

FlowProtocol

Wolfgang Maier

23. Januar 2022

Inhaltsverzeichnis

1	Allgemeine Beschreibung	2
1.1	Grundidee	2
1.2	Anwendungsbereiche	2
1.3	Konfiguration	3
1.4	Technische Ergänzungen	4
2	Befehlsreferenz	4
2.1	Grundlagen des Dateiaufbaus	4
2.2	Beschreibungstexte	6
2.3	Kommentare	6
2.4	Frage mit Antwortmöglichkeit	6
2.5	Sequenzen	7
2.6	Verschachtelungen	8
2.7	Gruppierungen	9
2.8	Unterpunkte	10
2.9	Wiederholungen	11
2.10	Implikationen	12
2.11	Ausführungen	12
2.12	Funktionen	13
2.13	Parametrisiert Funktionen	14
2.14	Variablen	16

1 Allgemeine Beschreibung

1.1 Grundidee

FlowProtocol ist eine Anwendung, mit der man Aufgaben-, bzw. Prüflisten zu ausgewählten Themengebieten anhand einfacher Kontrollfragen erstellen kann. Der Name leitet sich ab aus Flow für „Flussdiagramm“ und „Protokoll“ und steht für das Grundkonzept der Anwendung, anhand eines verzweigten Entscheidungsbaums ein Protokoll zusammenzustellen. Die Anwendung funktioniert dabei wie folgt: Nach Auswahl einer Vorlage wird eine Reihe von Multiple-Choice-Fragen gestellt, mit denen die relevanten Aspekte eines Themas Schritt für Schritt eingegrenzt werden, und aus denen am Ende eine genau abgestimmte Auflistung von Punkten zusammengestellt wird, die beispielsweise als Protokoll verwendet werden können.

Die Grundidee der Anwendung besteht darin, ein Medium zu bieten, um Spezialwissen ohne großen Aufwand zu erfassen, um es in sehr einfacher Form für andere bereitzustellen, ohne dass es dafür vermittelt und erworben werden muss. Die Erstellung der Vorlagen kann und soll direkt durch die Menschen erfolgen, die über das jeweilige Wissen verfügen, und von einer breiten Menge an Nutzern angewendet werden.

1.2 Anwendungsbereiche

Die Anwendungsbereiche für diese Anwendung sind sehr vielfältig. Die ursprüngliche Intention war die Aufstellung von Prüflisten für wiederkehrende komplexe Tätigkeiten, die in vielen einzelnen Aspekten variieren können, so dass jeweils nur bestimmte Auswahl von Prüfpunkten relevant ist. Eine Gesamtprüfliste, bei der immer ein Großteil der Punkte hätte ignoriert werden müssen, wäre sowohl aufwendig, als auch fehlerträchtig, und daher nicht effektiv. *FlowProtocol* kann hier durch sein Fragekonzept die für jeden Einzelfall relevanten Aspekte herausfiltern, und auf diese Weise sogar beliebig stark ins Detail gehen. Gleichzeitig hilft dem Anwender die Beantwortung der Fragen auch, sich gedanklich ausführlich mit seiner Aufgabe auseinanderzusetzen. Durch die Vermeidung offener Fragen wird gleichzeitig das Risiko minimiert, dass Dinge vergessen werden. Das Ergebnis ist ein zu 100 % relevantes und vollständiges Protokoll.

Ein besonders positiver Nebeneffekt dieser Vorgehensweise besteht darin, dass durch einfaches Auswählen von Antworten am Ende ein Ergebnisdokument erstellt wird, das nicht mehr manuell erstellt werden muss. Das An-

wendungsgebiet dehnt sich damit auf alle Arten von Dokumenten aus, die sich aus einer Aufzählung von Textbausteinen zusammensetzen, deren Zusammenstellung über eine Reihe iterierter Fragen erfolgen kann. Dies umfasst beispielsweise Bestandsaufnahmen, Analyseprotokolle, Zusammenfassungen, Aufgabenlisten und vieles andere mehr. Für kleinere Systeme lassen sich sogar mit überschaubarem Aufwand Vorlagen erstellen, die nicht nur das Ergebnis einer Analyse zusammenfassen, sondern ergänzend dazu auch schon Lösungsansätze und mögliche weitere Schritte. Man kann so vom Prinzip her ein kleines Expertensystem erstellen, das wertvolles Spezialwissen eines Experten in sehr einfacher Form für andere Mitarbeiter verfügbar macht, ähnlich wie ein Chatbot. Das schont wertvolle Unternehmensressourcen und beugt Verfügbarkeitsengpässen vor. Auch kleinere personalisierte Umfragen lassen sich damit durchführen, wenn man die Rücksendung des Ergebnisses instruiert. Aber insbesondere dort, wo man sich im Rahmen von Normen wie ISO9001 oder ISO27001 dazu verpflichtet hat, klare Qualitätsstandards im Bereich der Unternehmensprozesse einzuhalten, kann *FlowProtocol* gute Unterstützung leisten!

1.3 Konfiguration

FlowProtocol ist bewusst einfach gehalten und soll es auch bleiben. Als Web-Anwendung kann der Dienst innerhalb einer Einrichtung zentral zur Verfügung gestellt werden. Die einzige notwendige Konfiguration ist die Angabe eines serverseitig verfügbaren Vorlagenordners, in dem sich die Vorlagen befinden (einstellbar über die Eigenschaft `TemplatePath` in der Datei `appsettings.json`). Es werden weder Datenbank noch zusätzliche Dienste benötigt. Die Vorlage-Dateien sind normale Textdateien (in UTF-8-Codierung), die die eigentliche Logik enthalten, und die über die Anwendung ausgeführt werden. Die Syntax der Vorlage-Dateien ist ebenfalls sehr einfach und bewusst für die manuelle Erstellung in einem Editor vorgesehen. Der Aufbau ist zeilenbasiert und verwendet Einrückung zur Abbildung der Verschachtelung. Die Erstellung einer ersten eigenen Vorlage ist innerhalb von 2 Minuten möglich. Eine ausführliche Beschreibung des Sprachumfangs wird in der Befehlsreferenz gegeben. Das Ergebnis einer Bearbeitung wird am Ende auf dem Bildschirm ausgegeben, von wo aus es in die Zwischenablage übernommen werden kann. Tatsächlich wird in den meisten Fällen eine Weiterverarbeitung in einem anderen System (E-Mail, CRM, Projektverwaltung) erfolgen, wo noch Begriffe ergänzt und Prozesse angestoßen werden können und eine revisionssichere Verwaltung möglich ist.

Die Organisation der Vorlagen erfolgt auf zwei Ebenen. Auf unterster Ebene

ist eine Unterteilung in Anwendergruppen vorgesehen, die jeweils ihre eigenen Vorlagen verwenden. In einem Unternehmen oder einer Einrichtung wird sich diese Aufteilung normalerweise am Mitarbeiterorganigramm orientieren. Ein Beispiel wäre die Aufteilung in Vertrieb, Marketing, Entwicklung, Administration. Jeder Ordner im Vorlagenordner wird in *FlowProtocol* als Anwendergruppe auf der Seite Anwendergruppen angezeigt. Innerhalb dieser Ordner kann man die Vorlagen entweder direkt ablegen oder in weitere Unterordner unterteilen, um fachliche Gruppen zu bilden. Der Pfad einer Vorlage innerhalb dieser Struktur findet sich auch in der URL wieder, die innerhalb der Anwendung erzeugt wird, und kann so auch direkt als Link in einer E-Mail, einem Vorgang, im Wiki oder an einer anderen Stelle im Intranet bereitgestellt werden.

1.4 Technische Ergänzungen

FlowProtocol arbeitet vollständig zustandslos in dem Sinne, dass bei der Benutzung keinerlei Daten durch die Anwendung gespeichert werden. Es gibt weder eine Datenbank, noch eine Benutzerverwaltung und der Zugriff auf das Dateisystem erfolgt nur lesend. Die Verwaltung der gegebenen Antworten erfolgt vollständig in der URL, was die Möglichkeit bietet, zurückzuspringen oder einen Zwischenstand als Lesezeichen zu speichern oder zu versenden, allerdings werden Aufrufe von URLs in Unternehmen teilweise durch die IT-Infrastruktur protokolliert, sodass eine Verarbeitung schützenswerter Daten auf jeden Fall dahingehend betrachtet werden sollte.

FlowProtocol steht unter der MIT Lizenz und ist unter

`https://github.com/maier-san/FlowProtocol`

frei verfügbar.

Viel Freude beim Erstellen von Vorlagen und deren Anwendung!

2 Befehlsreferenz

2.1 Grundlagen des Dateiaufbaus

Vorlagendateien können in einem beliebigen Texteditor erstellt werden. Die sehr einfache Sprache verlangt weder Syntax-Hervorhebung, noch Auto-Ergänzung, wobei letzteres bei dem von mir für das Windows-Betriebssystem empfohlenen Editor Notepad++ schon in einem sehr komfortablen Maße gegeben ist.

Als Dateieindung für eine Vorlagendatei muss „.qfp“ (Quick Flow Protocol) gewählt werden, damit die Datei über die Anwendung aufrufbar ist.

Bei der Verwendung von Text mit Umlauten sollte darauf geachtet werden, dass diese innerhalb der Anwendung korrekt dargestellt werden. In meiner Umgebung war dies mit der Kodierung UTF-8 der Fall.

Die Verschachtelung der Struktur wird durch Einrückung abgebildet. Hier kann wahlweise mit Tabulator- oder Leerzeichen gearbeitet werden, jedoch sollte dies einheitlich geschehen, da die Struktur bei einer Durchmischung unter Umständen nicht mehr korrekt aufgelöst wird. Die Anwendung ersetzt Tabulatorzeichen durch vier Leerzeichen, unabhängig von ihrer Position. Die Interpretation einer Datei erfolgt zeilenweise, d.h. jeder Zeilenumbruch schließt einen Befehl ab. Der Einsatz von Zeilenumbrüchen zur Formatierung ist damit nicht möglich. Leerzeilen werden bei der Verarbeitung ignoriert und können in beliebiger Menge eingefügt werden.

Ein erstes Beispiel

```
/// Formatierte Hallo-Welt-Ausgabe
// Anwendungsbeispiel für die Befehlsreferenz
?F1: Wie soll "Hallo Welt" ausgegeben werden?
    #a1: Ganz normal
        >> Hallo Welt
    #a2: In Großbuchstaben
        >> HALLO WELT
    #a3: Rückwärts
        >> tleW ollaH
```

Diese Vorlage führt zu folgender Ausgabe:

Vorlage Demo, 01 HalloWelt

Formatierte Hallo-Welt-Ausgabe

Wie soll "Hallo Welt" ausgegeben werden?

- ☐ Ganz normal
- ☐ In Großbuchstaben
- ☐ Rückwärts

Der Anwender kann und eine der drei Optionen wählen, und die Bearbeitung mit der Weiter-Taste fortsetzen, bzw. in diesem Fall abschließen.

Wenn man zum Beispiel die zweite Option „In Großbuchstaben“ , so sieht das Ergebnis wie folgt aus:

Vorlage Demo, 01 HalloWelt

1 HALLO WELT

2.2 Beschreibungstexte

Syntax:

```
/// <Beschreibungstext>
```

Diese Angabe ist nur in der äußersten Ebene einer Vorlagendatei wirksam. Es können mehrere Zeilen dieser Art angegeben werden.

Der angegebene Beschreibungstext wird bei der Darstellung von Fragen im Kopfbereich ausgegeben. Damit lässt sich einem oder mehreren Sätzen kurz beschreiben, wozu die Vorlage dient, zu welchem Zweck sie eingesetzt werden kann.

2.3 Kommentare

Syntax:

```
// <Kommentar>
```

Diese Kommentare werden bei der Verarbeitung durch die Anwendung vollständig ignoriert und dienen ausschließlich zur Kommentierung des Vorlagencodes.

2.4 Frage mit Antwortmöglichkeit

Syntax:

```
?<Frageschlüssel>: <Fragetext>
    #<Schlüssel Antwortmöglichkeit 1>: <Text Antwortmöglichkeit 1>
        >> <Text Ausgabe 1>
    #<Schlüssel Antwortmöglichkeit 2>: <Text Antwortmöglichkeit 2>
        >> <Text Ausgabe 2>
    . . .
```

Das Kernelement der Vorlagen sind Fragen. Diese bestehen aus einem Frage-schlüssel und einem Frage Text. Die Schlüssel für Fragen und Antwortmöglichkeiten werden verwendet, um die gegebenen Antworten innerhalb der URL zu verwalten. Sie dürfen nur aus Buchstaben und Zahlen bestehen und sollten möglichst kurz gewählt werden. Die Schlüssel der Fragen müssen über die komplette Vorlage eindeutig sein, die der Antworten nur innerhalb einer Frage.

An der Oberfläche dargestellt wird nur der Fragetext und die Texte der Antwortmöglichkeiten.

Durch `>>` wird eine Ausgabe erzeugt, wenn die entsprechende Antwortmöglichkeit gewählt wurde. Die Ausgaben werden als nummerierte Liste zu einem Ergebnis zusammengefasst.

2.5 Sequenzen

Mehrere Fragen können hintereinander aufgelistet werden. Diese werden standardmäßig untereinander auf einer Seite aufgelistet und können zusammen bearbeitet werden.

Beispiel:

```
Formatierte Hallo-Welt-Ausgabe
// Anwendungsbeispiel für die Befehlsreferenz
?F1: Wie soll "Hallo Welt" ausgegeben werden?
    #a1: Ganz normal
        >> Hallo Welt
    #a2: In Großbuchstaben
        >> HALLO WELT
    #a3: Rückwärts
        >> tleW ollaH
?F2: Welcher Zusatz soll ergänzt werden?
    #z1: "Wie geht es dir?"
        >> Wie geht es dir?
    #z2: "Ich grüße dich!"
        >> Ich grüße dich!
```

Wie soll "Hallo Welt" ausgegeben werden?

- ☐ Ganz normal
- ☐ In Großbuchstaben
- ☐ Rückwärts

Welcher Zusatz soll ergänzt werden?

- ☐ "Wie geht es dir?"
- ☐ "Ich grüße dich"

Die Benutzerführung erlaubt es, auch nur einen Teil der angezeigten Fragen zu beantworten und dann die Weiter-Schaltfläche zu betätigen. In diesem Fall werden die nicht beantworteten Fragen einfach auf der Folgeseite erneut angezeigt.

2.6 Verschachtelungen

Beispiel:

Formatierte Hallo-Welt-Ausgabe

// Anwendungsbeispiel für die Befehlsreferenz

?F1: Wie soll "Hallo Welt" ausgegeben werden?

#a1: Ganz normal

>> Hallo Welt

#a2: In Großbuchstaben

?F1a: Soll "Hallo Welt" zusätzlich rückwärts ausgegeben werden?

#j: Ja

>> TLEW OLLAH

#n: Nein

>> HALLO WELT

#a3: Rückwärts

>> tleW ollaH

Die Darstellung ist zunächst identisch mit der des Ausgangsbeispiels. Die innere Frage wird erst und nur dann angezeigt, wenn in der ersten Ebene die Option „In Großbuchstaben“ gewählt wird.

Soll "Hallo Welt" zusätzlich rückwärts ausgegeben werden?

- ☐ Ja
- ☐ Nein

2.7 Gruppierungen

Das Ergebnis einer Bearbeitung kann in vielen Fällen mehr als eine Liste sein. Eine Vorlage zur Analyse kann beispielsweise zum einen den ist-Stand erfassen, mögliche Lösungsansätze formulieren und die noch ausstehenden Analyse-Maßnahmen auflisten. Zu diesem Zweck können die Ausgaben in einer sehr einfachen Form kopiert werden.

Syntax:

```
>> <Gruppe> >> <Text Ausgabe>
```

Beispiel:

```
Formatierte Hallo-Welt-Ausgabe
// Anwendungsbeispiel für die Befehlsreferenz
?F1: Wie soll "Hallo Welt" ausgegeben werden?
#a1: Ganz normal
    >> Deine Ausgabe >> Hallo Welt
    >> Die Alternativen >> HALLO WELT
    >> Die Alternativen >> tleW ollaH
#a2: In Großbuchstaben
    >> Deine Ausgabe >> HALLO WELT
    >> Die Alternativen >> Hallo Welt
    >> Die Alternativen >> tleW ollaH
#a3: Rückwärts
    >> Deine Ausgabe >> tleW ollaH
    >> Die Alternativen >> Hallo Welt
    >> Die Alternativen >> HALLO WELT
```

Wählt man die Antwortmöglichkeit „In Großbuchstaben“, so erhält man die folgende Ausgabe:

```
Ergebnisliste (Demo, 05 Gruppierungen)
Deine Ausgabe
    1 HALLO WELT
Die Alternativen
    1 Hallo Welt
    2 tleW ollaH
```

2.8 Unterpunkte

Ausgaben können je nach Vorlage selbst wieder Tätigkeiten oder Aufgaben beschreiben, die sich aus verschiedenen Schritten zusammensetzen, oder bei denen mehrere Teilaspekte zu berücksichtigen sind. Diese lassen sich in Form von Unterpunkten auflisten. Ein Unterpunkt, der mit der Protokollkennung „https://“ beginnt und als Link erkannt wird, wird dabei in der Ausgabe als ausführbarer Link dargestellt.

Syntax:

```
>> <Ausgabe Text>
> <Unterpunkt Text 1>
    > <Unterpunkt Text 2>
    . . .
```

Beispiel:

```
Formatierte Hallo-Welt-Ausgabe
// Anwendungsbeispiel für die Befehlsreferenz
?F1: Wie soll "Hallo Welt" ausgegeben werden?
    #a1: Ganz normal
        >> Hallo Welt
    #a2: In Großbuchstaben
        >> HALLO WELT
            > Mehr Infos findest du hier:
            > https://github.com/maier-san/FlowProtocol
    #a3: Rückwärts
        >> tleW ollaH
```

Wählt man die Antwortmöglichkeit „In Großbuchstaben“, so erhält man die folgende Ausgabe:

Ergebnisliste (Demo, 06 Unterpunkte)

1 HALLO WELT

- Mehr Infos findest du hier:
- <https://github.com/maier-san/FlowProtocol>

2.9 Wiederholungen

Eine Frage kann an anderer Stelle wiederholt gestellt werden. Ausschlaggebend dafür ist, dass der Schlüssel der Frage identisch mit einem bereits verwendeten Schlüssel ist. Ist die Antwort auf die Frage bei der Bearbeitung zu diesem Zeitpunkt bereits gegeben, so wird die Fragestellung nicht mehr angezeigt und direkt der jeweilige Antwortzweig verarbeitet. Wenn das aufgrund der Anordnung zwingend der Fall ist, können die Texte abgekürzt werden, da sie ja nicht ausgegeben werden. In diesem Fall empfiehlt es sich zur Verbesserung der Übersichtlichkeit, in den Texten kenntlich zu machen, dass es sich um eine Wiederholung handelt.

Beispiel:

```
Formatierte Hallo-Welt-Ausgabe
// Anwendungsbeispiel für die Befehlsreferenz
?F1: Wie soll "Hallo Welt" ausgegeben werden?
    #a1: Ganz normal
        >> Hallo Welt
    #a2: In Großbuchstaben
        >> HALLO WELT
    #a3: Rückwärts
        >> tleW ollaH
?F2: Welcher Zusatz soll ergänzt werden?
    #z1: "Wie geht es dir?"
        ?F1: Wdh. Wie soll "Hallo Welt" ausgegeben werden?
            #a2: In Großbuchstaben
                >> WIE GEHT ES DIR?
            #x: sonst
                >> Wie geht es dir?
    #z2: "Ich grüße dich!"
        >> Ich grüße dich!
```

Bei der Wiederholung einer Frage können auch nur Teile der ursprünglichen Antwortmöglichkeiten gegeben werden. Wird darunter die zuerst gegebene Antwort nicht gefunden, so wird keiner der Antwortzweige verarbeitet. Zur Vereinfachung kann jedoch mit dem Antwortschlüssel **#x** eine Antwortmöglichkeit gegeben werden, die stellvertretend für alle anderen Antwortmöglichkeiten steht. Wird im obigen Beispiel also für F1 die Antwortmöglichkeit **#a3** gewählt, so wird diese mit der Wiederholung der Frage dem Antwortschlüssel **#x** zugeordnet.

2.10 Implikationen

Die Antwort auf eine Frage kann in manchen Fällen die Antwort auf eine andere implizieren. Dies kann mit dem Implies-Befehl abgebildet werden.

Syntax:

```
~Implies <Frageschlüssel 1>=<Antwortschlüssel 1>; . . .
```

Beispiel:

```
Formatierte Hallo-Welt-Ausgabe
// Anwendungsbeispiel für die Befehlsreferenz
?F1: Wie soll "Hallo Welt" ausgegeben werden?
    #a1: Ganz normal
        >> Hallo Welt
    #a2: In Großbuchstaben
        >> HALLO WELT
        ~Implies F2a=j
    #a3: Rückwärts
        >> tleW ollaH
?F2: Welcher Zusatz soll ergänzt werden?
    #z1: "Wie geht es dir?"
        ?F2a: Soll der Zusatz mit Sternchen ausgegeben werden?
            #j: Ja
                >> *** Wie geht es dir? ***
            #n: Nein
                >> Wie geht es dir?
    #z2: "Ich grüße dich!"
        >> Ich grüße dich!
```

Bei Auswahl der Antwort „In Großbuchstaben“ Frage F1 wird für die Frage F2a die Antwort „Ja“ impliziert. Diese wird dann nicht mehr gestellt, sondern automatisch beantwortet. Im Falle der anderen Antwortmöglichkeiten wird keine Antwort impliziert, sodass die Frage F2a in diesem Fall ganz normal gestellt und verarbeitet wird.

2.11 Ausführungen

Bei längeren Sequenzen und insbesondere bei der Verwendung von Wiederholungen und Implikationen ist es wünschenswert, zunächst einen Teil der

Fragen vollständig abzuarbeiten, bevor die Fragen aus dem nachfolgenden Teil angezeigt werden. Dies ist mit dem Execute-Befehl möglich. Syntax:

```
~Execute
```

Beispiel:

```
Formatierte Hallo-Welt-Ausgabe
// Anwendungsbeispiel für die Befehlsreferenz
?F1: Wie soll "Hallo Welt" ausgegeben werden?
    #a1: Ganz normal
        >> Hallo Welt
    #a2: In Großbuchstaben
        >> HALLO WELT
    #a3: Rückwärts
        >> tleW ollaH
~Execute
?F2: Welcher Zusatz soll ergänzt werden?
    #z1: "Wie geht es dir?"
        >> Wie geht es dir?
    #z2: "Ich grüße dich!"
        >> Ich grüße dich!
```

Die beiden Fragen aus dem Sequenzen-Beispiel werden in diesem Fall einzeln angezeigt.

2.12 Funktionen

Vorlagen können mitunter sehr umfangreich und tief verschachtelt sein. Verwaltet man solche Vorlagen in einer einzigen Datei, kann das insbesondere aufgrund der Einrückungen sehr unübersichtlich werden, sodass sich Anpassungen und Erweiterungen schwierig gestalten. Hier ist es empfehlenswert Teile des Codes als Funktionen in jeweils eigene Dateien auszulagern. Eine Funktion ist vom Prinzip und vom Aufbau her das gleiche wie eine Vorlage und wird über eine Datei der Endung „*.qff“ (Quick Flow Function) bereitgestellt. Aufgrund der abweichenden Dateiendung werden Funktionen nicht als Vorlagen in der Anwendung zur Auswahl angeboten.

Die Funktionsdatei muss auf der gleichen Ebene liegen wie die Vorlage, die sie aufruft und der Name darf nur aus Buchstaben und Ziffern bestehen.

Syntax:

```
~Include <Name der Funktion>
```

Beispiel:

```
Formatierte Hallo-Welt-Ausgabe
// Anwendungsbeispiel für die Befehlsreferenz
~Include HalloWelt1
```

Mit der Funktionsdatei HalloWelt1.qff

```
// Hallo-Welt-Funktion
?F1: Wie soll "Hallo Welt" ausgegeben werden?
    #a1: Ganz normal
        >> Hallo Welt
    #a2: In Großbuchstaben
        >> HALLO WELT
    #a3: Rückwärts
        >> tleW ollaH
```

Neben der Erhöhung der Übersichtlichkeit hat die Verwendung von Funktionen viele weitere Vorteile. Auf technischer Ebene vereinfacht es die Verarbeitung, da eine Funktion erst dann geladen wird, wenn der entsprechende Pfad auch tatsächlich erreicht wird. Aus Anwendersicht ergibt sich der große Vorteil, dass man eine Funktion in verschiedenen Antwortzweigen aufrufen kann, und damit keinen doppelten Code pflegen muss. Entsprechend lassen sich Funktionen auch aus verschiedenen Vorlagen heraus aufrufen, sodass fachliche Einheiten ausgelagert und mehrfach wieder verwendet werden können.

2.13 Parametrisiert Funktionen

Möchte man eine Funktion innerhalb einer Vorlage mehrfach ausführen, so stößt man auf das Problem, dass die dort eingetragenen Frage-Schlüssel für beide Aufrufe identisch sind, und so der zweite Aufruf automatisch die Antworten des ersten Aufrufs übernehmen würde. Zudem wäre die Anzeige der Fragen identisch, sodass man diese beim Beantworten nicht den jeweiligen Aufrufen zuordnen könnte. Und dieses Problem zu lösen, lassen sich Aufrufe parametrisieren.

Syntax:

```
~Include <Name der Funktion> <Parameter 1>=<Wert 1>; . . .
```

Beispiel:

Formatierte Hallo-Welt-Ausgabe

```
// Anwendungsbeispiel für die Befehlsreferenz
~Include HalloWelt2 weltindex=1; weltbezeichnung=Meine Welt
~Include HalloWelt2 weltindex=2; weltbezeichnung=Deine Welt
```

Mit der Funktionsdatei HalloWelt2.qff

```
// Hallo-Welt-Funktion
// Parameter $weltindex: Schlüsselindex
// Parameter $weltbezeichnung: Weltbezeichnung
?F$weltindex1: Wie soll $weltbezeichnung begrüßt werden?
    #a1: Mit "Hallo"
        >> Hallo $weltbezeichnung
    #a2: Mit "Guten Morgen"
        >> Guten Morgen $weltbezeichnung
    #a3: Mit Aloah
        >> Aloah $weltbezeichnung
```

Beim Aufrufen der Funktion HalloWert2 werden in jeder Zeile die mit dem \$-Zeichen gekennzeichneten Variablen durch die beim Aufruf zugeordneten Werte ersetzt. Da auch in den Frageschlüsseln die Variable \$weltindex integriert wurde, ergeben sich bei den beiden aufrufen unterschiedliche Frageschlüssel F11 und F21, über die jeweils eigene Antworten verwaltet werden können. Die Texte der Fragen enthalten ebenfalls Variablen und sind damit klar zuzuordnen.

Es wird empfohlen die Parameter einer Funktion in Form von Kommentaren im Kopfbereich zu beschreiben.

Das obige Beispiel wird wie folgt angezeigt:

Vorlage Demo, 01 HalloWelt

Formatierte Hallo-Welt-Ausgabe

Wie soll Meine Welt begrüßt werden?

- ☐ Mit "Hallo"
- ☐ Mit "Guten Morgen"
- ☐ Mit "Aloah"

Wie soll Deine Welt begrüßt werden?

- ☐ Mit "Hallo"
- ☐ Mit "Guten Morgen"
- ☐ Mit "Aloah"

Eine eindeutige Benennung der Schlüssel wäre auch automatisch möglich gewesen, jedoch auf Kosten der Möglichkeit, innerhalb einer Funktion wahlweise auch auf Fragen der aufrufenden Stelle zuzugreifen. Deshalb wurde hierauf verzichtet.

2.14 Variablen

Variablen können auch unabhängig von Funktionsparametern gesetzt und in Fragen, Antworten und Ausgaben verwendet werden.

Syntax:

```
~Set <Variable 1>=<Wert 1>; <Variable 2>=<Wert 2>; . . .
```

Beispiel:

Formatierte Hallo-Welt-Ausgabe

```
// Anwendungsbeispiel für die Befehlsreferenz
```

```
?F1: Wie soll "Hallo Welt" ausgegeben werden?
```

```
#a1: Ganz normal
```

```
>> Hallo Welt
```

```
~Set Preis=5 Euro
```

```
#a2: In Großbuchstaben
```

```
>> HALLO WELT
```

```
~Set Preis=8 Euro
```

```
#a3: Rückwärts
```



```

>> tleW ollaH
~Set Preis=12 Euro
~Execute
?F2: Soll der Preis von $Preis ausgegeben werden?
#j: Ja
>> Der Preis beträgt $Preis.
#n: Nein

```

Wird eine Variable in einer Frage oder Antwort verwendet, sollte durch die Struktur sichergestellt werden, dass die Frage, die für das Setzen der Variablen zuständig ist, zuvor ausgeführt wurde. Dies kann z.B. mit dem oben beschriebenen Execute-Befehl geschehen. Additionen Über Variablen lassen sich auch einfache Additionen durchführen, beispielsweise um Kosten, Aufwände oder Komplexität zu ermitteln, die sich aus den gewählten Optionen zusammensetzen. Dies ist ebenfalls über den Set-Befehl möglich.

Syntax:

```
~Set <Variable 1>+<Inkrement 1>; . . .
```

Additionen und Variablenzuweisungen können auch im gleichen Set-Befehl gemischt werden. Bei ihrer ersten Verwendung in einer Addition wird der Variablen zuvor ein Wert von 0 zugewiesen. Hat eine Variable bei der Verwendung in einer Addition bereits einen Wert, der nicht als natürliche Zahl interpretierbar ist, so wird die Addition ignoriert.

Beispiel:

```

Formatierte Hallo-Welt-Ausgabe
// Anwendungsbeispiel für die Befehlsreferenz
?F1: Wie soll "Hallo Welt" ausgegeben werden?
#a1: Ganz normal
>> Hallo Welt
~Set Preis=5
#a2: In Großbuchstaben
>> HALLO WELT
~Set Preis=8
#a3: Rückwärts
>> tleW ollaH
~Set Preis=12
?F2: Welcher Zusatz soll ergänzt werden?
#z1: "Wie geht es dir?"
>> Wie geht es dir?

```

```
~Set Preis+=3
#z2: "Ich grüße dich!"
>> Ich grüße dich!
~Set Preis+=7
?F3: Soll der Preis ausgegeben werden?
#j: Ja
>> Der Preis beträgt $Preis Euro.
#n: Nein
```