deswegen bitten wir nun darum, sich von nun explizit an das folgende Schema zu halten. Der Aufwand ist für jeden einzelnen relativ gering, aber erspart uns ständiges Nachfragen und Suchen, welches bei der Menge der Gruppen ziemlich aufwendig ist.

Die Abgabe der Aufgaben erfolgt **per Mail** an `ms@informatik.uni-kiel.de`, und zwar *pro Gruppe* und pro Übungszettel.

Dabei sollte auf drei Dinge geachtet werden, die im folgenden genauer beschrieben werden:

1. Strukturierung der Lösung
2. Subject der Mail
3. Rumpf der Mail

Struktur der Lösung einer Aufgabe: Die Übungen sollen

- Ein Verzeichnis pro Übung und darunter
- ein Verzeichnis pro Aufgabe enthalten

Damit wir nicht lange suchen müssen, bitte ich, sich an folgender Struktur zu orientieren:

```
uebung1/README
uebung1/aufgabe1/README
uebung1/aufgabe1/Main.java
uebung1/aufgabe1/Myfile.java
uebung1/aufgabe1/file.html
```

Tabelle 1: Struktur

Enthaltene *.class-Files werden von uns nicht ausprobiert, wir schauen uns nur den Quellcode an (und versuchen ihn zu kompilieren) beziehungsweise erläuternden Text.

Subject der Mail: Abgabe: Gruppe x, Blatt y Die Mail an mich soll im Subject folgende drei Informationen in dieser Reihenfolge enthalten:

1. der *wörtliche String*: "Abgabe:"
2. Gruppennummer (als integer),
3. Nummer des Zettels (ebenfalls als integer)

Beispielsweise so:

Subject: Abgabe: Gruppe 17, Blatt 3

Der Grund, warum wir diese strenge Form des Subjekts wollen, ist, daß dies uns eine automatische Sortierung und Ablage der Aufgaben durch den Mailer erlaubt.

Rumpf der Mail = Archiv des Übungsverzeichnisses Aus der Mail soll die oben vorgeschlagene Struktur aus Tabelle 1 von uns *automatisch extrahierbar* sein. Das bedeutet,

eine einzige Abgabemail für einen Übungszettel enthält das Archiv des entsprechenden Verzeichnisses.

Insbesondere auch: keine Abgabe für einzelne Aufgaben oder gar einzelne Java-Files.

Damit es automatisch extrahierbar ist, *nicht* per Hand die Files hintereinanderhängen und mit Texten wie

```
----- Ab hier kommt Aufgabe 5, -----
----- bitte in die Datei File24.java kopieren! -----
```

versehen. Abgesehen davon, daß dies auch bei der Erstellung viel Arbeit macht und fehleranfällig ist, hat eine derartige Abgabe zur Folge, daß wir aus einer vorliegenden Mail per Hand sämtliche Klassen in die entsprechenden Dateien kopieren müssen. Was das bei der steigenden Anzahl der Klassen bei komplexeren Aufgaben für Konsequenzen hat, kann sich jeder ausmalen. Konkret: **Ab einschließlich Aufgabenzettel 3 werden wir keine Abgabe akzeptieren, deren Filestruktur wir erst manuell rekonstruieren müssen.**

Wie macht man ein Archiv? Im einfachsten Falle stellt man so ein *Archiv* (genauer gesagt, ein Tape-archive) her, hier am Beispiel von Übung 1. Nehmen wir dabei an, die Dateien zur Lösung dieser Übung lägen unter `~/java/uebung1/`. Dann mache man:

```
cd ~/java
tar cvf uebung1.tar uebung1
```

Das Programm `tar` sollte auf allen Unix-Maschinen vorhanden sein.¹

Schlußbemerkung: da wir das Einsortieren etc. der Abgabemails weitgehend automatisieren, und da das über das `matchen` der oben angegebenen Ausdrücke im Subject geschieht, bitte derartige Subjects nur zum Zwecke der Abgabe verwenden.

¹Alternativ sind wir auch in der Lage, mit zip-Archiven, tar-mime-attachements und natürlich evtl. zusätzlicher uuencoding umzugehen. Der **entscheidende** Punkt ist, daß die Verzeichnisstruktur der gesamten Aufgabenblattes automatisch extrahierbar sein muß!