

### 7.3.3 Klasse Queue (gb.data)

Die Klasse *Queue* – (Warte-)Schlange oder Reihe – stellt einen Datentyp bereit, mit dessen Hilfe Sie eine Vielzahl von Objekten (Elemente der Queue) als Sequenz verwalten können. Die Klasse *Queue* verwendet den generischen Datentyp *Variant* für ihre Elemente. Die schematische Bezeichnung FIFO (First in, first out) für die Queue verdeutlicht das Prinzip: Das Element, das zuerst in die Queue eingereiht wurde, wird diese auch zuerst verlassen. Alle anderen Elemente rücken dann eine Position nach vorn. Der Vergleich dieser Datenstruktur mit einer Schlange im Markt ist gängig: Der Kunde, der sich zuerst anstellt, wird auch zuerst bedient. Danach folgt der zweite und so weiter. Jeder neue Kunde muss sich an das Ende der Schlange anstellen.

#### 7.3.3.1 Eigenschaften

Die Eigenschaften der Klasse *Queue* finden Sie in der folgenden Tabelle:

Eigenschaft	Datentyp	Beschreibung
.Size	Integer (ReadOnly)	Anzahl der Elemente der Queue
.IsEmpty	Boolean (ReadOnly)	Wahr, wenn keine Elemente in der Queue sind, sonst falsch

Tabelle 7.3.3.1.1: Eigenschaften der Klasse *Queue*

#### 7.3.3.2 Methoden eines Queue-Objektes

Das sind die Methoden der Klasse *Queue*:

Methode	Beschreibung
Clear()	Entfernen aller Elemente
Enq(vElement)	Synonym für Enqueue(vElement)
Enqueue(vElement)	Einreihen eines neuen Elements (an das Ende der Queue)
Deq()	Synonym für Dequeue()
Dequeue()	Entfernen und zurückgeben des ersten Elements
Peek()	Rückgabe des ersten Elements, ohne es zu entfernen

Tabelle 7.3.3.2.1: Methoden der Klasse *Queue*

**Achtung:**

Beim Einsatz von *Deq()* ist Vorsicht geboten! Der Aufruf dieser Methode verändert die Queue und gibt gleichzeitig einen Wert zurück. Soll dieser Wert mehrmals verwendet werden, so muss er in einer Variablen gespeichert werden. Ein weiterer Aufruf von *Deq()* würde bereits das nächste Element zurückgeben. Möchte man auf eine Variable verzichten, kann alternativ auch die *Peek()*-Methode benutzt werden. Man muss aber daran denken, das Element bei der letzten Benutzung mit *Deq()* aus der Queue zu entfernen.

#### 7.3.3.3 Projekt zum Einsatz der Klasse Queue

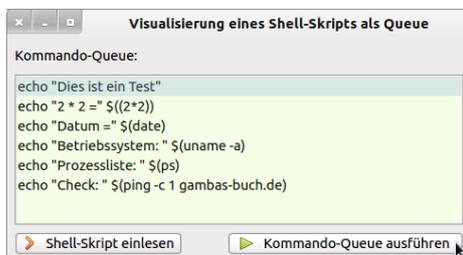


Abbildung 7.3.3.3.1: Projekt Queue

Queues sind geeignet, eine Sequenz oder eine Reihenfolge von Objekten darzustellen. In diesem Sinne wird sie in einem Demonstrationsprojekt genutzt, um ein Shell-Skript zu repräsentieren, das selbst nur eine Sequenz von Kommandos ist.

Es wird ein vom Benutzer ausgewähltes Shell-Skript in eine Queue eingelesen und danach die Kommandos als Sequenz von einer Shell interpretiert. Der Text des Kommandos und die Ausgaben der Shell werden in der Konsole ausgegeben. Die einzelnen Kommandos sind nicht repräsentativ.

Man beachte dabei, dass die Shell in der Regel einige syntaktische Strukturen beherrscht, die nicht als Sequenz verstanden werden können. Die im Beispiel verwendete Bourne-Again-Shell (bash) kann zum Beispiel eine While-Kontrollstruktur interpretieren. Da eine Wiederholung von Anweisungen keine Sequenz ist, versagt die Queue in diesem Fall.