

5.8.0 Gambas-Programme mit Optionen und Argumenten

Den Start von Programmen in der Konsole mit der Übergabe von Programm-Optionen und Argumenten kennen Sie aus der Arbeit mit Linux. So startet der erste Befehl das Editor-Programm 'gedit', öffnet die Datei 'set_color.sh' im Home-Verzeichnis und setzt dann den Cursor in der geöffneten Datei in der 5. Zeile an die Position 9:

```
hans@linux:~$ gedit ./set_color.sh +5:9
hans@linux:~$ gnome-terminal --geometry 80x20+100+100 -t 'ROOT-TERMINAL' -e command.sh
```

5.8.0.1 Exkurs – Programme mit Argumenten

Die Funktion des vorgestellten Programms (Bash-Skript) 'set_color.sh' besteht darin, in Abhängigkeit von der Anzahl der übergebenen Argumente entweder den Variablen \$1 bis \$3 über den Befehl 'let' intern 3 (Start-) Werte zuzuweisen und diese danach den Variablen R, G und B oder die Variablen R, G und B mit den Werten der 3 übergebenen Argumente zu belegen. Anschließend wird mit R, G und B ein (Farb-)Wert berechnet und angezeigt.

```
#!/bin/bash
if [ $# -ne 3 ]      # Wenn die Anzahl der übergebenen Argumente ungleich 3 ist, ...
then
  set 220 20 180    # dann RGB-Standardwerte setzen: $1=220(R), $2=20(G) und $3=180(B) ...
  R=$1
  G=$2
  B=$3
else                # sonst R, G und B aus den drei Argumenten setzen
  R=$1
  G=$2
  B=$3
fi
let color=$R*256*256+$G*256+$B # Farbwert berechnen
echo Farbwert = $color
echo Weiter mit ENTER...
read dummy
```

Allgemein gilt:

- Der Wert \$* steht für die Anzahl *aller* Programm-Parameter!
- Das erste Argument \$0 enthält den Namen des gestarteten Programms – das ist Standard.
- \$1 bis \$k bilden eine Argument-Liste der übergebenen Argumente, in der die Reihenfolge auch einer Rangfolge entspricht.
- Der Wert \$# gibt die Anzahl der übergebenen Argumente \$1 ... \$k zurück.

Der Start des Skripts 'set_color.sh' erfolgt zuerst mit genau drei Argumenten und dann nur mit 2 Argumenten. Die Argumente werden nicht auf Validität im Kontext mit RGB-Farbwerten [0..255] geprüft:

```
hans@linux:~$ chmod +x set_color.sh # Das Skript ausführbar machen

hans@linux:~$ ./set_color.sh 10 20 128 # 3 Argumente
Farbwert = 660608
Weiter mit ENTER...

hans@linux:~$ $HOME/set_color.sh 0 128 # 2 Argumente
Farbwert = 14423220 ($1=220 , $2=20 , $3=180)
Weiter mit ENTER...
```

5.8.0.2 Gambas-Programm mit Argumenten

Selbstverständlich können Sie auch einem Gambas-Programm – das vom Gambas-Interpreter ausgeführt werden soll – Argumente übergeben:

```
hans@linux:~$ gbx3 $HOME/ColorSelectBP -- 225 110 60
Anzahl der übergebenen Argumente = 3
0. Argument = ColorSelectBP
1. Argument = 225
2. Argument = 110
3. Argument = 60
```

Als Trennsymbol zwischen dem Gambas-Projekt-Verzeichnis und den Argumenten wird "--" benutzt, wie das in der Hilfe zum Interpreter gbx3 auch angedeutet wird, wenn man <arguments> als Platzhalter für eine durch Leerzeichen getrennte Argumente-Liste interpretiert:

```
hans@linux:~$ gbx3 -h
...
Usage: gbx3 [options] [<project file>] [-- <arguments>]
```

5.8.0.3 Gambas-Programm mit Optionen und Argumenten

In den folgenden zwei Beispielen wird dem Interpreter gbx3 genau eine Option k übergeben. Es wird das Projekt \$HOME/E/OSMapA gestartet und es werden eine Gambas-Programm-Option s sowie genau drei Gambas-Programm-Argumente übergeben:

```
hans@linux:~$ gbx3 -k $HOME/E/OSMapA -- -s n -- 52.7904 11.7533 15
hans@linux:~$ gbx3 -k $HOME/E/OSMapA -- --shape j -- -30 -50 4
```

Mit dem Wert der Option -s oder --shape (j(a) oder n(ein)) festgelegt, ob ein Symbol für die Kennzeichnung des geografischen Start-Koordinaten angezeigt wird oder nicht. Eine ausführliche Beschreibung des Projekts finden Sie im → Kapitel 5.8.2 Projekte.



Abbildung 5.8.0.3.1: GeoMap mit Symbol

Die Verwendung von Optionen für ein Gambas-Programm ist fakultativ. Im nächsten Beispiel wird dem Interpreter ein Gambas-Projekt – ohne Verwendung einer Option – jedoch mit drei Argumenten übergeben:

```
hans@linux:~$ gbx3 $HOME/ColorSelectBP -- 225 110 60
```