

5.8.1 Optionen und Argumente bei Gambas-Programmen

Für die Arbeit mit Optionen und Argumenten bei Gambas-Programmen, die dem Gambas-Interpreter `gbx3` übergeben werden, stellt Gambas spezielle Klasseneigenschaften, eine Klasse und eine Komponente bereit:

- Einsatz der drei Nur-Lese-Eigenschaften (`All`, `Count`, `Max`) der Klasse `Args` (`gb`)
- Nutzung der statischen Eigenschaft `Application.Args` (Daten-Typ `Args`) der Klasse `Application`
- Verwendung der Klasse `Args` in der Komponente `gb.args` mit fünf Methoden, um einen Parser für Programm-Optionen und Programm-Argumente zu implementieren

5.8.1.1 Klasse `Args`

Diese Klasse `Args` (`gb`) implementiert ein Array, das die an ein Gambas-Programm auf der Konsole übergebenen Argumente enthält. Diese Klasse verhält sich wie ein statisches Nur-Lese-Array. Die folgende For-Each-Kontrollstruktur zum Beispiel durchläuft alle Argumente der Argument-Liste:

```
Dim sElement AS String
For Each sElement In Args
  Print sElement
Next
```

Den Programm-Namen ermitteln Sie zum Beispiel über das erste Element im Array `Args` (Index = 0):

```
Dim sArgumentPN As String
sArgumentPN = Args[0] ' Programm-Name
```

Die native Klasse `Args` der Komponente `gb` verfügt nur über drei Eigenschaften:

Eigenschaft	Datentyp	Beschreibung
<code>All</code>	<code>String[]</code>	Gibt eine Kopie <u>aller</u> Argumente als <code>String</code> -Array zurück.
<code>Count</code>	<code>Integer</code>	Gibt die Anzahl der in der Konsole übergebenen Argumente zurück.
<code>Max</code>	<code>Integer</code>	Gibt <code>Args.Count - 1</code> zurück.

Tabelle 5.8.1.1.1 : Eigenschaften der Klasse `Args` (`gb`)

Mit diesen Anweisungen

```
Print "Anzahl der Array-Elemente = "; Args.Count
Print "Liste aller Argumente:"
Print Args.All.Join(" | ")
```

sehen Sie nach dem Programm-Start diese Ausgaben in einer Konsole:

```
hans@linux:~$ gbx3 $HOME/E/ColorSelectA2 -- 50 120 30
Anzahl der Array-Elemente = 4
Liste aller Argumente:
ColorSelectA2 | 50 | 120 | 30
hans@linux:~$
```

5.8.1.2 Ausgewählte Eigenschaft der Klasse `Application` der Komponente `gb`

Von der Klasse `Application` interessiert im Zusammenhang mit Argumenten für Gambas-Programme nur die Eigenschaft `Application.Args`, die ein Array vom Daten-Typ `Args` (→ Kapitel 5.8.1.1) zurück gibt, das nur ausgelesen werden kann.

5.8.1.3 Komponente `Args`

Die Komponente `gb.args` ermöglicht Ihnen:

- das Extrahieren von Optionen,
- den Zugriff auf alle übergebenen Argumente in einem `String`-Array und
- übernimmt die automatische Verarbeitung der Optionen `--version` (`-V`) und `--help` (`-h`).

Die Klasse *Args* verfügt über diese fünf Methoden:

Methoden	Beschreibung
Static Sub Begin ([Usage As String])	Startet die Programm-Argument-Analyse. <i>Usage</i> ist ein optionaler Parameter, der einen Programm-Hilfetext repräsentiert.
Static Function End () As String[]	Beendet die Programm-Argument-Analyse und liefert alle Argumente, die <u>nicht</u> als Option interpretiert werden können, in einem String-Array.
Static Function Get (ShortName As String, LongName As String [, Description As String, ArgName As String]) As String	Die Funktion definiert eine Option vom Typ String, deren Parameter den Namen der Option (Kurz- und Lang-Format) und optional eine Beschreibung der Option enthält.
Static Function GetFloat (ShortName As String, LongName As String [, Description As String, ArgName As String, Default As Float]) As Float	Die Funktion definiert eine Option vom Typ Float, deren Parameter den Namen der Option (Kurz- und Lang-Format) und optional eine Beschreibung der Option sowie einen Vorgabewert enthält.
Static Function GetInteger (ShortName As String, LongName As String [, Description As String, ArgName As String, Default As Integer]) As Integer	Die Funktion definiert eine Option vom Typ Integer, deren Parameter den Namen der Option (Kurz- und Lang-Format) und optional eine Beschreibung der Option sowie einen Vorgabewert enthält.
Static Function Has (ShortName As String, LongName As String [, Description As String]) As Boolean	Die Funktion definiert eine Option <u>ohne</u> Parameter.

Tabelle 5.8.1.3.1 : Methoden der Klasse *Args*

Hinweise:

- Mit der *Begin(..)*-Methode starten Sie die Analyse der Optionen und beenden diese Analyse mit der *End()*-Methode.
- Wenn der optionale Parameter (Daten-Typ String) in der *Begin([hilfetext])*-Methode angegeben wurde, dann sollte er die Programm-Syntax und weitere Programm-Hilfen beschreiben.
- Die Programm-Hilfe wird mit der Option *--help* oder *-h* (Kurzform) aufgerufen. Fehlt die optionale Hilfe im Methodenaufruf, so wird ein Standard-Text ausgegeben.
- Die Versionsnummer des Gambas-Projekts erhalten Sie über die Option *-V* oder *--version*.
- Wenn *gb.args* einen Fehler in der übergebenen Kommando-Zeile findet, dann wird das Gambas-Programm mit dem Exit-Code 1 beendet. Entdeckt *gb.args* zum Beispiel in der Kommandozeile eine nicht definierte Option *'-x'*, dann ist das eine unbekannte Option. Eine unbekannte Option aber ist ein Fehler und *gb.args* beendet selbständig den Gambas-Prozess mit Exit-Code 1. Gut zu wissen, dass der Error-Code den Wert 1 hat, weil Sie dann gute Möglichkeiten haben zu ermitteln – beispielsweise für ein Shell-Skript – warum der Gambas-Prozess beendet wurde.

Es gilt – in Abhängigkeit von den Funktionsparametern der in der oberen Tabelle aufgeführten Funktionen – folgende Beschreibung:

- **ShortName** ist die Kurzbezeichnung der Option (ohne den kennzeichnenden) Bindestrich.
- **LongName** ist der vollständige Name der Option (ohne die genau zwei kennzeichnenden Bindestriche).
- **Description** ist ein optionaler Text, der die Option beschreibt. Der Text wird angezeigt, wenn die Programm-Hilfe über *-h* oder *--help* angefordert wird.
- **ArgName** ist eine optionale Kurzbeschreibung der Argumente der Option. Der Text wird angezeigt, wenn die Programm-Hilfe über *-h* oder *--help* angefordert wird.
- **Default** ist der Standardwert, der zurückgegeben wird, wenn die Option *ohne* Fließkomma-Zahl oder *ohne* ganze Zahl notiert wurde.

5.8.1.4 Beispiel

Der folgende Quelltext ist ein Ausschnitt aus dem Quelltext eines Projekts, das im Kapitel 5.8.3 vorgestellt wird:

```
[1] Public sShape As String
```

```

[2] Public aArguments As Variant[]
[3]
[4] Public Sub Form_Open()
[5]     FMain.Center
[6]     FMain.Caption = "Gmbas-Programm '" & Application.Name & "' mit 1 Option und 3 Argumenten"
[7]
[8]     Args.Begin(SetHelpContent())
[9]     ' Deklaration einer Gmbas-Programm-Option: shape
[10]     sShape = Args.Get("s", "shape", "Startpunkt? (j)a oder (n)ein", "j|n")
[11]     aArguments = Args.End() ' Args.End() = Array der 'echten' Programm-Argumente
[12]
[13] ***
[14]
[15] If Not sShape And aArguments.Count = 0 Then
[16]     SetDefaultArguments()
[17] Else
[18]     SetOptionAndArguments() ' Inklusive Analyse der Option und der 3 Argumente
[19] Endif
[20]
[21] ShowMap()
[22] End ' Form_Open()
[23]
[24] Private Function SetHelpContent() As String
[25]     Dim sUsage As String
[26]
[27]     sUsage = gb.NewLine
[28]     sUsage &= "Führt ein Gmbas-Programm aus, um Karten von OpenStreetMap anzuzeigen" & gb.NewLine
[29]     sUsage &= gb.NewLine
[30]     sUsage &= "Aufruf: " & "gbx3 ProjektPfad " & "-- [Optionen] [-- <Breite> <Länge> <Zoom>]"
[31]     sUsage &= "Usage : " & "gbx3 ProjectPath " & "-- [options] [-- <latitude> <longitude> <zoom>]"
[32]     sUsage &= gb.NewLine
[33]     sUsage &= "Syntax: " & gb.NewLine
[34]     sUsage &= "gbx3 ProjektPfad -- -s [j|n] [-- LAT LON ZOOM] (ohne °Einheit)" & gb.NewLine
[35]     sUsage &= "gbx3 ProjektPfad -- --shape [j|n] [-- LAT LON ZOOM]" & gb.NewLine
[36]     sUsage &= gb.NewLine
[37]     sUsage &= "Format für die geografischen Koordinaten Breite (LAT) und Länge (LON)" & gb.NewLine
[38]     sUsage &= "Dezimalgrad -90° < Breite° " & String.Chr(8804) & " +90°"
[39]     sUsage &= " | -180° < Länge° " & String.Chr(8804) & " +180°" & gb.NewLine
[40]     sUsage &= gb.NewLine
[41]     sUsage &= "Beispiel 1: gbx3 $HOME/Map -- -s j -- 52,787969 11,752522 15" & gb.NewLine
[42]     sUsage &= "Example 2 : gbx3 $HOME/Map -- --shape n -- 0.02 -90.01 7" & gb.NewLine & gb.NewLine
[43]     sUsage &= "Der Zoom-Faktor liegt im Intervall: 1 " & String.Chr(8804) & " ZOOM " & \
[44]         String.Chr(8804) & " 18 ( ZOOM " & String.Chr(8714) & " Integer )."
[45]
[46]     Return sUsage
[47]
[48] End ' Function SetUsage()

```

Zur Demonstration ist an der Stelle ******* dieser Quelltext-Abschnitt eingefügt worden:

```

' Kontrolle:
Print "Wert der Option 'Shape' = "; sShape
For i = 0 To Args.End().Max
    Print i + 1; ". Argument: Wert = "; Args.End()[i]
Next
Print "-----"
For Each sElement In aArguments
    Print "Wert des Arguments = "; sElement
Next

```

Kommentar:

- In der Zeile 8 startet die Programm-Argument-Analyse. Der Prozedur ist als Parameter der Hilfe-Text aus den Zeilen 24 bis 48 übergeben worden.
- Wie eine Option (vollständig) definiert wird, die zwei Werte vom Daten-Typ String besitzt, sehen Sie in der Zeile 10.
- Die End()-Methode beendet die in der Zeile 8 gestartete Analyse. Danach enthält das String-Array Args.End() alle echten Programm-Argumente.

Interessant ist die Syntax der folgenden Zeile:

```
hans@linux:~$ gbx3 $HOME/E/OSMapA -- --shape j -- 52.7904 1.0200 9
```

- Der Interpreter gbx3 erwartet den Pfad zum Projekt-Verzeichnis; hier \$HOME/E/OSMapA.
- Der rote Doppelstrich teilt dem Interpreter mit, dass alles folgende dem Gmbas-Prozess unverändert übergeben werden soll. Mit diesen Daten operiert gb.args.
- Nach mindestens einem Leerzeichen folgt unmittelbar nach einem Doppelstrich der Name der

Option, dem nach mindestens einem Leerzeichen der Wert der Option folgt.

- Nach der Liste der Optionen folgen die blauen Doppelstriche, die den Beginn der Aufzählung der Argumente kennzeichnen. Beachten Sie auch hier als Trennzeichen (mindestens) ein Leerzeichen.

Das sind die (alternativen) Aufrufe und die Ausgabe in der Konsole:

```
hans@linux:~$ gbx3 $HOME/E/OSMapA -- -s j -- 52.7904 1.0200 9
hans@linux:~$ gbx3 $HOME/E/OSMapA -- --shape j -- "52.7904" "1.0200" "9" # Alternative
Wert der Option 'Shape' = j
1. Argument: Wert = 52.7904
2. Argument: Wert = 1.0200
3. Argument: Wert = 9
-----
Wert des Arguments = 52.7904
Wert des Arguments = 1.0200
Wert des Arguments = 9
hans@linux:~$
```

So rufen Sie die Programm-Hilfe auf:

```
hans@linux:~$ gbx3 $HOME/E/OSMapA -- --help
hans@linux:~$ gbx3 $HOME/E/OSMapA -- -h

Führt ein Gmbas-Programm aus, um Karten von OpenStreetMap anzuzeigen

Aufruf: gbx3 ProjektPfad -- [Optionen] [-- <Breite> <Länge> <Zoom>]
Usage : gbx3 ProjectPath -- [options] [-- <latitude> <longitude> <zoom>]

Syntax:
gbx3 ProjektPfad -- -s [j|n] [-- LAT LON ZOOM] (ohne °Einheit)
gbx3 ProjektPfad -- --shape [j|n] [-- LAT LON ZOOM]

Format für die geografischen Koordinaten Breite (LAT) und Länge (LON)
Dezimalgrad -90° < Breite° ≤ +90° | -180° < Länge° ≤ +180°

Beispiel 1: gbx3 $HOME/Map -- -s j -- 52,787969 11,752522 15
Example 2 : gbx3 $HOME/Map -- --shape n -- 0.02 -90.01 7

Der Zoom-Faktor liegt im Intervall: 1 ≤ ZOOM ≤ 18 ( ZOOM ∈ Integer ).

Options:
-s --shape <j|n>           StartPunkt? (j)a oder (n)ein
-V --version              Display version
-h --help                 Display this help
hans@linux:~$
```

Mehr war nach diesem Aufruf als Ausgabe auch nicht zu erwarten:

```
hans@linux:~$ gbx3 $HOME/E/OSMapA -- --version
0.1.28
hans@linux:~$
```

Auf zwei Besonderheiten bei der Eingabe von Optionen und Programm-Argumenten soll mit Beispielen kurz eingegangen werden.

Beispiel 1:

Da alle Argumente als Zeichenkette eingelesen werden, ist es bei bestimmten Zeichen in der Konsole notwendig, diese innerhalb der Zeichenkette zu maskieren:

```
hans@linux:~$ gbx3 $HOME/E/OSMapB -- --modus s -- 52°47'20''N 11°45'36''E 13
hans@linux:~$ gbx3 $HOME/E/OSMapB -- --modus s -- "52°47'20''N" "11°45'36''E" "13"
```

Beispiel 2:

Wenn eine Option definiert wurde, dann müssen Sie die zwei (roten) Doppelstriche vor der Liste der Optionen eingeben – auch dann, wenn Sie die fehlende Eingabe einer Option im Quelltext explizit behandeln:

```
hans@linux:~$ gbx3 $HOME/E/OSMapA -- -- 52.7904 1.0200 9
Die Option 'shape' wird von 'nicht-definiert' auf j(a) gesetzt.
hans@linux:~$
```