

Name:	Lehrgang: Betriebssysteme	Datum:
Arbeitsblatt Nr.	Shell-Skript-Programmierung: Entscheidungsstrukturen	Seite 1 von 3

## Entscheidungsstrukturen

Innerhalb von Shell-Skripten müssen häufig Entscheidungen getroffen werden - ganz so, wie in höheren Programmiersprachen-, um auf unterschiedliche Situationen reagieren zu können.

In der Shell-Skript-Programmierung existieren - wie auch in höheren Programmiersprachen- zwei verschiedene Entscheidungsstrukturen

1. `if - then - else` und
2. `case ... in`

Gerade für die erste Entscheidungsstruktur sind die bereits besprochenen Vergleichs- und Test-Operationen von großer Bedeutung.

### Ein erstes Beispiel zur `if - then - else` Struktur

Im nachfolgenden Beispiel soll entschieden werden, ob bei Aufruf des Shell-Skriptes ein Parameter übergeben wurde.

```
#!/bin/bash
clear
#ALTERNATIV!!!
#if test $# -eq 0 ; then
if [ $# -eq 0 ]; then
    echo "Es wurde kein Parameter angegeben!"
fi
echo "Habe fertig"
```

Semikolon hinter der Bedingung nicht vergessen!

Hinter dem Schlüsselwort `if` folgt der zu testende Ausdruck. Hier zeigt sich auch die etwas größere Lesbarkeit der obigen Schreibweise `[ $# -eq 0 ]` gegenüber der Schreibweise `test $# -eq 0`, die auch möglich ist.

Wichtig ist, dass direkt hinter der öffnenden bzw. vor der schließenden eckigen Klammer ein Leerzeichen steht, da ansonsten der Syntax verletzt ist!

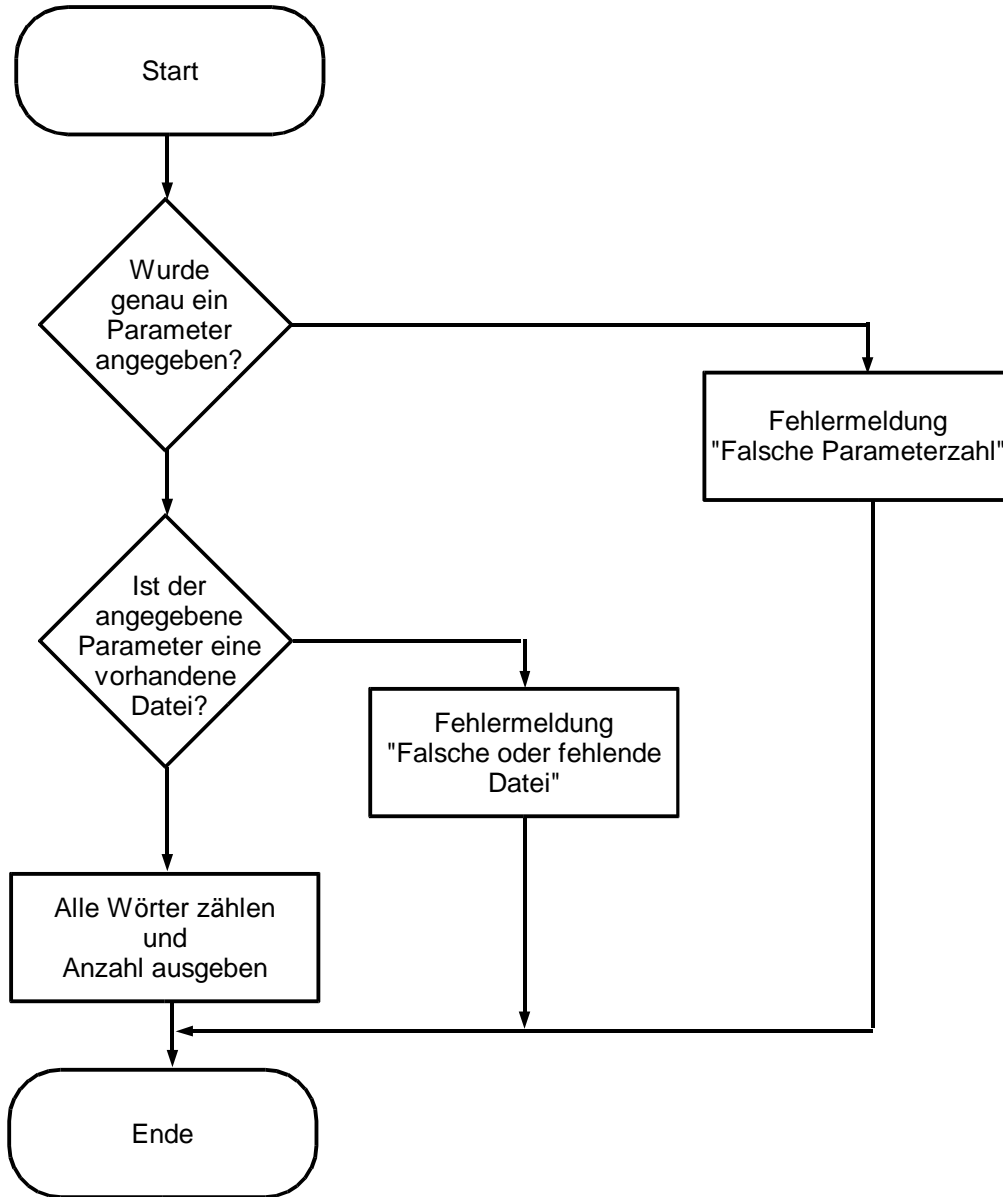
Hinter dem Test-Ausdruck folgt ein Semikolon und das Wort `then`. Wenn die Bedingung erfüllt ist, wird der nachfolgend angegebene Befehl ausgeführt. Hier können auch mehrere Befehle angegeben werden!

Abgeschlossen wird die Struktur durch das Schlüsselwort `fi`. Nach diesem Schlüsselwort können weitere Shell-Skript-Befehle folgen, die nach dieser Struktur abgearbeitet werden.

### Übungen:

1. Erstellen Sie das obige Skript in ihrem Home-Verzeichnis unter dem Namen `if-test` und setzen Sie das Execute-Recht. Testen Sie das Skript aus. Testen Sie auch beide Varianten der Bedingung!
2. Informieren Sie sich entweder im Linux-Buch auf dem WWW-Server oder im Buch UNIX Shell-Programmierung über den `else`-Teil der Entscheidungsstruktur.
3. Kopieren Sie das Shell-Skript im Home auf den Namen `if-test2` und öffnen Sie diese Datei mit einem ASCII-Editor. Ändern Sie das Skript so ab, dass nun bei der Angabe von Parametern folgender Text erscheint:  
 "Es wurden insgesamt *N* Parameter angegeben!", wobei *N* die wirkliche Anzahl der Parameter angibt.  
 "Angegebene Parameter sind: X Y Z...", wobei X Y Z... die angegebenen Parameter
4. Erstellen Sie ein Skript, das die im nachfolgenden PAP dargestellte Aufgabe löst.

**Programmablaufplan zur Übung 4 von Aufgabe 1**



**Hinweis**

Um ein Shell-Skript vorzeitig zu beenden, kann der Befehl `exit` verwendet werden.

Name:	Lehrgang: Betriebssysteme	Datum:
Arbeitsblatt Nr.	Shell-Skript-Programmierung: Entscheidungsstrukturen	Seite 3 von 3

## Fallunterscheidungen mit case

Mit der **case**-Struktur lassen sich mehrere Fälle unterscheiden. Schauen wir uns doch einfach mal ein Beispiel an:

```

#!/bin/bash
clear
case $1 in
a)
    echo "Der Buchstabe a ist was feines"
;;
1)
    echo "Mit der Ziffer 1 geht es gut ab"
;;
*)
    echo "Wow-das kenne ich noch nicht"
;;
esac

```

Beginn der **case**-Struktur  
Als Beispiel enthält `$1` den zu untersuchenden Ausdruck  
Abschluß durch das Wort **in**

Einer der möglichen Fälle. Es wird der Wert angegeben, für den Befehle ausgeführt werden sollen. Gefolgt von einer runden Klammer.  
Der Default-Fall wird durch den Stern angegeben.

Jeder Fall wird durch zwei Semikola abgeschlossen  
(entspricht dem **break;** in C/C++)

Kennzeichnet das Ende

Was macht das Shell-Skript nun? Wird das Skript mit einem Parameter aufgerufen, wird dieser untersucht. Hat er den Wert "1" wird der entsprechende Text ausgegeben; hat er den Wert "a", wird wiederum der entsprechende Text ausgegeben. Wurde irgendwas anderes angegeben, tritt der Default-Fall in Aktion.

### Hinweis

Mit dem Kommando **read** lässt sich eine Eingabe in eine Variable lesen (z.B. **read ANSWER**). Mit der Anweisung **case \$ANSWER in** kann diese Eingabe ausgewertet werden. Hiermit lassen sich hervorragend Menüs erstellen.

### Übungen:

1. Erstellen Sie das obige Shell-Skript unter dem Namen **casetest** in ihrem Home und führen Sie das Skript aus. Testen Sie die verschiedenen Fälle für den ersten Parameter!
2. Erstellen Sie ein Shell-Skript namens **menue**. Das Shell-Skript soll mit **echo** ein Menü mit drei Menüeinträgen (beliebiger Text!) ausgeben. Anschließend soll ein Eintrag ausgewählt werden. In Abhängigkeit des gewählten Menüeintrages soll dann ein entsprechender Text (als Dummy für ansonsten auszuführende Befehle) erscheinen.