Algorithmen und Struktogramme

Algorithmus

Bisher wurden im Wesentlichen objektorientierte Modelle zu Problembeschreibungen entworfen und erste Implementierungsversuche dieser Modelle vorgenommen. Bei der Implementierung der Modelle mithilfe einer Programmiersprache zeigte sich relativ schnell, dass komplexere Methoden noch nicht hinreichend gut beschrieben wurden.

Die Frage, was die einzelnen Methoden *tun*, wurde bisher nur undeutlich umschrieben oder war sofort ersichtlich, weil die entsprechende Methode nicht viel zu tun hatte (vgl. gib- und setze-Methoden).

Im Folgenden stellt sich aber die Frage, wie komplexere Abläufe so beschrieben werden, dass sie von einem Computer verarbeitet werden können. Eine solche Ablaufbeschreibung bezeichnet man als Algorithmus (Plural: Algorithmen):

> »Algorithmus: eine Verarbeitungsvorschrift, die so präzise formuliert ist, dass sie von einem mechanisch oder elektronisch arbeitenden Gerät durchgeführt werden kann. Aus der Präzision der sprachlichen Darstellung des Algorithmus muss die Abfolge der einzelnen Verarbeitungsschritte eindeutig hervorgehen. Hierbei sind Wahlmöglichkeiten zugelassen« Claus und Schwill 2006.

Beschreibung von Algorithmen mit Struktogramme

Zur systematischen Beschreibung von Algorithmen werden sogenannte Struktogramme (nach ihren Entwicklern auch Nassi-Shneiderman-Diagramme genannt) verwendet. Ein Struktogramm liest man von oben nach unten, wobei jedes Rechteck (auch *Strukturblock* genannt) eine Anweisung des beschriebenen Algorithmus wiedergibt.

Die Formulierung der einzelnen Anweisungen soll dabei umgangssprachlich (nicht programmiersprachenabhängig) *aber strukturiert* erfolgen. Damit ist gemeint, dass die einzelnen Anweisungen eindeutig zu verstehen sind und später in eine beliebige Programmiersprache überführt werden können (vgl. Definition Algorithmus).

Über jedes Struktogramm gehört zudem ein Name, über den das jeweilige Struktogramm eindeutig zu identifizieren ist, so dass Verweise auf andere Struktogrammen möglich sind. Üblicherweise stellt **ein Struktogramm** später in der Implementierung in einer Programmiersprache **eine Methode** dar. Ein Verweis bedeutet dann den Aufruf einer anderen Methode.

berechne Augenzahlsumme

Betrachte Augenzahl von wuerfel1
Betrachte Augenzahl von wuerfel2
Addiere die beiden Augenzahlen
Gib die Summe zurück

Abbildung 1: Ein Algorithmus, der die Augenzahlen zweier Würfel addiert.

fülle Kaffeemaschine

Fülle 0,6L Wasser in die Kaffekanne
Gieße das Wasser in den Wasserbehälter
Lege eine Filtertüte in den Filter
...

Abbildung 2: Struktogramme können auch beliebige Abläufe darstellen, die von einer Programmierung unabhängig sind.

Verzweigungen in Struktogrammen

Algorithmen sind nicht immer lineare Abfolgen von Anweisungen, sondern können auch Wahlmöglichkeiten enthalten. Man spricht hier üblicherweise von *Verzweigungen*. Eine Verzweigung gibt – abhängig von einer Bedingung – an, welcher von zwei Algorithmenabschnitten im Folgenden durchlaufen werden soll.

Eine beispielhafte Verwendung einer Verzweigung beim Algorithmus *koche Kaffee* würde in etwa wie folgt aussehen. Ist die Bedingung (Sind die Kaffeetrinker müde?) erfüllt, so wird der linke Abschnitt abgearbeitet, ansonsten der Rechte. Danach wird unabhängig von der Entscheidung bei der Verzweigung auf jeden Fall noch die letzte Anweisung (Drücke auf den Start-Knopf) ausgeführt.

koche Kaffee

Fülle 1L Wasser in die Kaffekanne	
Gieße das Wasser in den Wasserbehälter	
Lege eine Filtertüte in den Filter	
Sind die Kaffeetrinker müde?	
Ja	Nein
Gib 6 Löffel Pulver hinein	Gib 5 Löffel Pulver hinein
Drücke auf den Start-Knopf	

Abbildung 3: Struktogramm eines Algorithmus mit einfacher Verzweigung

Schleifen in Struktogrammen

Anweisungen können sich in Algorithmen wiederholen. Will man diese wiederholenden Anweisungen zusammenfassen benutzt man dafür üblicherweise Schleifen. Eine Schleife wiederholt einen Block von Anweisungen so lange, bis ein Abbruchkriterium erreicht ist.

Eine beispielhafte Verwendung einer Schleife beim Algorithmus *koche Kaffee* würde in etwa wie folgt aussehen. Solange der Kaffe noch nicht durchgelaufen ist, wird die Anweisung (ungeduldig warten) wiederholt. Erst wenn das Abbruchkriterium (Ist der Kaffee durchgelaufen?) erfüllt ist, kann der Kaffee in die Tasse gegossen und getrunken werden.

trinke Kaffee

Solange der Kaffee noch nicht durchgelaufen ist
Warte ungeduldig
Gieße den Kaffee in die Tasse
Trinke die Tasse

Abbildung 4: Struktogramm eines Algorithmus mit while-Schleife

Literatur

Claus, Volker und Andreas Schwill (Feb. 2006). *Duden Informatik A–Z. Fachlexikon für Studium und Praxis*. Hrsg. von Meyers Lexikonredaktion. 4., überarb. u. aktualis. Aufl. Mannheim, Leipzig, Wien, Zürich: Bibliographisches Institut. ISBN: 3-411-05234-1.