

Verhalten

- Def. und Nutzen von Verhalten
 - Algorithmen
- Pseudocode
- Schreibtischtest

Verhalten & Pseudocode

- Das Verhalten beschreibt, wie sich die Datenstrukturen (Variablen) eines Programms verändern müssen, um eine gewünschte Funktionalität des Programms zu erhalten
- Ein Verhalten wird in der Programmierwelt Algorithmus genannt und mit Hilfe von Anweisungen der Programmiersprache umgesetzt

Definition Algorithmus

"Ein Algorithmus ist eine eindeutige, endliche Beschreibung eines allgemeinen, endlichen Verfahrens zur schrittweisen Ermittlung gesuchter Größen aus gegebenen Größen.

Die Beschreibung erfolgt in einem Formalismus mit Hilfe von anderen Algorithmen und, letztlich, elementaren Algorithmen. Ein Algorithmus muß ausführbar sein."

(Def. aus Balzert 99, S. 469/470)

Definition Terminieren

Ein Algorithmus heißt terminierend, wenn er (bei jeder erlaubten Eingabe von Parametern) nach endlich vielen Schritten abbricht.

Definition Determinismus

Ein Algorithmus hat einen deterministischen Ablauf, wenn er eine eindeutige Schrittfolge besitzt. Ein Algorithmus liefert ein determiniertes Ergebnis, wenn bei vorgegebener Eingabe (auch bei mehrfacher Durchführung) stets ein eindeutiges Ergebnis erreicht wird.

Programm

- Begriffe
 - Programm
 - Algorithmus in einem eindeutigen und präzisen Formalismus
 - Computer erhalten durch Programme mitgeteilt, welche Aufgaben sie ausführen sollen
 - Da ein Computer ohne ein Programm nicht arbeiten kann, spricht man von Computersystemen, wenn das technische Gerät Computer und die Programme zur Steuerung des Computers gemeint sind.

Pseudocode

- Um den Ablauf eines Programm zu beschreiben, bietet sich Pseudocode an
 - Verzichtet auf eine zu technische Darstellung, ist auch für Nicht-Programmierer verständlich und lehnt sich dabei an die Maschinenverarbeitung von Daten an
 - Durch gute Lesbarkeit kann das Programm "mental" ausprobiert und verbessert werden
 - eine Pseudocodedefunktion besteht aus dem Funktionskopf (Name, Ein- Ausgabeparameter) und Funktionsbody

Pseudocode-Funktion

- Eingeleitet wird eine Pseudocode-Funktion mit `function <name> (parameter) : Rückgabe`
- Parameter werden zuerst mit Namen gefolgt vom Typ angegeben, getrennt von „:“
- Bei gleichen Typen können die Parameter auch durch Komma getrennt werden
- Funktionen mit Rückgabe müssen mit `return <ausdruck>` enden
 - erzeugt eine Funktion keine Rückgabe, muss diese mit `void` enden
- Beispiel

```
function plus2 (a:Zahl):Zahl
return a+2
```

Pseudocode Bestandteile

- Schleifen
 - Für alle Bücher b in der Liste...
 - Belege i mit 1 bis 10...
 - Wiederhole bis ...
 - Für $m:=1$ bis 10
- Bedingungen
 - Wenn Farbe gleich blau dann..
 - Ist Liste gefüllt dann...
- Einrückung
 - Durch Einrücken wird der "Body" einer Schleife, Bedingung oder Funktion gekennzeichnet
 - Für alle Pustebäumen p
←→ Puste p

Pseudocode Bestandteile

- Zuweisungen
 - Durch ein Doppelpunkt gefolgt von „=“ wird die Zuweisung eines Wertes zu einer Variablen veranschaulicht
 - $i:=i+1$
 - $k:=0$
- Vergleiche
 - In booleschen Ausdrücken wird Sprache oder das „==“ Zeichen genutzt
 - A ist gleich B
 - $A==B$
 - Ist A gleich B?

Funktionen in Pseudocode

- Funktionsaufrufe

- Alle definierten Funktionen können im Pseudocode anderer Funktionen aufgerufen werden

- `function plus(a,b:Zahl):Zahl`

`return a+b`

- `function multipli(p,m:Zahl):Zahl`

`für i:=1 bis m`

`p:=plus(p,p)`

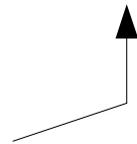
`return p`

Schreibtischtest

- Macht die Zeitliche Veränderung von Variablen sichtbar
- Wird genutzt, um die korrekte Funktionsweise von Pseudocode zu testen
- In einer Tabelle
 - Spalten enthalten die Variablen des Algorithmus
 - Die Zeilen entsprechen dem sich fortentwickelnden Zeitstrahl als Schrittnummer oder Zeilennummer
- Nach dem Ausführen eines Algorithmus ist die Änderung der Variablen sichtbar

Schreibtischtest Beispiel

Schritt	Zeile	A	i
1	1	10	1
2	2	10	2
3	1	11	2
4	2	11	3
5	1	12	3
6	2	12	4
7	3	13	4
8	1	10	1



Durch Nummerieren der Zeilen im Pseudocode kann ganz genau auf die Anweisung verwiesen werden, welche die Variable abändert. Die Zeilen können „zurückspringen“, falls sich die Programmausführung in einer Schleife befindet.

Aufgabe

- Addition aller Vielfache von 5 zwischen 0 und 30
 - Erstellen sie Pseudocode, der die obige Summe berechnet und in der variable **ergebnis** speichert
 - vergeben sie Zeilennummern
- Testen sie ihren Algorithmus mit einem Schreibtischtest und geben sie die berechnete Summe aus, nachdem der Algorithmus terminiert