Christian-Albrechts-Universität zu Kiel
Sommersemester 2000
Fortgeschrittenenpraktikum (Java)

Handout 5: Form der Abgabe

Ausgabetermin: 19. April 2000

Unser Kurs ist zu groß, als eine Abgabe der Aufgaben in freier Form zu ermöglichen. Der Aufwand, sich an die unteren Hinweise zu richten, ist für jeden einzelnen relativ gering, aber erspart uns Nachfragen und Suchen und Sortieren, welches bei der Menge der Gruppen zu aufwendig ist. Deswegen bitten wir darum, sich für die zunächst kommenden Aufgaben explizit an das folgende Schema zu halten.

Form der Abgabe: eine Mail pro Gruppe und pro Blatt

Die Abgabe der Aufgaben erfolgt **per Mail** an die Adresse `unix01@ap.informatik.uni-kiel.de`, und zwar eine Mail *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 (abhängig natürlich von den jeweiligen Aufgaben) sich von folgender Struktur inspirieren zu lassen:

```
uebung1/Readme
uebung1/aufgabe1/Makefile
uebung1/aufgabe1/Main.java
uebung1/aufgabe1/<file>.java
```

Tabelle 1: Struktur

`*.class`-Files brauchen nicht mit abgegeben werden, wir probieren sie auch nicht aus, sondern schauen uns nur den Quellcode an (und versuchen ihn zu kompilieren) beziehungsweise erläuternden Text.

Es soll bei Programmieraufgaben pro Verzeichnis ein *Makefile* mitgeliefert werden, dessen erstes *target* die Klassen kompiliert (d.h., ein bloßes "`make`" kompiliert die jeweilige Aufgabe. Wer keine Makefiles kennt: (javac nimmt einem auch Arbeit bei der Abhängigkeitsanalyse ab): wir werden eine banales, selbsterklärendes Makefile auf unserer Seite bereitsstellen, das kann man dann in seine Verzeichnisse kopieren (bzw. falls notwendig anpassen.)

Subject der Mail: `Abgabe: Gruppe x, Blatt y` Die Mail an den angegebenen account 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 für den Übungszettel.

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 bzw. müssten. Konkret: Wir werden wir keine Abgabe akzetieren, deren Filestruktur wir erst *manuell* rekonstruieren müssen.

Wie macht man ein Archiv? Im einfachsten Falle stell man so ein *Archiv* (genauer gesagt, ein Tape-achive) 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.

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