

- ◆ Bitte tragen Sie am unteren Blattrand Ihren Namen, Lehrgang und das heutige Datum ein.
- ◆ Dieses Übungsblatt ist zum Ende des jeweiligen Praktikums bei der Praktikumsbetreuung als Nachweis abzugeben. Sie können sich eine Kopie dieses Blattes unter <http://lunabox.de/fh-isny/13.info/> oder unter <http://lunabox.de/13.info> herunterladen.

1. Zur Erinnerung an das letzte Praktikum nochmal das einfache Shell-Script, die verschiedenen Schleifen / Konstrukte / Tests und ein Beispiel:

```
#!/bin/bash
# Kommentare können mit "#" eingeleitet werden
echo "Scriptname + Parameter($# Stueck): $0 $*"
echo "PID der aktuellen Shell: $$"
echo "Parameter1: $1 ; Parameter2: $2 "
```

Beispiel-Aufruf: ./scriptname par1 par2

```
if Ausdruck then Kommando1 else Kommando2 fi
case $var in muster1|m1) Kommando1 ; ; muster2|m2) Kommando2 ; ; *) KommandoX ; ; esac
while Ausdruck do Kommando1 Kommando2 done
until Ausdruck do Kommando1 Kommando2 done
for var in 1 2 3 do Kommando1 Kommando2 $var done
[ $? -eq 0 ] && echo "erfolg"
[ $? -eq 0 ] || echo "fehler"
( $? bezieht sich nur jeweils auf den direkt zuvor ausgeführten Befehl )
```

Einfache arithmetische Operationen in Shell-Scripts lassen sich mit dem Kommando **let** realisieren, z.B. die Inkrementierung einer Variablen könnte man mit **let var=\$var+1** realisieren. Um Variablen per Eingabeaufforderung des Benutzers mit Werten zu versehen kann das Kommando **read** verwendet werden.

Beispiel:

```
#!/bin/bash
DATEI="/etc/passwd"
echo -n "Bitte mehrere beliebige Benutzernamen durch Leerzeichen getrennt eingeben: "
read eingabe
z=0
for name in $eingabe
do let z=$z+1
if [ -r $DATEI ]
then echo -n "$z.: Benutzername $name in $DATEI "
grep $name $DATEI >/dev/null
[ $? -eq 0 ] && echo "gefunden." || echo "leider nicht gefunden."
else echo "$DATEI nicht gefunden."
fi
done
```

3. Schreiben Sie ein ausführbares Shell-Script mit dem Namen **adressverwaltung**, das folgendes leistet:

- Der Dateiname einer Adress-Datenbank¹ soll als einziger Parameter übergeben werden. Fehlt der Parameter, soll das Programm mit einer entsprechenden Fehlermeldung und dem Rückgabewert 1 beendet werden. Existiert die eingegebene Datei noch nicht, soll sie angelegt werden.
- Folgendes Menü soll angezeigt werden:
 - 1.: Adresse eintragen
 - 2.: Adresse suchen
 - 3.: Adressen anzeigen (alle)
 - 4.: Adresse löschen
 - 5.: Programm beenden
 Bitte waehlen: _
- Bei einer fehlerhaften Auswahl soll eine entsprechende Fehlermeldung ausgegeben und wiederholt nach einer Eingabe verlangt werden.
- **Menüpunkt 1:** Name, Vorname, Ort und Telefon sollen abgefragt und durch ";" getrennt an die Adress-Datei angehängt werden (pro Zeile eine komplette Adresse).
- **Menüpunkt 2:** Alle bisher eingegebenen/gespeicherten Adressen sollen nach einem einzugebendem Namen durchsucht, und das Ergebnis im folgenden Format angezeigt werden:
 - <Name>, <Vorname> aus <Ort> hat die Telefonnummer <Telefon>
 Wenn keine Übereinstimmung gefunden werden sollte, soll eine entsprechende Meldung ausgegeben werden. (Denken Sie an cut und die Kommandosubstitution für die Anzeige)
- **Menüpunkt 3:** Alle bisher eingegebenen/gespeicherten Adressen sollen in folgendem Format sortiert angezeigt werden:
 - <Name>, <Vorname> aus <Ort> hat die Telefonnummer <Telefon>
- **Menüpunkt 4:** Alle zu einem einzugebendem Namen gehörenden Adresseinträge sollen aus der Adressdatei entfernt werden.
- Nach Abarbeitung eines Menüpunktes soll anschliessend das Menü wieder angezeigt werden - solange, bis **Menüpunkt 5** gewählt wurde.

Testen Sie das Programm/Script ausführlich und führen es der Praktikumsbetreuung vor.

4. Löschen Sie bitte alle in diesem Praktikum erstellten Dateien und Verzeichnisse. Danke.

1 einfache Text-Datei; pro Zeile eine volle Adresse; Strichpunkt zur Trennung von Name, Vorname, Ort und Telefon

Nachname, Vorname	Lehrgang	Datum	Unterschrift StudentIn	Unterschrift Betreuer