Carl-Zeiss-Gymnasium Jena
Fachbereich INFORMATIK

http://www.carl-zeiss-gymnasium.de

Algorithmen

Mirko König

Diese Unterlagen dienen der Informatik-Ausbildung der Spezialklassen am Carl-Zeiss-Gymnasium Jena in den Klassenstufen 9 und 10. Sie sind in erster Linie Arbeitsanleitung zur selbstständigen Tätigkeit und bieten darüberhinaus Übungsmaterialien.

Dieses Dokument wurde mit LaTeX gesetzt.

Jena, 5. Februar 2016

Inhaltsverzeichnis

1 Lerneinheit Eins
Algorithmen

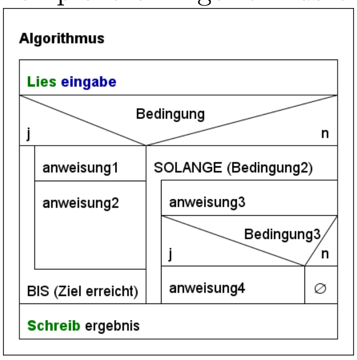
1.1 Eigenschaften von Algorithmen

Eigenschaft	Bedeutung
Abstraktion	
Determiniertheit	
Terminierung	
Endlichkeit	
Effektivität	
Diskretheit	

1.2 Beispiele für Algorithmen

1.2.1 Struktogramme

Wir verwenden für die Darstellung von Algorithmen **Struktogramme**. Das sind Rechtecke oder bestimmte Kombinationen von Rechtecken. Dabei steht ein **Rechteck** für eine **Anweisung** (im Sinne der Diskretheit). Im Bild ist ein komplexerer Algorithmus beispielhaft dargestellt.



Das Struktogramm wird von oben nach unten gelesen und auch in dieser Reihenfolge ausgeführt.

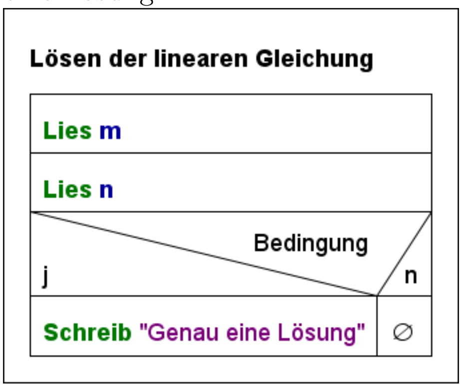
1.2.2 Zähne putzen - Eine Sequenz aus Anweisungen

Das morgendliche Putzen der Zähne ist ein Beispiel für einen Vorgang, der in einer Abfolge (Sequenz) von Handlungsschritten erledigt werden kann. Vervollständige das nachfolgende Struktogramm als Abfolge von Handlungsanweisungen. Die Anzahl der Zwischenschritte darf selbstverständlich variiert werden.

1.2.3 Eine lineare Gleichung lösen - Entscheidungen fällen (Alternative)

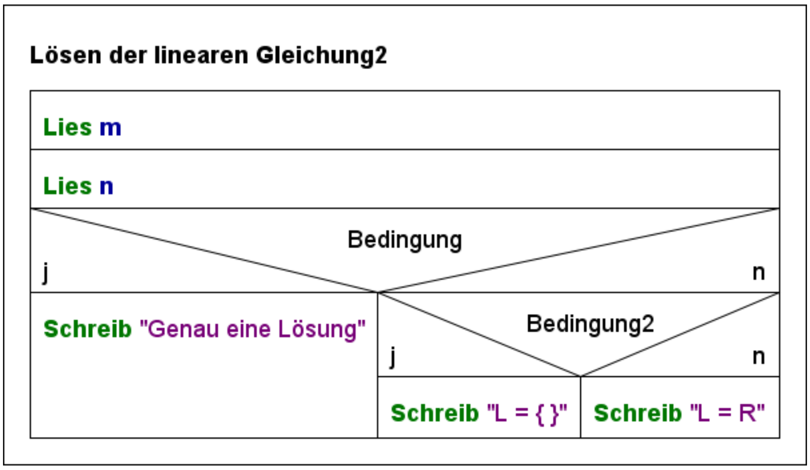
Gegeben ist die lineare Gleichung $0 = m \cdot x + n$. Es soll ihre Lösungsmenge bestimmt werden. Wie sicher bekannt ist, hängt die Lösungsmenge von den Parametern m und n ab. So gibt es Fälle, in denen es genau eine Lösung der Gleichung gibt. Weiterhin kann es keine Lösung geben oder unendlich viele.

Aufgabe Bestimme die Bedingung (`Bedingung`) für den Fall: Es gibt genau eine Lösung.



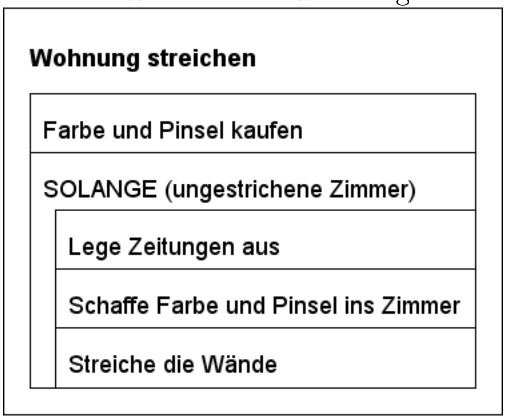
Der Algorithmus teilt sich nun in zwei Zweige: Den Ja-Zweig und den Nein-Zweig. Der Nein-Zweig ist noch leer (das ist auch möglich).

Aufgabe Bestimme nun die Bedingung `Bedingung2` für die Alternative im Nein-Zweig. Wir müssen ja noch entscheiden, ob es keine oder unendlich viele Lösungen gibt.
Unser Algorithmus sieht nun so aus:



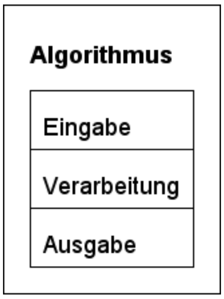
1.2.4 Wohnung streichen - Wiederholung von Anweisungen

Eine spezielle Form der Algorithmen sind die Schleifen. Dabei wird eine (Sequenz von) Anweisung(en) mehrfach unter Bedingungen wiederholt. Die Bedingung kann am Beginn der Schleife stehen (Schleife mit Eingangsbedingung), am Ende (Schleife mit Endbedingung), oder es gibt eine feste Anzahl Wiederholungen (gezählte Wiederholung).
Ein (sicher noch sehr grob formulierter) Algorithmus dieser Art stellt das Streichen der Wände einer Wohnung dar:

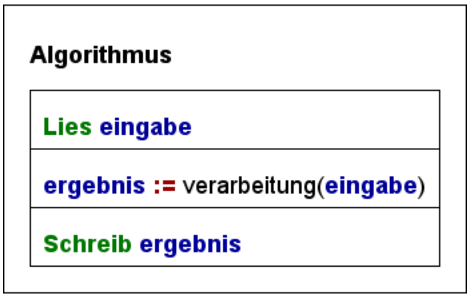


1.3 EVA-Prinzip

Eine wichtige Eigenschaft von Algorithmen ist die *Abstraktion*. Sie bedeutet: Es sind **Variablen** nötig, die es dem Algorithmus ermöglichen, bestimmte Werte von außen entgegenzunehmen. Diese Werte werden also **eingegeben**, dann **verarbeitet**. Anschließend werden Ergebnisse **ausgegeben**.



Für die Eingabe und zum Rechnen werden Variablen benötigt. Ihre Namen müssen zusammenhängende Zeichenketten sein, die mit einem Buchstaben beginnen. Die Namen sollten den Zweck der Variable beschreiben („sprechende Namen“).



1.4 Aufgaben

1. Drei eingegebene Namen sollen in alphabetischer Reihenfolge wieder ausgegeben werden. Hinweis: Zeichenketten lassen sich mittels der Relation „<“ anordnen; z. B. gilt „`Kulcke" < „Kulle"`.
2. Eine Bank führt bei jedem Girokonto die ersten zehn Buchungen kostenlos durch. Für die nächsten zehn Buchungen berechnet sie 30 Cent pro Buchung, für jede weitere Buchung 20 Cent. Ein Algorithmus ist zu formulieren, der zur gegebenen Anzahl von Buchungen die Gebühren berechnet.
3. Gegeben ist eine lineare Gleichung $0 = m \cdot x + n$. Die Parameter m und n werden eingegeben. Ausgegeben werden soll die Lösung der Gleichung, falls sie existiert und eindeutig ist. Anderenfalls soll die entsprechende Lösungsmenge ausgegeben werden.
4. Eine Firma gewährt Großhändlern 15 %, Einzelhändlern 10 % und normalen Kunden 5 % Rabatt. Schreibe einen Algorithmus, der je nach Kundentyp mittels Mehrfachauswahl eine Rechnung erstellt.
5. Es bezeichnen a, b, c, Ao, V die Länge, Breite, Höhe, Oberfläche und das Volumen eines Quaders. Aus je dreien lassen sich die restlichen beiden Größen berechnen. Z. B. ist bei gegebenen a, b, V die Höhe `c=V/(a*b)` und `Ao=2*(a*b+V/b+V/a)` die Oberfläche. Schreibe einen Algorithmus, der dem Benutzer die Wahl der zu berechnenden Größen gestattet.
6. Auf dem Computer soll eine Briefwaage mit Portoanzeige (laut Gebührenordnung der Post) simuliert werden. Eingabe: Masse des Briefes, Ausgabe: Briefporto.
7. Ein Algorithmus ist gesucht, der alle Lösungen der quadratischen Gleichung $ax^2 + bx + c = 0$ ermittelt und ausgibt.
8. Entwickle einen Algorithmus, der folgendes leistet: Der Benutzer gibt einige ganze Zahlen ein, der Computer nennt nach jeder Eingabe die aktuelle größte und kleinste der bereits eingegebenen Zahlen.
9. Zu gegebenen positiven Zahlen soll der Computer das arithmetische Mittel berechnen und ausgeben. (Verwende als Abbruchbedingung die Eingabe einer negativen Zahl und achte darauf, dass diese nicht in die Berechnung des Mittelwerts eingeht.)
10. Der Computer soll die Zahlenfolgen a) 1, 3, 6, 10, 15, ... (Dreieckszahlen), b) 1, 4, 9, 16, 25, ... (Quadratzahlen), c) 1, 5, 12, 22, 35, ... (Fünfeckzahlen) berechnen und ausgeben. Verwende als Abbruchbedingung einmal die *Länge* der Folge, zum anderen die *Größe* der letzten noch auszugebenden Zahl.
11. Bankier Dagobert schätzt sein Vermögen auf 1 bis 2 Millionen Dukaten. Um sich die Zeit zu vertreiben, legt er seine Dukaten einmal in Quadrat- und einmal in Dreiecksform, wobei er jeweils alle Münzen verwendet, ohne eine übrig zu lassen. Wieviele Dukaten besitzt Dagobert? (Anleitung: Der Computer muss die Folge der Dreieckszahlen und die der Quadratzahlen durchlaufen.)

12. Ein Fahrscheinautomat gibt Wechselgeld bis zu 9,99 Euro aus. Er soll so programmiert werden, dass er für den auszugebenden Wechselbetrag möglichst wenig Münzen benötigt. Dialogbeispiel:

    ```
    Eingabe: Wechselbetrag (in EUR): 6.88
    Ausgabe: 5 EUR, 1 EUR, 50 c, 20 c, 10 c, 5 c, 2 c, 1 c
    ```

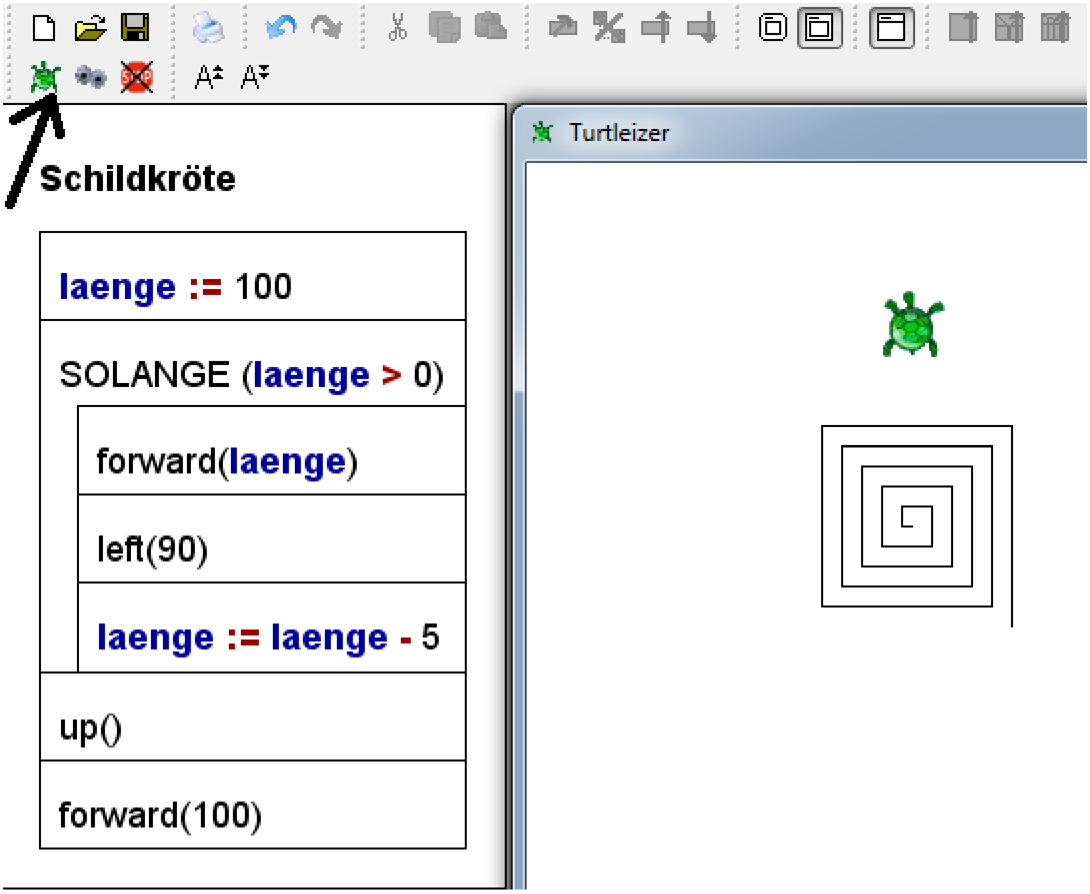
    Verwende folgenden Algorithmus (*Pseudocode*)

    ```
    Eingabe: Wechselbetrag in EUR
    Rest := gerundet(Wechselbetrag * 100)
    SOLANGE Rest >= 500 WIEDERHOLE
      Ausgabe "5 EUR"
      Rest := Rest - 500
    ENDE-WIEDERHOLE
    SOLANGE Rest >= 200 WIEDERHOLE
      Ausgabe "2 EUR"
      Rest := Rest - 200
    ENDE-WIEDERHOLE
    usw.
    ```

13. Schreibe einen Algorithmus, der die Summe aller natürlichen Zahlen zwischen zwei gegebenen Zahlen a, b bestimmt (Beispiel: $a = 5$, $b = 10$, $5 + 6 + 7 + 8 + 9 + 10 = 45$).
14. Es sollen alle natürlichen Zahlen zwischen zwei gegebenen Grenzen ausgedruckt werden, die a) durch 5 und durch 7, b) durch 5 oder 7 oder 13 teilbar sind. (Die Teilbarkeit von a durch b lässt sich mit Hilfe der Funktion `mod` testen: Es gilt $b \mid a$ genau dann, wenn `a mod b = 0`.)
15. Zu einer gegebenen Zahl n sollen alle Teiler ermittelt werden. Beispiel: Die Zahl 2345 hat folgende Teiler: 1, 5, 7, 35, 67, 335, 469, 2345. Lass zunächst eine Zählvariable von 1 bis n laufen. Überlegen dann, ob es auch weniger aufwendig geht; bedenke dazu, dass die Teiler stets in Paaren (Teiler, Gegenteiler) auftreten.
16. Ein Brief soll mit 18 Kopeken frankiert werden. Zur Verfügung stehen Marken im Wert von 4, 6, und 10 Kopeken, und zwar in beliebiger Anzahl. Welche Möglichkeiten gibt es, den Brief zu frankieren? (Verwende drei ineinander geschachtelte Zählschleifen.)
17. Ein Hufschmied beschlägt ein Pferd, und zwar befestigt er jedes der vier Hufeisen mit sechs Nägeln. Als es ans Bezahlen geht, fragt er den Reiter: „Wie willst Du bezahlen - entweder für alle 24 Nägel zusammen 50 EUR oder für den ersten Nagel 1 Cent, für den zweiten Nagel 2 Cent, für den dritten Nagel 4 Cent und so fort?“ Natürlich entscheidet sich der Reitersmann für das Pfenniggeschäft. Hat er klug gehandelt? (Lass den Computer rechnen.)
18. Der Benutzer denkt sich eine ganze Zahl (zwischen 1 und 32000); der Computer soll sie mit möglichst wenig Rückfragen ermitteln. Zu diesem Zweck teilt der Benutzer mit, ob eine vom Computer vorgeschlagene Zahl kleiner bzw. größer als die gedachte Zahl oder ihr gleich ist. Algorithmus gesucht.

19. In einem antiken Rätselgedicht wird das Leben des Mathematikers Diophant von Alexandria wie folgt beschrieben:
    „Hier dies Grabmal deckt Diophantos. Schauet das Wunder! Durch des Entschlafenen Kunst lehret sein Alter der Stein. Knabe zu sein, gewährte im Gott ein Sechstel des Lebens; noch ein Zwölftel dazu, sproßt' auf der Wange der Bart; dazu ein Siebentel noch, da schloß er das Bündnis der Ehe, nach fünf Jahren entsprang der Verbindung ein Sohn. Wehe, das Kind, das vielgeliebte, die Hälfte der Jahre hatt' es des Vaters erreicht, als es dem Schicksal erlag. Darauf vier Jahre hindurch - mit der Zahlen Betrachtung den Kummer verscheuchend - kam auch er an das irdische Ziel."
    Wie alt wurde Diophant? (Lass den Computer die im Text versteckte *diophantische Gleichung* lösen!)
20. Teste den Zufallsgenerator, indem Du einen Spielwürfel simulierst und den Computer eine Häufigkeitstabelle für das Auftreten der Augenzahlen 1, 2, . . . , 6 aufstellen lässt.
21. Wie oftmuss man im Mittel würfeln, bis eine 6 erscheint? Simuliere den Vorgang!
22. Ein Maß für die „Zufälligkeit" einer Ziffernfolge aus Nullen und Einsen ist die Häufigkeit der Wechsel $0 \to 1$ und $1 \to 0$. Der theoretische Wert hierfür ist 0,495. Teste nach dieser Methode vom Computer erzeugte 0/1-Folgen.
23. Klasse 10g hat 24 Schüler. Wie groß ist die Wahrscheinlichkeit, dass zwei von ihnen am gleichen Tag Geburtstag feiern können? Schreibe ein Simulationsprogramm!

1.5 *Turtle*-Grafiken mit dem Turtleizer



Im *Structorizer* sind einige Grafikanweisungen vorhanden, mit deren Hilfe eine Schildkröte (engl. *turtle* über ein quadratisches Fenster laufen kann. Sie kann geradeaus gehen und dabei eine oder keine Spur hinterlassen sowie sich nach rechts und links drehen. Zu Beginn blickt die Schildkröte nach oben.

`forward(anzahlPixel)`	Bewegung um `anzahlPixel` nach vorn
`backward(anzahlPixel)`	Bewegung um `anzahlPixel` nach vorn
`right(winkelInGrad)`	Drehung um `winkelInGrad` Grad nach rechts
`left(winkelInGrad)`	analog
`up()`	„Heben des Stiftes“ von nun an keine Spur
`down()`	„Absenken des Stiftes“ von nun an wird Spur gezeichnet
`gotoXY(x,y)`	Bewegung an die Koordinaten `x,y`
`hideTurtle()`	Schildkröte wird von nun an versteckt
`showTurtle()`	Schildkröte wird ab jetzt angezeigt