

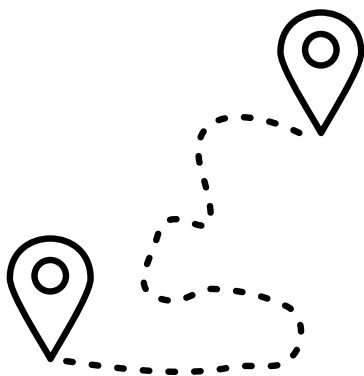
ALGORITHMEN



01 Algorithmen im Alltag und erste Definition

EIN KUCHEN NACH REZEPT

Wenn du einen Kuchen backen willst und dabei einem Rezept folgst, nutzt du einen Algorithmus. Ein Backrezept ist ja nichts anderes als eine Handlungsanweisung, die du Schritt für Schritt ausführst. Machst du alles richtig, gelangst du an dein Ziel. In diesem Fall ein leckerer Kuchen.



Auch wenn euch jemand nach dem Weg fragt, gebt ihr der Person einen Algorithmus. Ihr beschreibt den Weg als eine Abfolge von Anweisungen wie

1. Gehe geradeaus bis zur Kreuzung
2. Biege an der Kreuzung links ab
Folge der Straße bis zum Ziel

Wenn die andere Person diese Anweisungen richtig befolgt, wird sie ihr Ziel erreichen.

AUFGABE 1

Sogar der Weg zur Schule kann als Algorithmus formuliert werden. Beschreibe deinen persönlichen 'Schulweg-Algorithmus'. Schreibe dafür fünf Schritte auf, die du morgens auf den Weg zur Schule nacheinander ausführst. Mögliche Schritte könnten z.B. 'aus dem Haus gehen' oder 'aufs Fahrrad steigen' sein.

1. Ich gehe aus dem Haus

2. Steige auf das Fahrrad

3. Fahre mit dem Fahrrad die Strecke zur Schule

4. Währenddessen halte ich mich an die Verkehrsregeln

5. Komme an der Schule an

Du siehst also, dass uns solche Abfolgen von Anweisungen ständig begegnen. Denn nicht nur Computer folgen Regeln. Auch unser Verhalten kann manchmal mit einem Algorithmus beschrieben werden.

ALGORITHMEN

01 Algorithmen im Alltag und erste Definition



ERSTE DEFINITION VON "ALGORITHMUS"

Ein Algorithmus ist eine Verarbeitungsvorschrift, die aus einer **endlichen** Folge von **eindeutig ausführbaren** Anweisungen besteht.

Unter gleichen Voraussetzungen liefert die Ausführung eines Algorithmus stets gleiche Ergebnisse.

EIGENSCHAFTEN VON ALGORITHMEN

Aus der oben stehenden Definition können wir ableiten, dass ein Algorithmus die folgenden Eigenschaften hat: **endlich**, **ausführbar** und **eindeutig**.

Doch was bedeuten diese Eigenschaften genau. Dies wollen wir uns als nächstes anschauen.

AUFGABE 2

Hier stehen die Eigenschaften von Algorithmen auf der einen und deren Definition auf der anderen Seite. Bilde richtige Paare.

Ausführbarkeit

Jede einzelne Anweisung muss für den Ausführenden Algorithmus verständlich und eindeutig ausführbar sein.

Eindeutigkeit

Ein Algorithmus besteht aus endlich vielen Anweisungen. In der Praxis soll ein Algorithmus nach endlich vielen Schritten ein Resultat liefern

Endlich

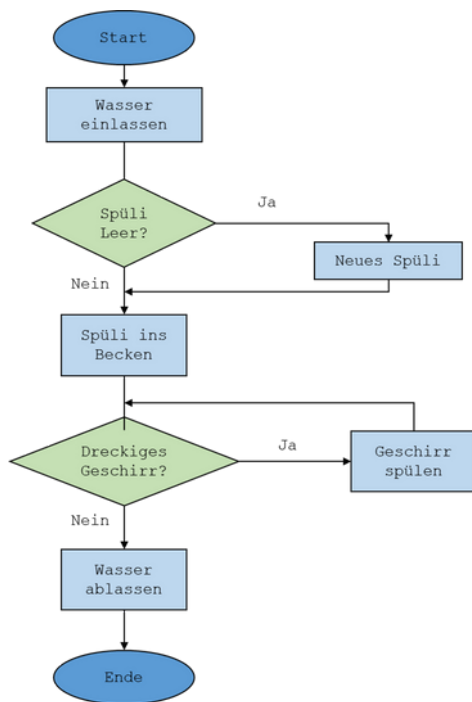
Mit jeder Anweisung ist auch die nächstfolgende festgelegt. Wird der Algorithmus mit den gleichen Voraussetzungen gestartet, so liefert die Ausführung stets gleiche Ergebnisse.

ALGORITHMEN



02 Algorithmen bildlich darstellen

Wir haben gesehen, dass Algorithmen uns in unserem Alltag begegnen, aber wofür brauchen wir sie eigentlich beim Programmieren? Prozesse und Abläufe als Algorithmus darzustellen, erleichtert das Schreiben von Programmen. Das Flussdiagramm ist eine Möglichkeit den Ablauf eines Programmes algorithmisch darzustellen.



Das links dargestellte Flussdiagramm besteht aus verschiedenen Elementen wie runden/eckigen Kästchen und Pfeilen.

DEFINITION: FLUSSDIAGRAMME

Ein Flussdiagramm ist eine grafische Darstellung eines Algorithmus, der als Programm umgesetzt werden soll. Sie helfen beim Schreiben und Verstehen von Programmen, da sie den Ablauf eines Algorithmus anschaulich beschreiben. Der Name kommt daher, dass sie den Ablauf oder "Fluss" eines Programms grafisch darstellen.

Ein anderer Name für ein Flussdiagramm für ein Algorithmus ist Programmablaufplan (PAP)

AUFGABE 3 - MULTIPLE CHOICE

Kreuze dir richtige Antwort an.

Woran erkennt man, wo das Programm anfängt und wo es endet?

- Start und Ende werden als eckige Kästchen dargestellt
- x • Start und Ende werden als Kästchen mit abgerundeten Ecken dargestellt
- Das ist nicht klar, man muss raten

Wie werden Anweisungen in einem Flussdiagramm dargestellt?

- x • Als viereckige Kästchen
- Als abgerundete Kästchen
- Als Dreiecke



AUFGABE 3 - MULTIPLE CHOICE

Kreuze dir richtige Antwort an.

Wie wird eine Abfolge von Anweisungen (Sequenz) dargestellt?

- Alle Anweisungen werden einfach irgendwie dargestellt. Die Reihenfolge ist egal
- Eine Sequenz wird nebeneinander dargestellt und mit Pfeilen in beide Richtungen kombiniert
- ☒ • Die Anweisungen einer Sequenz werden untereinander dargestellt und mit Pfeilen verbunden, die die "Fließrichtung" darstellen

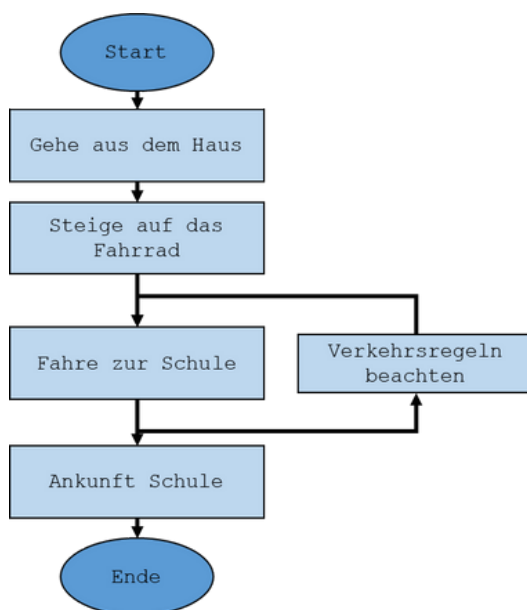
Wie werden Bedingungen dargestellt? Wie die Frage, ob dreckiges Geschirr da ist?

- Als viereckige Kästchen
- ☒ • als Rauten
- Als Dreiecke

Der Vorteil von Flussdiagrammen ist, dass sie sehr leicht lesbar sind. Man versteht schnell, was ein Algorithmus macht. Deshalb kann man mit ihrer Hilfe den Ablauf eines Algorithmus gut darstellen, bevor man ein Programm schreibt. Außerdem kann man Flussdiagramme nutzen, um anderen zu erklären, was ein Programm genau macht oder machen soll.

AUFGABE 4

Schaue dir deine Antwort von Aufgabe 1 an. Erstelle dazu ein Flussdiagramm



ALGORITHMEN



03 Algorithmen - Kontrollstrukturen

Hier schauen wir uns genauer die Kontrollstrukturen von Algorithmen an.

Kontrollstrukturen bestimmen den Ablauf von Algorithmen. Aber zuvor wiederholen wir nochmal das zuvor gelernte.

AUFGABE 5 - LÜCKENTEXT

Fülle die Lücken im folgenden Text mit den richtigen Wörtern aus. ACHTUNG: nicht alle der angegebenen Wörter passen!

Verarbeitungsvorschrift

endlichen

gleiche

eindeutig

ungleich

Ausbauvorschrift

Ein Algorithmus ist eine Verarbeitungsvorschrift, die aus einer endlich Folge von eindeutig ausführbaren Anweisungen besteht.

Unter gleichen Voraussetzungen liefert die Ausführung eines Algorithmus stets gleichen Ergebnisse

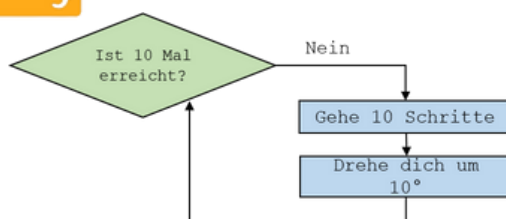
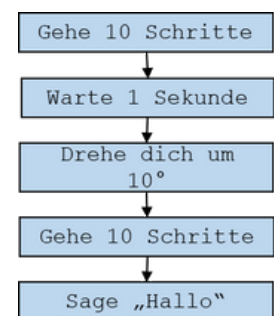
KONTROLLSTRUKTUREN VON ALGORITHMEN

Wie du in dem Flussdiagramm gesehen hast, gibt es für jedes Symbol eine andere Bedeutung. Wir sind schon den Begriffen Sequenz und Bedingung begegnet. Das sind Kontrollstrukturen eines Algorithmus, die wir uns jetzt genauer anschauen wollen.

SEQUENZ

Eine Folge von Methoden, die nacheinander ausgeführt werden, heißt **Sequenz**.

Rechts sehen wir einmal einen Code in Scratch und das dazugehörige Flussdiagramm.



SCHLEIFEN / WIEDERHOLUNGEN

Schleifen sind eine Möglichkeit, bestimmte Aufgaben wiederholt auszuführen. Eine Schleife besteht aus einer Bedingung und einem Code-Block, der ausgeführt wird, solange die Bedingung wahr ist.

Links ist wieder ein Scratch Code und ein Flussdiagramm

ALGORITHMEN

03 Algorithmen - Kontrollstrukturen



VERSCHIEDENE ARTEN VON SCHLEIFEN

Es gibt unterschiedliche Arten von Schleifen, die in unterschiedlichen Situationen mehr oder weniger Sinn machen. In der rechten Abbildung siehst du zwei verschiedene Schleifen.



AUFGABE 6

Beschreibe den Unterschied zwischen den beiden Schleifen. (Antworte in ganzen Sätzen!)

Die erste Schleife wird 10 mal durchlaufen.

Die zweite Schleife ist eine Art Dauerschleife, die solange läuft, wie das Programm läuft

BEDINGUNGEN

Das letzte was wir uns jetzt anschauen wollen sind **Bedingungen**. Die haben wir auch schon in dem Flussdiagramm kennengelernt. Bei Bedingungen wird überprüft, ob eine Voraussetzung erfüllt ist oder nicht. Je nachdem wird ein Code dann ausgeführt oder nicht.

Häufig werden auch Schleifen mit Bedingungen verknüpft. Beispielsweise wird oben die Schleife solange wiederholt, bis sie 10 Mal wiederholt wurde.



AUFGABE 7

Beschreibe was in den beiden Code-Blöcken passiert (Antworte in ganzen Sätzen)
Beim ersten Block wird geprüft, ob die Leertaste gedrückt wird.

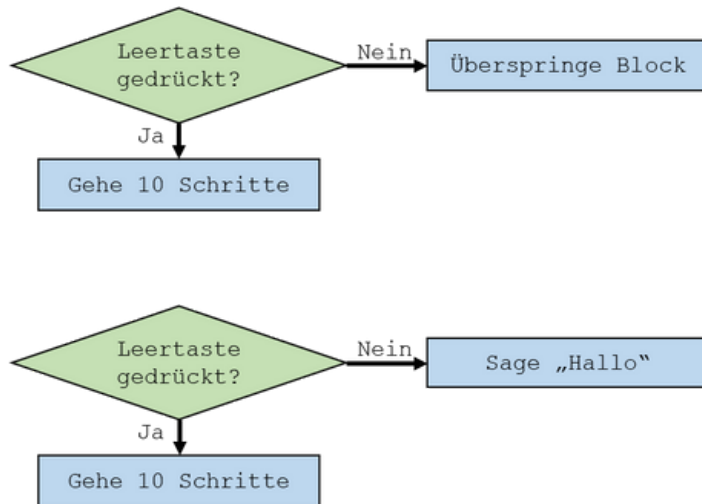
Ist diese Bedingung erfüllt, dann gehe 10 Schritte.

Beim 2. Block wird erst wieder das gleiche gemacht, wie beim ersten Block. Zusätzlich wird angegeben, was passiert, wenn die Bedingung nicht erfüllt wird: Es wird "Hallo" gesagt



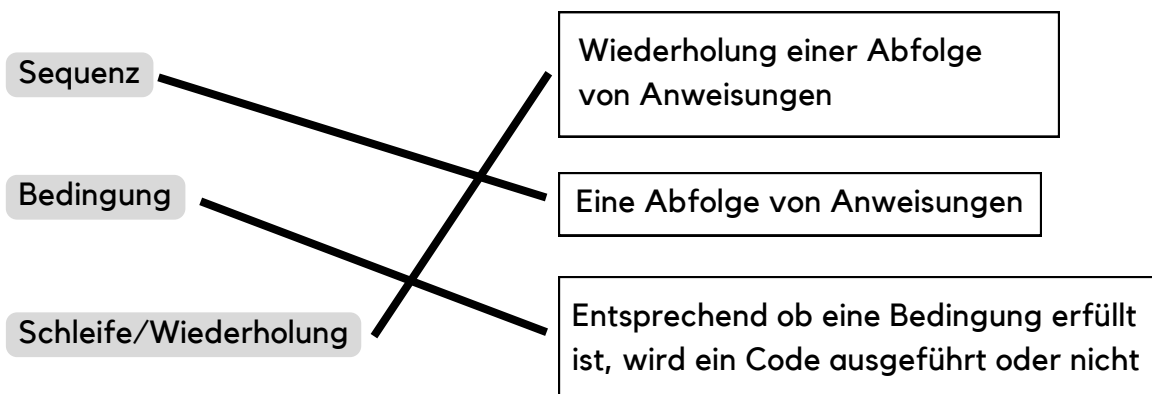
AUFGABE 8

Versuche selbst das Flussdiagramm für einen der bei den Bedingungen gezeigten Code-Blöcke zu erstellen.



AUFGABE 9

Versuche nun wieder das Gelernte anzuwenden, in dem du Paare bildest zu den gelernten Kontrollstrukturen und deren Definition.



MUSTERLÖSUNG

ALGORITHMMEN

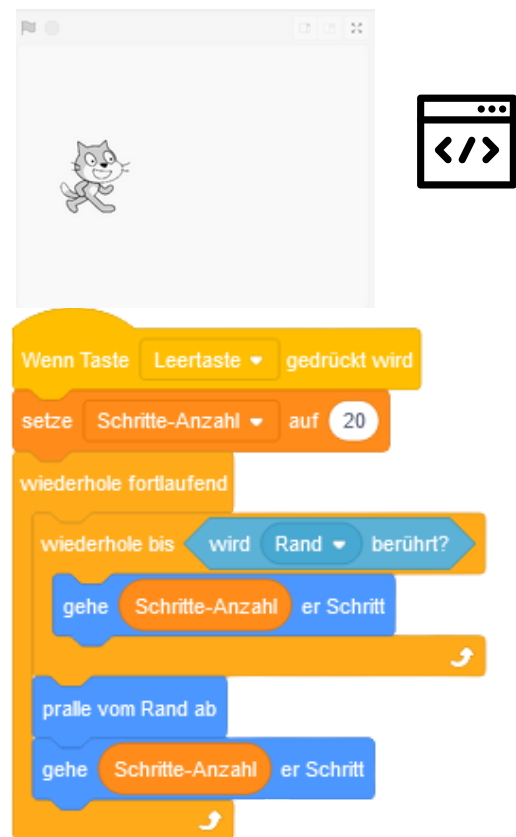
04 Variablen

Nun hast du also schon Schleifen, Sequenzen und Bedingungen kennengelernt. Eine Sache fehlt noch. Und zwar **Variablen**.

Variablen können genutzt werden, um Daten zu speichern. Rechts siehst du eine Sequenz, wo in der Variable Namens "Schritte-Anzahl", die Anzahl der Schritte gespeichert wird, die gegangen werden sollen.

Variablen in Scratch bestehen aus den folgenden Teilen:

Variablen Wert und Variablenname (hier: Schritte-Anzahl) und den Variablen-Wert (hier: 20)



Der Vorteil von Variablen ist, dass sich so Informationen für ein Programm gespeichert und genutzt werden können.

AUFGABE 10

Was macht der Algorithmus? Antworte in ganzen Sätzen

Wenn die Leertaste gedrückt ist, wird in der Variable "Schritte-Anzahl" der Wert 20 gespeichert.

Dann wird fortlaufend wiederholt:

Solange der Rand nicht berührt wird, gehe "Schritte-Anzahl"

Wird der Rand berührt, pralle vom Rand ab und gehe weiter

"Schritte-Anzahl"

Wie muss die Variable verändert werden, damit der Rand langsamer erreicht wird? Begründe deine Antwort.

Der Wert für die Variable "Schritte-Anzahl" muss kleiner sein.

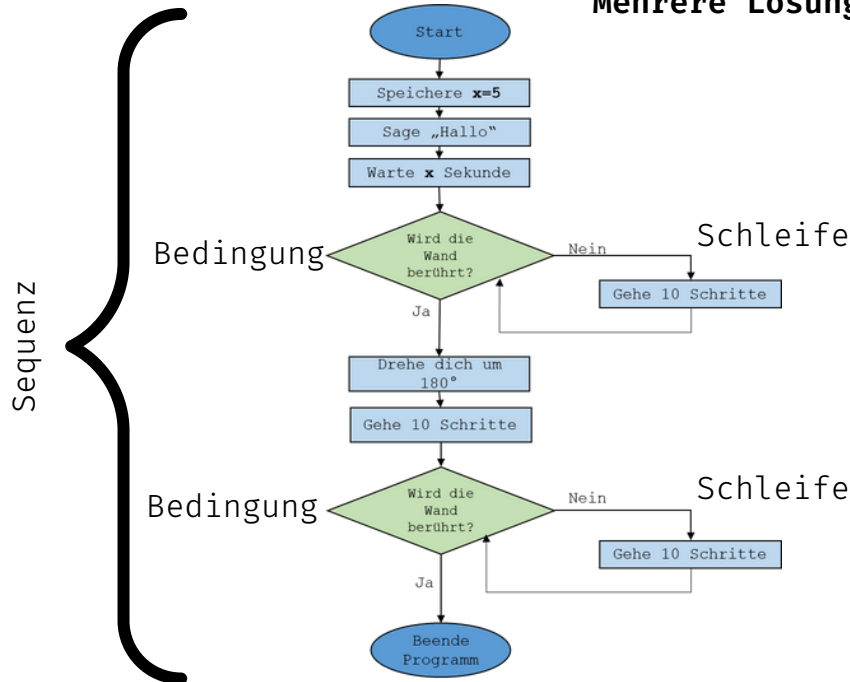
Durch einen kleineren Wert, geht die Katze weniger Schritte hintereinander und der Rand wird langsamer erreicht.



AUFGABE 11

Findest du in dem Algorithmus alle Grundstrukturen, die du kennengelernt hast: Bedingungen, Schleifen, Sequenzen und Variablen? Markiere die jeweiligen Grundstrukturen direkt an dem Algorithmus.

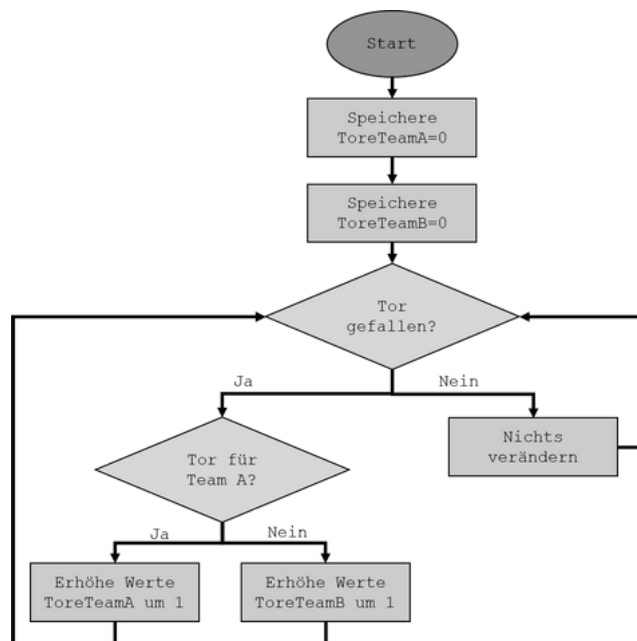
Mehrere Lösungen möglich



AUFGABE 12

Schreibe einen Algorithmus (Flussdiagramm), der von einem Fußballspiel die gefallenen Tore von Team A und Team B in jeweils einer Variable speichert.

Mehrere Lösungen möglich



ALGORITHMEN

05 Übungsaufgaben

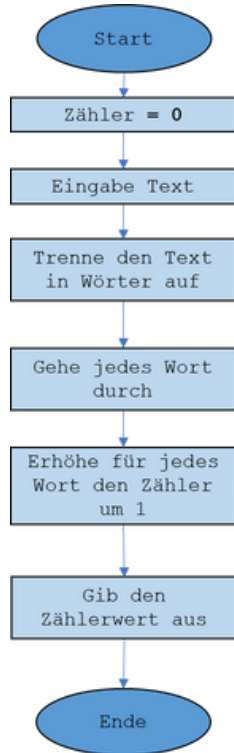
MUSTERLÖSUNG



AUFGABE 13

Gehe den folgenden Algorithmus Schritt für Schritt durch.

Beschreibe was der Algorithmus macht und notiere auch die Ausgabe



- Der Algorithmus startet mit dem Zähler von 0
- Der Text wird eingegeben
- Der Text wird in Wörtern getrennt
- Jedes Wort wird einzeln durchgegangen
- Jedes Mal, wenn Wort gefunden wird, wird der Zähler um 1 erhöht
- Sobald alle Wörter geprüft sind, wird der Zählerwert ausgegeben

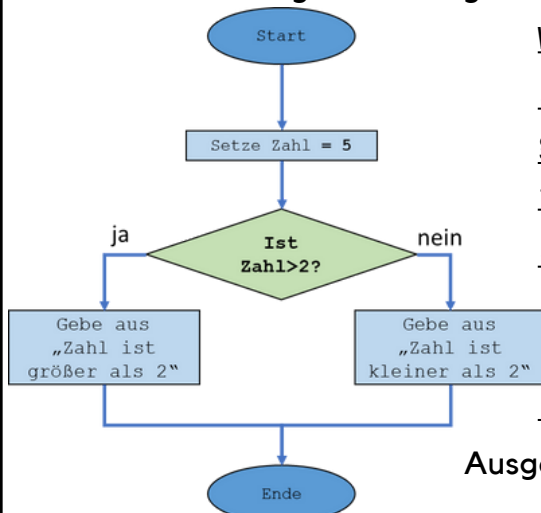
Ausgabe: Anzahl der Wörter im Text

AUFGABE 14

Schaue dir den folgenden Algorithmus an.

Beantworte die folgenden Fragen:

- Was sind die möglichen Ergebnisse, wenn die Bedingung einmal wahr und einmal falsch ist?
- Was ist die Ausgabe des Algorithmus?

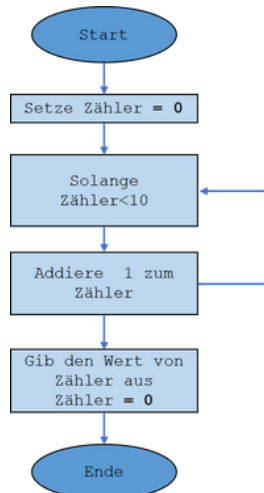


Wenn die Bedingung "Zahl > 2" wahr ist, lautet die Ausgabe "Zahl ist größer als 2". Wenn die Bedingung falsch ist, lautet die Ausgabe "Zahl ist nicht größer als 2".

Ausgabe: "Zahl ist größer als 2"



AUFGABE 15



Beschreibe was der Algorithmus macht.

- Der Algorithmus startet mit dem Zähler von 0
- Solange der Zähler einen Wert kleiner als 10 hat, wird dieser immer um 1 erhöht.
- Sobald der Zähler den Wert 10 hat, wird dieser ausgegeben

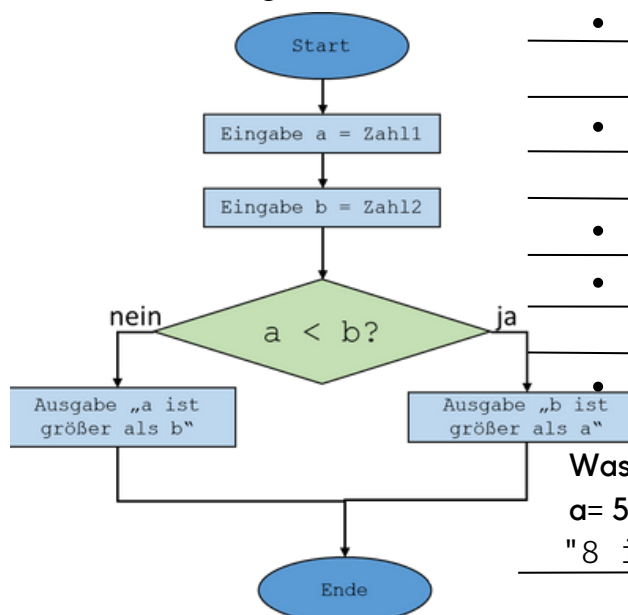
Wie häufig wird die Schleife durchlaufen?

Die Schleife wird 10 Mal durchlaufen

Was ist die Ausgabe? Die Ausgabe ist "10"

AUFGABE 16

Schau dir den Algorithmus genau an. In welcher Reihenfolge werden die Anweisungen bei der Eingabe $a=9$ und $b=3$ ausgeführt. Schreibe die Schritte auf, die nacheinander ausgeführt werden.



- Eingabe 9 wird in der Variable a gespeichert
- Eingabe 3 wird in der Variable b gespeichert
- a und b werden verglichen
- a ist größer als b, die Bedingung ist falsch
- Ausgabe "a ist größer als b"

Was ist die Ausgabe bei folgenden Eingaben:

$a=5$ $b=8$

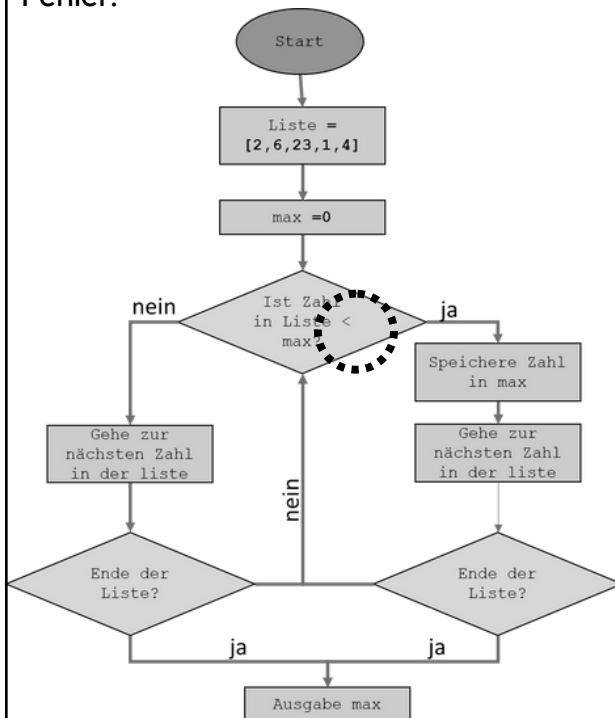
"8 ist größer als 5"

HINWEIS: Häufig werden bei solchen Ausgaben die Variablennamen direkt mit den Variablenwerten ausgegeben!

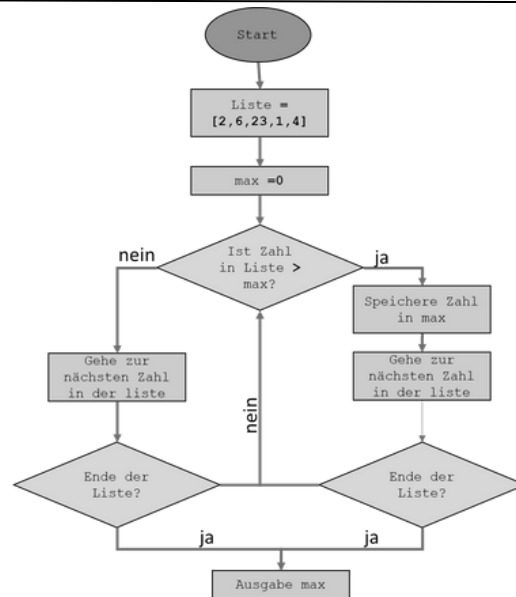


AUFGABE 17

Der folgende Algorithmus soll die größte Zahl in einer Liste von Zahlen finden. Momentan kommt aber nicht das gewollte raus. Finde, markiere und beschreibe den Fehler.



Die Bedingung überprüft, ob die Zahl in der Liste kleiner als max ist, so wird nicht die größte Zahl der Liste gefunden

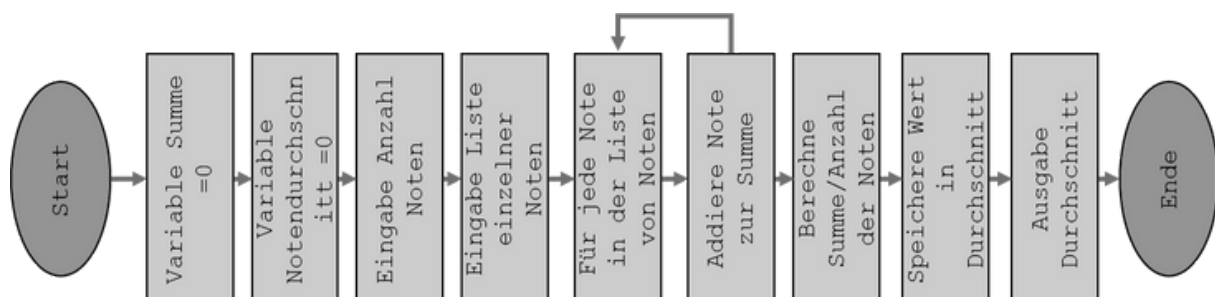


Wie muss der Algorithmus korrigiert werden, damit er das gewollte macht. Schreibe den korrigierten Algorithmus als Flussdiagramm auf. Dafür kannst du den Kasten rechts neben den falschen Algorithmus benutzen

AUFGABE 18

Entwickle einen Algorithmus zu dem folgenden Problem: Berechnung des Durchschnitts der Noten in einem Kurs. Stelle deinen Algorithmus als Flussdiagramm dar.

Mehrere Lösungen möglich





AUFGABE 19

Wir haben zwei verschiedene Sortialgorithmen zum sortieren einer Liste von Zahlen.

Algorithmus 1: Selection Sort

- Wähle das kleinste Element in der Liste und tausche es mit dem Element an erster Stelle.
- Wähle das nächstkleinste Element in der Liste und tausche es mit dem Element an zweiter Stelle.
- Wiederhole diesen Vorgang, bis die Liste sortiert ist.

Algorithmus 2: Bubble Sort

- Vergleiche das erste und zweite Element in der Liste. Wenn das zweite Element größer ist, tausche sie.
- Vergleiche das zweite und dritte Element in der Liste. Wenn das dritte Element größer ist, tausche sie.
- Wenn am ende Der Liste angekommen, fange von vorne an
- Wiederhole diesen Vorgang, bis die Liste sortiert ist.

Wir haben die folgende Liste: 4,3,7,2

Siehe die jeweiligen Algorithmen mit der Liste durch und schreibe jeden einzelnen Schritt durch. Welchen Algorithmus würdest du zum schnellen sortieren von Zahlen verwenden? **Antwort:** Selection Sort ist in diesem Fall schneller

ALGORITHMUS 1: SELECTION SORT	ALGORITHMUS 2: BUBBLE SORT
• Wähle kleinstes Element (2) der Liste. Tausche es mit dem ersten Element (4) -> [2,3,7,4]	• Vergleiche ersten 2 Elemente (4,3), 4>3 ja -> tauschen [3,4,7,2]
• Wähle nächst kleineres Element (3) und tausche es mit dem Element an der 2. Stelle -> kein tausch, da 3 schon an 2. Stelle -> [2,3,7,4]	• Vergleiche nächsten 2 Elemente (4,7), 4>7 nein -> so lassen [3,4,7,2]
• Wähle nächst kleineres Element (4) und tausche es mit dem 3. Element (7) -> [2,3,4,7]	• Vergleiche nächsten 2 Elemente (7,2), 7>2 ja -> tauschen [3,4,2,7]
• Wähle das nächst kleinere Element (7) und tausche es mit dem Element an der 4. Stelle -> kein tausch, da 7 schon an 4. Stelle	• Ende der Liste, fange von vorne an
• Fertig	• Vergleiche ersten 2 Elemente (3,4) 3>4 nein -> so lassen [3,4,2,7]
	• Vergleiche nächsten 2 Elemente (4,2), 4>2 ja -> tauschen [3,2,4,7]
	• vergleiche nächsten 2 Elemente (4,7), 4>7 nein -> so lassen [3,2,4,7]
	• Fertig

MUSTERLÖSUNG ALGORITHMEN



06 Test

Kreuze die richtige Antwort an.

Was ist ein Algorithmus?

- a) Ein mathematisches Problem
- ☒ b) Eine endliche Folge von eindeutigen und ausführbaren Verarbeitungsschritten
- c) Eine Liste von Zufallszahlen

Was bedeutet Endlichkeit in Bezug auf Algorithmen?

- a) Ein Algorithmus besteht aus einer unendlichen Anzahl von Schritten
- ☒ b) Ein Algorithmus hat nur eine begrenzte Anzahl von Schritten
- c) Endlichkeit hat in Bezug auf Algorithmen keine Bedeutung

Was bedeutet Eindeutigkeit in Bezug auf Algorithmen?

- ☒ a) Ein Algorithmus hat nur eine einzige Schrittfolge, die zum Ziel führt
- b) Ein Algorithmus kann auf viele verschiedene Arten ausgeführt werden
- c) Ein Algorithmus muss keine eindeutige Schrittfolge haben

Was ist eine Sequenz in einem Algorithmus?

- a) Eine Wiederholung eines Teils des Algorithmus
- ☒ b) Eine Reihe von Anweisungen, die hintereinander ausgeführt werden
- c) Eine Abfrage, die eine Entscheidung trifft, welche Anweisung ausgeführt wird

Was ist eine Bedingung in einem Algorithmus?

- a) Eine Anweisung, die nur einmal ausgeführt wird
- b) Eine Anweisung, die wiederholt ausgeführt wird
- ☒ c) Eine Abfrage, die eine Entscheidung trifft, welche Anweisung ausgeführt wird

Was ist eine Schleife in einem Algorithmus?

- a) Eine Anweisung, die nur einmal ausgeführt wird
- ☒ b) Eine Anweisung, die wiederholt ausgeführt wird
- c) Eine Abfrage, die eine Entscheidung trifft, welche Anweisung ausgeführt wird

ALGORITHMEN

07 Extra-Aufgaben



Entwickle einen Algorithmus für jeden der folgenden Problemen:

1. Ein Algorithmus zur Konvertierung von Grad cm in mm.
2. Ein Algorithmus zur Bestimmung der Anzahl der Wörter in einem Textdokument.
3. Ein Algorithmus zur Suche nach einem bestimmten Element in einer Liste.
4. Ein Algorithmus zur Berechnung der Fläche eines Rechteckes.