

# Themen

---

Grundlegende Suchalgorithmen

8

---

Sortieralgorithmus Bubble Sort

9

---

Listen anlegen und befüllen

10

---

Listen anwenden

11

---

Lineare Suche

12

---

Bubble Sort

13

---

Logische Verknüpfungen

14

---

Zufallszahlen

15

---

Unterprogramme

---

16

# Struktogramme

Die ersten Schritte auf dem Weg zu einer komplexen Software sind eine sorgfältige Planung und eine strukturierte Vorgehensweise. Dabei geht es darum, eine komplexe Aufgabenstellung in mehrere kleinere Problemstellungen zu zerlegen, die jede für sich einfacher zu lösen sind.

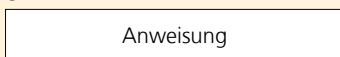
Eine anschauliche Methode zur Zerlegung und übersichtlichen Darstellung von Programm-entwürfen sind die so genannten Struktogramme.

Das Wort Struktogramm ist ein Kofferwort aus Struktur und Diagramm. Nach seinen Erfindern, den US-amerikanischen Informatikern Isaac Nassi und Ben Shneiderman werden Struktogramme auch Nassi-Shneiderman-Diagramme genannt.

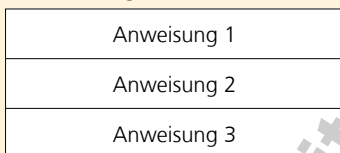
Struktogramme werden stets programmiersprachenunabhängig formuliert und sind dadurch leicht zu lesen. Sie helfen, den Ablauf eines Programms zu durchdenken und Logikfehler zu vermeiden. Im Struktogramm werden auch die benötigten Variablen und Bedingungen sowie alle vom Benutzer zu tätigen Eingaben definiert.

Für den Aufbau von Struktogrammen gibt es Regeln, die auch in einer DIN-Norm festgeschrieben sind.

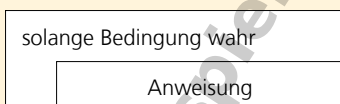
Anweisungen werden in rechteckige Blöcke geschrieben.



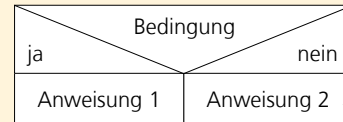
Mehrere aufeinander folgende Anweisungen werden direkt untereinander platziert. Im Programm werden sie später in genau dieser Reihenfolge durchlaufen.



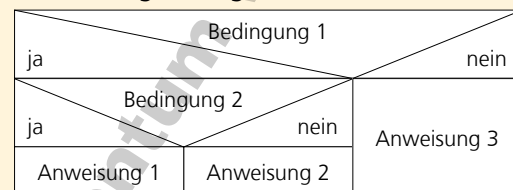
Wiederholungen werden als Winkel dargestellt, die eine oder mehrere Anweisungen umschließen.



Die Abfrage von Bedingungen wird als Verzweigung dargestellt. Nur wenn die Bedingung zutreffend (wahr) ist, wird der Anweisungsblock 1 durchlaufen. Trifft die Bedingung nicht zu (unwahr), wird Anweisung 2 ausgeführt.



In Struktogrammen sind auch mehrfache Verzweigungen möglich. Nur wenn Bedingung 1 zutrifft, wird die Abfrage 2 mit Bedingung 2 ausgeführt. Trifft Bedingung 1 nicht zu, wird stattdessen Anweisung 3 ausgeführt.



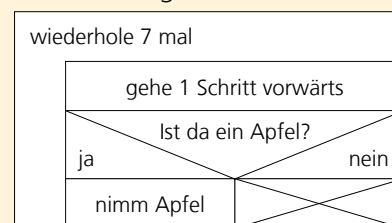
Die einzelnen Blöcke eines Struktogramms werden jeweils bündig aneinandergesetzt. Die Breite des ganzen Struktogramms richtet sich nach dem breitesten Block.

Das Prinzip des Struktogramms wird anhand einer einfachen Beispielaufgabe leicht verständlich.

Der Roboter soll geradeaus bis auf das gelbe Feld laufen. Wenn auf einem Feld ein Apfel liegt, soll er ihn aufheben.



Das Struktogramm für dieses Beispiel sieht so aus:

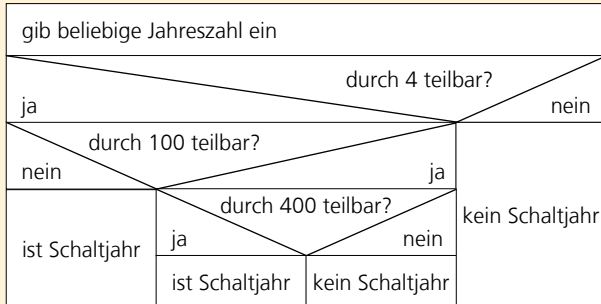


Wenn auf einem Feld kein Apfel liegt, soll der Roboter nichts tun. In diesem Fall wird in den Block der Anweisung ein Kreuz gezeichnet.

# Struktogramme

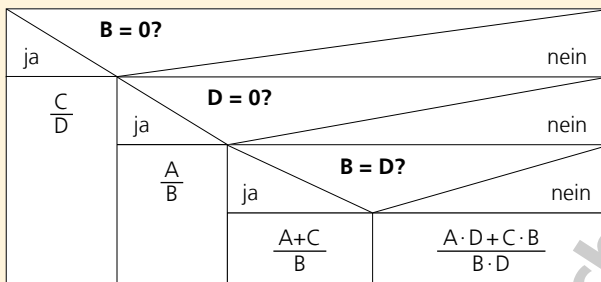
## Aufgabe 1

Was macht der in diesem Struktogramm dargestellte Algorithmus?



## Aufgabe 2

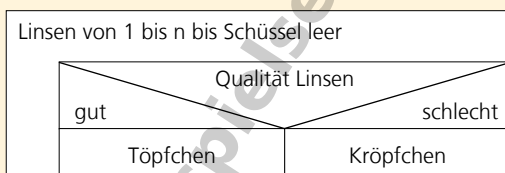
Beschreibe den im Struktogramm dargestellten Algorithmus für das Addieren der beiden Brüche  $\frac{A}{B} + \frac{C}{D}$ .



## Aufgabe 3

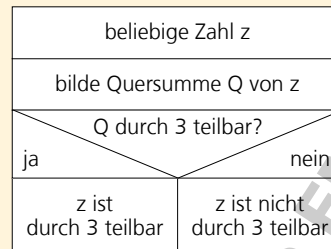
Im gleichnamigen Märchen der Brüder Grimm ruft Aschenputtel die Vögel zu Hilfe: „Ihr zahmen Täubchen, ihr Turteltäubchen, all ihr Vöglein unter dem Himmel kommt und helft mir lesen, die guten ins Töpfchen, die schlechten ins Kröpfchen.“

Beschreibe, mit Hilfe welcher Blöcke sich die Aufgabe der Vögel in einem Struktogramm darstellen lässt.



## Aufgabe 4

Beschreibe die im Struktogramm dargestellte Vorgehensweise beim Ermitteln, ob eine beliebige Zahl z durch 3 teilbar ist.



## Aufgabe 5

Zeichne ein Struktogramm für folgende Situation: Bevor du aus dem Haus gehst, schaust du, ob es regnet. Ist das der Fall, ziehst du deine Regenjacke an. Ist es trocken, ziehst du eine Strickjacke an.

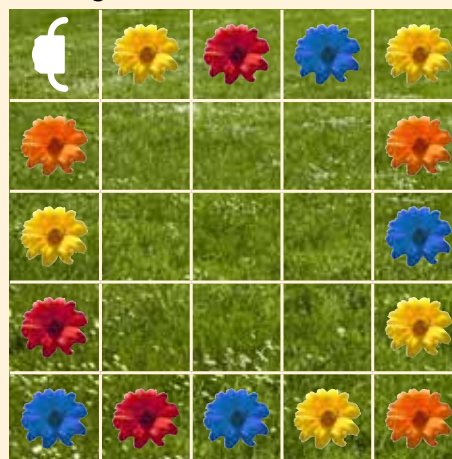
## Aufgabe 6

Zeichne ein Struktogramm für die Vorgehensweise beim Ermitteln, ob eine Zahl durch 6 teilbar ist.

## Aufgabe 7

Der Roboter soll auf dem Blumenweg einmal ringsherum laufen und alle blauen Blumen pflücken. Auf dem letzten Feld darf er sich auch noch einmal drehen.

Zeichne ein Struktogramm für diese Aufgabenstellung.



# Abfragen und Bedingungen

## Wiederholungen und Programmende



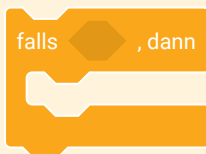
Mit diesem Block gibt man die Anweisung, dass ein Befehl oder eine Reihe von Befehlen wiederholt werden soll, und zwar so lange, bis das Skript durch einen Befehl gestoppt wird.

Für das Stoppen von Skripten kann dieser Block verwendet werden.



Das Ausklappmenü bietet die Möglichkeit, nur das aktuelle Skript, nur die anderen Skripte der aktuellen Figur oder alles zu stoppen.

## Abfragen und Bedingungen



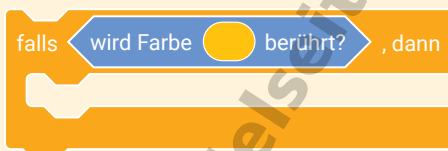
Sollen Befehle nur ausgeführt werden, wenn eine Bedingung erfüllt ist, wird dieser Block eingesetzt.

Die Bedingungen werden mit Hilfe sechseckiger Blöcke wie diesem angegeben:

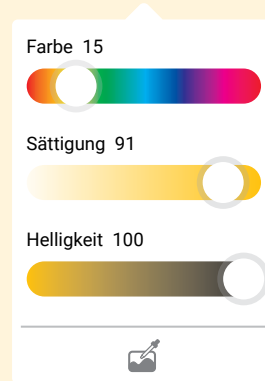


Er legt fest, dass die nachfolgenden Befehle nur ausgeführt werden, nachdem die Figur die Farbe Gelb berührt hat.

Der Bedingungs-Block wird in das sechseckige Feld im Abfrage-Block eingefügt.



Die in der Bedingung enthaltene Farbe kann frei gewählt werden. Mit einem Klick in das Farbfeld öffnet sich ein Menü zum Einstellen der Farbe. Die Pipette ermöglicht das Übernehmen einer beliebigen Farbe aus dem gewählten Bühnenbild.



Sollen Befehle nur ausgeführt werden, sobald eine Taste gedrückt wird, verwendet man diesen Block.



Durch einen Klick auf das Dreieck öffnet sich ein Ausklappmenü, in dem weitere mögliche Tasten gewählt werden können.

## Zeigen und Verstecken



Mit Hilfe dieser beiden Blöcke kann eine Figur während eines Skriptes ein- und ausgeblendet werden.

## Meldungen



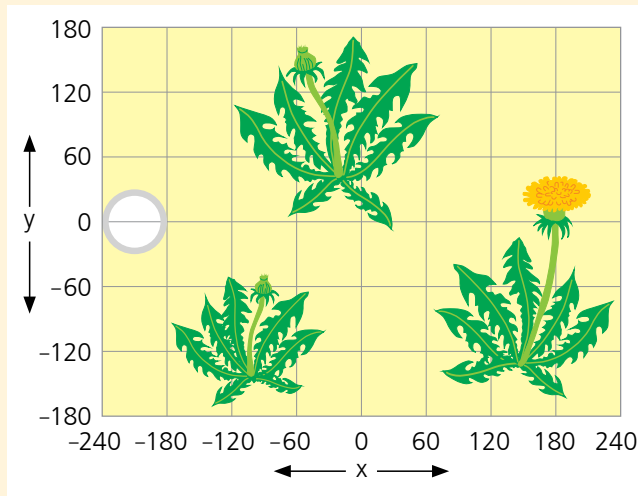
Mit diesem Block kann man eine Figur eine Meldung oder Mitteilung ausgeben lassen. Sie wird in das Parameterfeld eingegeben und erscheint während des Skriptes als Sprechblase.



# Abfragen und Bedingungen

## Bühnenbild

Loewenzahn.svg




## Figur

Seifenblase.svg




Beachte, dass die Figur sich nicht mit ihrem linken oder rechten Rand, sondern mit ihrem Mittelpunkt auf den eingestellten Koordinaten platziert.

### Aufgabe 1

Sobald auf die grüne Fahne  geklickt wird, soll die Seifenblase in der Mitte des grauen Ringes erscheinen.

### Aufgabe 2

Sobald die Cursortasten gedrückt werden, soll sich die Seifenblase einen 10er-Schritt in die jeweilige Richtung bewegen.

Platziere dafür neben dem Skript aus Aufgabe 1 ein weiteres Skript, das ebenfalls mit einem Klick auf die grüne Fahne  gestartet wird. Die Abfragen, ob eine Taste gedrückt wird, sollen fortlaufend ausgeführt werden.

### Aufgabe 3

Sobald die Seifenblase ein grünes Löwenzahnblatt berührt, soll sie platzen (also verschwinden). Füge diese Abfrage zu dem Skript aus Aufgabe 1 hinzu. Die Abfrage soll fortlaufend ausgeführt werden.

### Aufgabe 4

Füge eine weitere Anweisung in das Skript aus Aufgabe 3 ein: Sobald die Seifenblase ein grünes Löwenzahnblatt berührt hat und daraufhin geplatzt (also verschwunden) ist, soll das gesamte Skript gestoppt werden. (Sobald das Skript gestoppt ist, verschwindet der gelbe Saum um das Skript herum.)

### Aufgabe 5

Sobald die Seifenblase die gelbe Löwenzahnblüte berührt, soll sie platzen (also verschwinden). Auch diese Abfrage soll fortlaufend ausgeführt werden. Füge diese Abfrage zu dem Skript aus Aufgabe 4 hinzu.

### Aufgabe 6

Da es gar nicht so einfach ist, die Seifenblase bis zur Löwenzahnblüte zu manövrieren, soll zwei Sekunden lang die Meldung „Super!“ erscheinen, bevor die Seifenblase an der Löwenzahnblüte zerplatzt (also verschwindet). Füge diese Anweisung zu dem Skript aus Aufgabe 5 hinzu.