

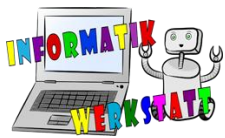
Dieses Heft begleitet dich durch die Stationen der Informatik-Werkstatt. Du findest darin Erklärungen und Aufgaben zu einigen spannenden Themen der Informatik.

Erlebnisheft

Dein Name: _____

Datum: _____





Was finde ich im Erlebnisheft?

CODIEREN	3
BINÄRCODES.....	5
BINÄRZAHLEN	6
VERSCHLÜSSELUNG	9
LOGIK.....	11
HARDWARE	14
MODELLIEREN	16
ALGORITHMEN und PROGRAMME	19
ROBOTER.....	22

CODIEREN



Die **Codierung** ist eines der wichtigsten Themen der Informatik. Ohne sie könnten wir keine Maschinen, zum Beispiel einen Computer, programmieren. Beim **Codieren** geht es darum, Symbole eines Alphabets, wie zum Beispiel die Buchstaben von A bis Z, durch andere Symbole zu ersetzen.

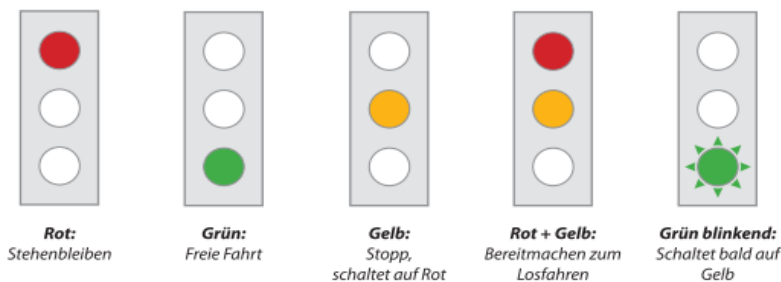


Abbildung 1: Ampel als Code (aus „Informatik erLeben“)

Es gibt verschiedene Arten von **Codes**. So stellen zum Beispiel die Farben der **Ampel** einen **Code** dar. Die Farbe **Rot** steht dabei für „Stehenbleiben!“, **Grün** für „Freie Fahrt!“, **Gelb** für „Stopp, Ampel schaltet auf

Rot!“, Rot und Gelb gleichzeitig bedeutet „Bereitmachen zum Losfahren!“ und ein grün blinkendes Signal heißt „Ampel schaltet bald auf Gelb“.

Ein weiteres Beispiel, das euch sicher bekannt vorkommt, ist der nach seinem Erfinder Samuel Morse benannte **Morsecode**. Durch diesen wurde Kommunikation über weitere Strecken ermöglicht. Damit war der **Morsecode** zusammen mit einer Morsetaste der Vorreiter des Telefonierens. Dieser **Code** unterscheidet zwischen einem **langen Signal**, hier als **Strich** dargestellt, und einem **kurzen Signal**, durch einen **Punkt** dargestellt. In der Tabelle rechts findest du einen Teil des **Morsealphabets**.

Wie du siehst wird dabei jeder Buchstabe durch eine Folge von kurzen oder langen Signalen ersetzt.

A	· —	N	— ·
B	— ...	O	— — —
C	— · — ·	P	· — — ·
D	— ..	Q	— — · —
E	·	R	· — ·
F	· · — ·	S	... ·
G	— — ·	T	— —
H	U	· · —
I	··	V	... —
J	· — — —	W	· — —
K	— · —	X	— · —
L	· — ..	Y	— · — —
M	— —	Z	— — ..

Abbildung 2: Morsealphabet

Wenn du nun ein Wort bilden möchtest, musst du jeden Buchstaben einzeln codieren und dann zusammenfügen.

Ein sehr bekanntes Beispiel ist der **Code** für Notrufe S O S:

S	O	S
...	---	...

Dabei wird einfach in der Tabelle das „S“ gesucht und durch den entsprechenden **Code** ersetzt. Gleiches gilt für das „O“. Nun kannst du selbst schon etwas **codieren**.

Aufgabe 1: Versuche diesen Code zu decodieren und schreibe die Lösung darunter:

..|— ·|.. — ·|— — — |· — ·|— — |· — |— |·|— · —

Aufgabe 2: Schreibe deinen eigenen Namen als Morsecode auf.

Aufgabe 3: Übermittle einem anderen ein Wort im Morsecode, indem du auf den Tisch klopfst oder mit der Taschenlampe blinkst.

Ein weiteres Beispiel, das jeder kennt, ist der **Strichcode** oder „**Europäische Artikel Nummer (EA-Nummer)**“, der auf allen Produkten im Supermarkt zu finden ist. Mithilfe diese Codes kann jedes Produkt und der jeweilige Preis durch einen Scanner an der Kasse erkannt werden. So werden bei einem großen Einkauf schnell alle Preise der gescannten Produkte zusammengezählt.



Abbildung 3: Europäische Artikel Nummer

Auch der heute sehr gerne verwendete **QR-Code** (Abkürzung für „Quick Response“ – schnelle Antwort) ist eine etwas kompliziertere Form der Codierung.



Abbildung 4: QR-Code

BINÄRCODES



Um mit einem Computer zu kommunizieren, muss zuerst bedacht werden, dass er nur **zwei Zustände** kennt: **Spannung** oder **keine Spannung**, also **Strom fließt** oder **Strom fließt nicht**. Aus diesem Grund müssen wir einen **Code** finden, durch den wir jeden Buchstaben und jede Zahl durch nur zwei Symbole darstellen können. Dafür werden die **1** für „**Strom fließt**“ und die **0** für „**Strom fließt nicht**“ verwendet. Mit diesen beiden Ziffern können alle Buchstaben unseres Alphabets **codiert** und alle Zahlen umgerechnet werden. Wie das funktioniert, sehen wir uns nun an.

Um die Buchstaben mit Hilfe von 0 und 1 zu **codieren**, brauchen wir wiederum eine Tabelle, die so genannte **ASCII** („American Standard Code for Information Interchange“ – Amerikanischer Standardcode für Informationsaustausch) Tabelle. Darin wird jedem Buchstaben eine mehrstellige **Binärzahl** (besteht nur aus 0 und 1) zugewiesen. Dabei wird darauf geachtet, dass jede Binärzahl sieben Stellen hat. Einen Ausschnitt dieser Tabelle seht ihr hier:

Buchstabe	Binärzahl	Buchstabe	Binärzahl	Buchstabe	Binärzahl	Buchstabe	Binärzahl
A	1000001	H	1001000	O	1001111	V	1010110
B	1000010	I	1001001	P	1010000	W	1010111
C	1000011	J	1001010	Q	1010001	X	1011000
D	1000100	K	1001011	R	1010010	Y	1011001
E	1000101	L	1001100	S	1010011	Z	1011010
F	1000110	M	1001101	T	1010100	_	1011111
G	1000111	N	1001110	U	1010101	-	0101101

Aufgabe 4: Versuche diesen Code zu decodieren und schreibe die Lösung darunter:

1010100 1000101 1001100 1000101 1000110 1001111 1001110

Aufgabe 5: Schreibe deinen Namen mit Hilfe von 0 und 1.

BINÄRZAHLEN

Umwandeln ins binäre Zahlensystem

Um Zahlen mithilfe von 0 und 1 darzustellen, geht man etwas anders vor. Es gibt nämlich neben dem 10er-System (Dezimalsystem), in dem wir meistens rechnen, noch andere Systeme. Eines davon heißt „**Binärsystem**“ und verwendet statt den 10 Ziffern, von 0 bis 9, nur die **zwei Ziffern 0 und 1**.

Beim **Umwandeln einer binären Zahl** in eine Zahl unseres 10er-Systems helfen wir uns mit Punktekärtchen. Jedes Punktekärtchen gehört zu einem Stellenwert und so wird, bei der Ziffer ganz rechts beginnend, unter jede Ziffer ein Punktekärtchen gelegt.

Hier ein Beispiel mit 11011_2 :

1	1	0	1	1
Oder: Alle Punktkarten mit 1 oberhalb bleiben aufgedeckt, alle anderen werden umgedreht.				

Nun werden all jene Punkte der Punktekärtchen zusammengezählt, über denen eine 1 steht. In dem Beispiel sind das alle Kärtchen außer das mit vier Punkten in der Mitte. Alle Punkte zusammen ergeben dabei 27. Also ist die Binärzahl 11011_2 gleich der Dezimalzahl 27.

Aufgabe 6: Berechne die Dezimalzahlen von 101_2 , 1101_2 und 10101_2 .

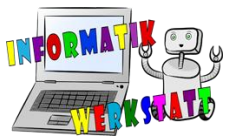
Das Umrechnen einer Zahl aus dem 10er-System in das Binärsystem funktioniert so, dass du die Zahl immer durch 2 dividierst bis du bei 0 oder 1 angelangt bist. Wichtig ist, dass du dir den Rest aufschreibst, weil der ergibt von rechts nach links gelesen die gesuchte Binärzahl. Oder du überlegst dir, welche Punktekarten du für die Binärzahl brauchst.

Ein **Beispiel:**

$$34 : 2 = 17 : 2 = 8 : 2 = 4 : 2 = 2 : 2 = 1 : 2 = 0$$

$$0 \text{ R } 1 \text{ R } 0 \text{ R } 0 \text{ R } 0 \text{ R } 1 \text{ R } \quad \longrightarrow \quad 100010_2$$

Hier muss der Rest von rechts nach links gelesen werden.



Aufgabe7: Stelle die Zahlen 7, 15 und 56 als Binärzahlen dar.

VERSCHLÜSSELUNG

Der römische Feldherr Julius **Caesar** (100 bis 44 v. Chr.) verschlüsselte seine **geheimen** Nachrichten, indem er jeden Buchstaben durch einen anderen ersetzte. Dabei wurde der Buchstabe immer durch den um eine bestimmte Anzahl von Stellen im Alphabet verschobenen Buchstaben ersetzt. Diese Anzahl der Stellen heißt **Caesar-Schlüssel**.

Beispiel: Beim Schlüssel **2** wird immer der Buchstabe genommen, der im Alphabet **zwei** Stellen weiter **rechts** steht.

Dazu schreibt man das Alphabet zweimal untereinander. Das untere Alphabet wird allerdings um zwei Stellen verschoben.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B

Die grünen Buchstaben sind der **KLARTEXT**, die roten Buchstaben sind der **VERSCHLÜSSELTE TEXT**.

Beispiele:

KINDER → MLPFGT

DAS IST GEHEIM → FCU KUV IGJGKO

Verschlüsseln und Entschlüsseln mit dem Caesar-Rad:

Stelle den Caesar-Code (1-25) mit der inneren Scheibe ein. Nimm den **Klartext** und schaue jeden Buchstaben auf der **inneren Scheibe** nach. Auf der **äußeren Scheibe** steht der entsprechende **Geheimtext**.

Neben der Caesar-Verschlüsselung gibt es viele andere Verfahren zur Verschlüsselung. Eine einfache Variante ist das **gespiegelte Alphabet**:

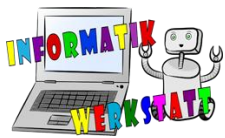
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Z	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A

Beispiele:

KINDER → PRMWVI

DAS IST GEHEIM → WZH RHG TVSVRN

Aufgabe 1: Verschlüsse die Wörter der linken Seite mit der Cäsar-Chiffre und dem Schlüssel **2**. Entschlüsse auch die beiden verschlüsselten Texte.



Aufgabe 8: Ergänze die leeren Zeilen mit Wörtern im Klartext und verschlüsselten Text.

Klartext	Verschlüsselter Text
guten Tag	
hallo Kinder	
	dtcxg mncuug
	pcejtkejv

Aufgabe 9: Der Schlüssel ist **1**. Verwende das Caesar-Rad und entschlüssele den Text:

Verschlüsselter Text: kvmjvt dbftbs

Entschlüsselter Text:

Aufgabe 10: Wähle selbst ein Wort und einen Schlüssel:

Schlüssel:

Verschlüsseltes Wort:

Entschlüsselter Wort:

LOGIK



Nach dem ziemlich mathematischen Thema Binärzahlen kommen wir zu einem anderen Thema. Hier geht es mehr um Sprache und logisches Denken. Unter **Logik** wird oft vernünftiges Schlussfolgern verstanden. Dabei wird überprüft, ob **Aussagen** „wahr“ oder „falsch“ sind. Aussagen sind Sätze, die entweder „wahr“ oder „falsch“ sein können.

Aufgabe 11: Gib in der Tabelle an, ob folgende Aussagen „wahr“ oder „falsch“ sind!

	wahr	falsch		wahr	falsch
„Draußen liegt Schnee.“			„Ich bin in der Informatikwerkstatt.“		
„Es ist Sommer.“			„Wir tragen alle Badehosen.“		

Durch so genannte **Bindewörter** können **Aussagen** miteinander verbunden werden. Das sind zum Beispiel **„UND“** und **„ODER“**. Dadurch, dass man Aussagen verbindet, werden sie zu einer neuen Aussage, die wieder „wahr“ oder „falsch“ sein kann. Ob sie „wahr“ oder „falsch“ sind, hängt dann von den einzelnen Aussagen und dem Bindewort ab. So ist eine durch **„UND“** verbundene Aussage nur dann „wahr“, wenn auch alle einzelnen Aussagen „wahr“ sind. So ist die Aussage *„Draußen liegt Schnee **UND** es ist Sommer.“* nur dann „wahr“, wenn „Draußen liegt Schnee.“ und auch „Es ist Sommer.“ wahr sind.

Aufgabe 12: Gib an, ob folgende verbundene Aussagen „wahr“ oder „falsch“ sind!

	wahr	falsch
„Draußen liegt Schnee UND es ist Sommer.“		
„Ich habe Ferien UND die Schulen sind geschlossen.“		
„Ich bin in der Informatikwerkstatt UND trage Badehosen.“		

Aussagen die durch ein „ODER“ verbunden werden, sind dann „wahr“, wenn mindestens eine der einzelnen Aussagen „wahr“ ist, aber auch wenn mehr als eine Aussage „wahr“ sind. Bei unserem Beispiel würde das bedeuten, dass „*Draußen liegt Schnee ODER es ist Sommer.*“ dann „wahr“ ist, wenn draußen Schnee liegt, wenn es Sommer ist oder wenn beides zutrifft.

Aufgabe 13: Gib an, ob folgende verbundene Aussagen „wahr“ oder „falsch“ sind!

	wahr	falsch
„Draußen liegt Schnee ODER es ist Sommer.“		
„Ich habe Ferien ODER die Schulen sind geschlossen.“		
„Ich bin in der Informatikwerkstatt ODER trage Badehosen.“		

Um die Übersicht zu behalten, kann man eine so genannte Wahrheitstafel aufschreiben. Darin wird festgehalten, welche Aussage „wahr“ und welche „falsch“ ist. Dabei schreiben wir statt „wahr“ 1 und statt „falsch“ 0. Für unser Beispiel sieht sie dann so aus:

UND		
„Draußen liegt Schnee.“	„Es ist Sommer.“	„Draußen liegt Schnee UND es ist Sommer.“
1	1	1
1	0	0
0	1	0
0	0	0

ODER		
„Draußen liegt Schnee.“	„Es ist Sommer.“	„Draußen liegt Schnee ODER es ist Sommer.“
1	1	1
1	0	1
0	1	1
0	0	0

Aufgabe 14: Finde eigene Aussagen und verbinde sie mit den Bindewörtern „UND“ und „ODER“. Sind die verbundenen Aussagen „wahr“ oder „falsch“?

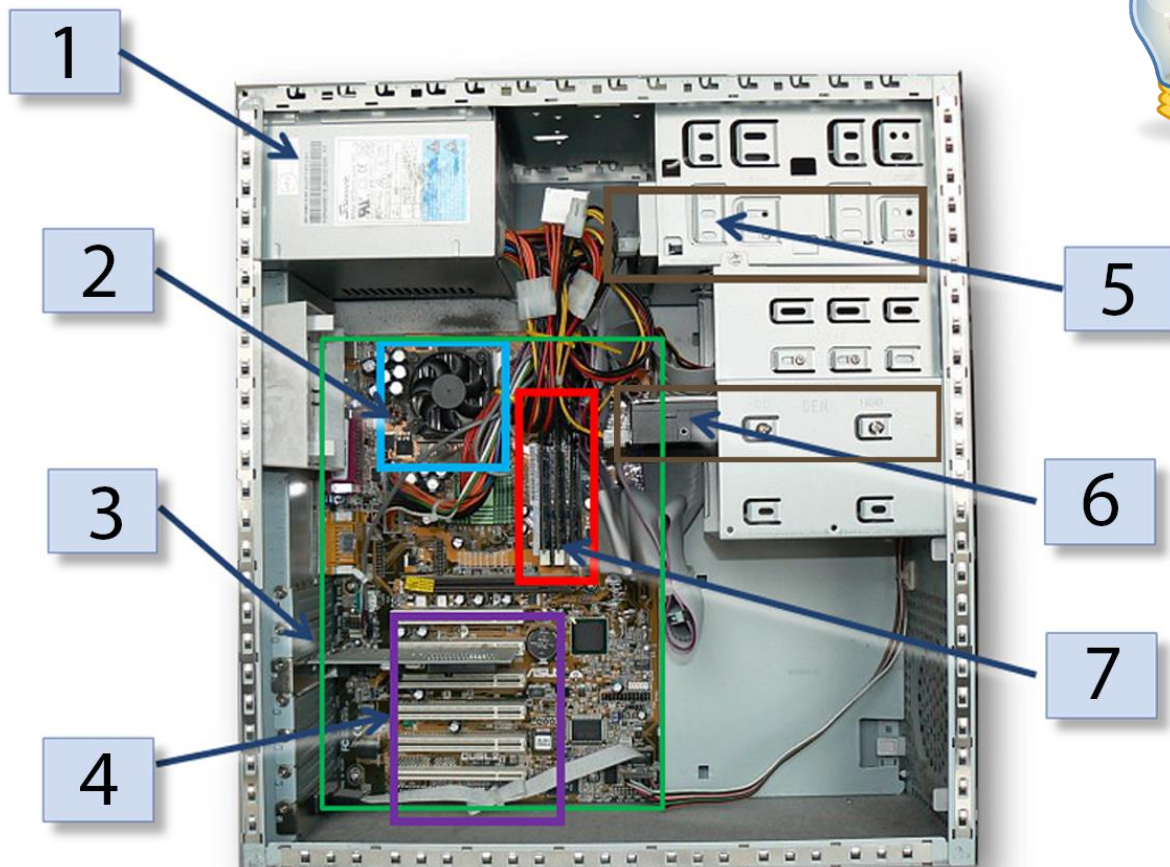
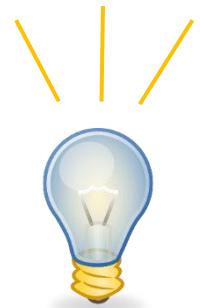
Aufgabe 15: Fülle folgende Wahrheitstafeln mit fehlenden 1 und 0 auf.

UND		
„Ich habe ein Geschenk“	„Ich bin verkleidet.“	„Ich habe ein Geschenk UND ich bin verkleidet.“
1	1	
1	0	
0	1	
0	0	

ODER		
„Ich habe ein Geschenk“	„Ich bin verkleidet.“	„Ich habe ein Geschenk ODER ich bin verkleidet.“
1	1	
1	0	
0	0	

HARDWARE

In der Abbildung seht ihr das „Innenleben“ eines Computers, also seine Hardwarekomponenten. Da nun jeder von euch ein Experte für eines dieser Teile ist, besteht eure Aufgabe nun darin, gemeinsam die einzelnen Komponenten zu bestimmen und die jeweilige Nummer in der unten stehenden Tabelle einzutragen. Zusätzlich nimmt zu diesem Zweck auch das Gehäuse von dem PC vor euch ab und versucht die jeweiligen Teile auch dort zu identifizieren.



Netzteil,

Grafikkarte und Soundkarte
Prozessor

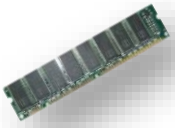






Motherboard

Arbeitsspeicher

Festplatte

CD-ROM-Laufwerk

Aufgabe 16: Schreibe die richtige Nummer hinein und ergänze die Wörter:

Nummer	Beschreibung
	<p>Hier befindet sich der _____</p> 
	<p>Hier befindet sich das _____</p> 
	<p>Hier befindet sich der _____</p> 
	<p>Hier befinden sich die Steckplätze für die _____ und die _____</p> 
	<p>Hier befindet sich das _____</p> 
	<p>Hier befindet sich die _____</p> 
	<p>Hier befindet sich das _____</p> 

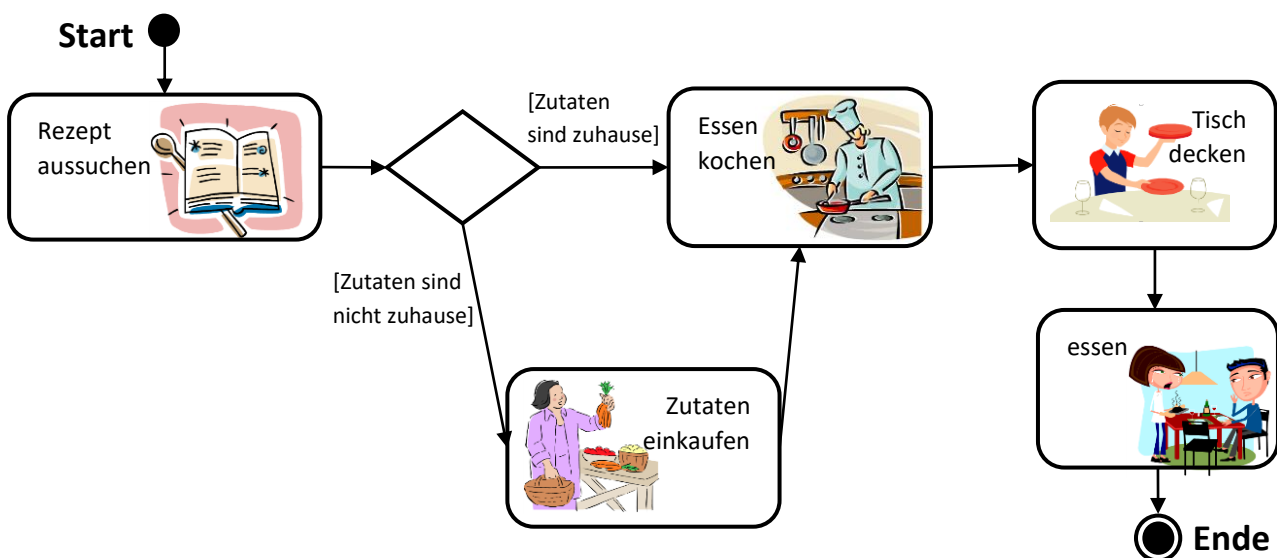
MODELLIEREN



Ein **Modell** ist eine **Abbildung der Wirklichkeit**. Das Herstellen eines Modells heißt **Modellieren**. Modelle sollen **Ereignisse übersichtlich und anschaulich darstellen**. Für die Informatik ist die Modellierung als **Vorbereitung** für alle möglichen Projekte zu sehen. Vergleichbar dazu sind die mathematischen Berechnungen und Zeichnungen eines Architekten, bevor eine Brücke gebaut wird. In der Informatik werden viele verschiedene Modelle eingesetzt. Jedes davon erfüllt eine andere Aufgabe.

Aktivitätsdiagramme

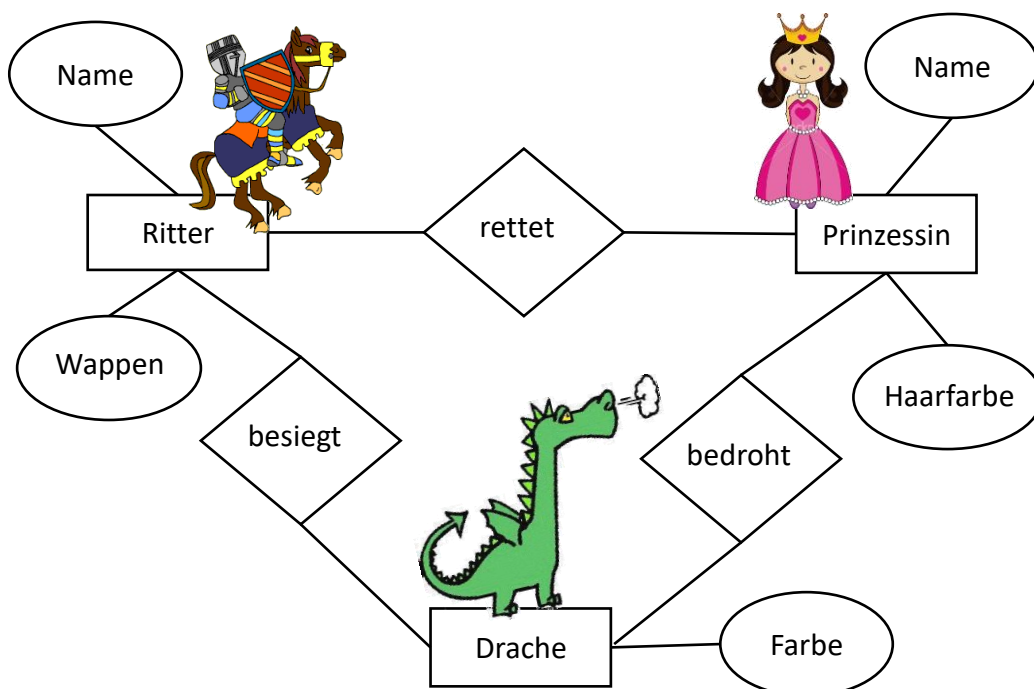
In einem **Aktivitätsdiagramm** werden Abläufe beschrieben. Sie geben eine Reihe von *Aktivitäten* an, die von einem *Anfang (schwarzer Punkt)* zu einem *Ende (schwarzer Punkt mit Kreis darum)* führen. Hier im Beispiel wird die Vorbereitung zum Essen beschrieben. Wie das Beispiel zeigt, kann es auch zu *Verzweigungen* kommen, die als ein auf eine Ecke gestelltes Viereck gezeichnet werden. Eine Verzweigung ist notwendig, wenn es zwei verschiedene Möglichkeiten gibt und jede davon eigene Aktivität erfordert. Sollten alle Zutaten, die für das Rezept benötigt werden, zuhause sein, kann gleich gekocht werden. Sonst müssen die Zutaten erst eingekauft werden.

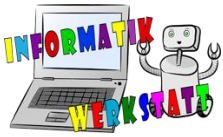


Aufgabe 17: Zeichne ein Aktivitätsdiagramm, das beschreibt, wie du deine Hausaufgaben erledigst. Gib dazu mindestens 5 Aktivitäten an.

Gegenstand-Beziehung-Modell (ER-Diagramme)

Das **Gegenstand-Beziehung-Modell** (in Englisch „**Entity-Relationship-Modell**“) besteht aus **drei Bausteinen** in verschiedenen Formen: **Gegenstandstypen** wie Personen (z.B. eine *Prinzessin*) werden als **Rechteck** gezeichnet und ihre **Eigenschaften** (Attribute), wie z.B. die *Haarfarbe* der Prinzessin, **in Ellipsen**. Die auf die Ecke gestellten **Vierecke** beschreiben die **Beziehung** zwischen den Gegenstandstypen. In unserem Beispiel ist die Beziehung zwischen Ritter und Prinzessin das Wort *rettet*.



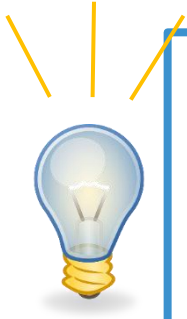


Aufgabe 18: Beschrifte die Bausteine im Ritter-Prinzessin-Drache-Modell mit den jeweiligen Anfangsbuchstaben (**G**egenstandstypen, **B**eziehung, **E**igenschaften)!

Aufgabe 19: Erweitere das Ritter-Prinzessin-Drache-Modell auf der vorhergehenden Seite um einige Figuren, Beziehungen oder Eigenschaften, die du dir selbst ausdenkst.

Aufgabe 20: Denke dir eine eigene Geschichte aus und zeichne dazu ein Gegenstand-Beziehung-Modell.

ALGORITHMEN und PROGRAMME



Unter einem **Algorithmus** wird eine möglichst genaue Beschreibung eines Ablaufes verstanden. Bei den **Aktivitätsdiagrammen** hast du bereits einen **Algorithmus** für die Zubereitung eines Essens kennengelernt und selbst hast du einen **Algorithmus** für das Erledigen deiner Hausaufgaben entworfen. Dabei ist es wichtig, dass alle einzelnen **Schritte** eines **Algorithmus** auch tatsächlich ausführbare Aktivitäten sind.

Aufgabe 21: Was beschreibt der folgende Algorithmus? Schreib rechts die Antwort.

Schritte:

1. Smartphone/Handy nehmen
2. Taste drücken
3. Code/Muster eingeben
4. Kontakt suchen oder Nummer eingeben
5. Auf „Anrufen“ tippen
6. Sprechen

Aufgabe 22: Erstelle ein Aktivitätsdiagramm aus den Schritten in der vorigen Aufgabe.

Algorithmen können in verschiedenen Formen niedergeschrieben werden. Zum Beispiel als ein **Aktivitätsdiagramm**, als **Text**, als **Liste von Schritten** oder auch als **Programm** in einer bestimmten **Programmiersprache**. **Programme** können, wenn sie richtig geschrieben wurden, von Maschinen, also Computern oder Robotern, automatisch ausgeführt werden. Dazu muss das Programm in einer Programmiersprache verfasst werden, die diese Maschine auch kennt.

Aufgabe 23: Öffne die Webseite studio.code.org/hoc/1 und folge den Anweisungen.

Wenn du Aufgabe 6 erfüllt hast, hast du bereits „programmiert“. Dabei sind dir auch schon wichtige Elemente des Programmierens untergekommen, die wir hier besprechen. Dazu werden aus Bausteinen zusammengesetzte Programme, wie du sie kennengelernt hast, mit Programmen der Programmiersprache „JavaScript“ und den dazu passenden Aktivitätsdiagrammen verglichen.

Sequenzen

Eine Sequenz ist eine einfache Abfolge von Befehlen, die der Reihe nach abgearbeitet werden können. Hier siehst du eine Sequenz in drei verschiedenen Formen:

Programm aus Bausteinen	Programm in JavaScript (Programmiersprache)	Programm als Aktivitätsdiagramm
	<pre> moveForward(); moveForward(); turnRight(); moveForward(); </pre>	

Wiederholungen

Hier werden bestimmte **Befehle** wiederholt. Dabei gibt es die Möglichkeit anzugeben, **wie oft** Befehle wiederholt werden sollen, oder **bis zu welchem Ereignis** Befehle wiederholt werden sollen.

Programm aus Bausteinen	Programm in JavaScript	Programm als Aktivitätsdiagramm
	<pre>for (var count = 0; count < 5; count++) { moveForward(); }</pre>	
	<pre>while (notFinished()) { moveForward(); }</pre>	

Verzweigungen

Die Idee hinter einer **Verzweigung** ist, dass bestimmte Befehle nur unter bestimmten **Bedingungen** ausgeführt werden sollen. So kann man zum Beispiel in einem Labyrinth nur dann nach rechts gehen, wenn auch ein Weg nach rechts existiert.

Programm aus Bausteinen	Programm in JavaScript	Programm als Aktivitätsdiagramm
	<pre>if (isPathRight()) { turnRight(); moveForward(); }</pre>	

Aufgabe 24: Löse einige weitere Aufgaben auf studio.code.org.

Aufgabe 25: Öffne die Webseite scratch.mit.edu und probiere ein paar Sachen aus.

ROBOTER

An **Robotern** kannst du die verschiedenen Elemente der **Programmierung**, die du kennengelernt hast, ausprobieren. Dazu stehen verschiedene Roboter zur Verfügung.

Bee-Bot

Mit dem Bee-Bot können einfache **Sequenzen** programmiert werden. Dazu musst du dir nur überlegen, **welche Route** der Bee-Bot fahren soll. Diese gibst du dann mithilfe der **Pfeiltasten** ein. Wenn du den Bee-Bot fertig programmiert hast, drücke auf OK. Dann wird das Programm gestartet und der Bee-Bot fährt los.

Aufgabe 26: Programmiere den Bee-Bot so, dass er ein Quadrat fährt und schreibe die Befehle hier in dein Heft.

Aufgabe 27: Programmiere den Bee-Bot so, dass er diesen Buchstaben nachfährt und schreibe die Befehle hier in dein Heft.



Pro-Bot

Um den Pro-Bot zu programmieren stehen mehrere Möglichkeiten zur Verfügung. Eine **Route** kann gleich wie mit dem Bee-Bot eingegeben werden, mit dem Unterschied, dass du auf dem **Display** das Programm lesen kannst. Zusätzlich kannst du auch eine **Wiederholung** einbauen. Diese beginnst du mit der Taste „**Rpt[**“ und beendest du mit der Taste „**]**“. Alles, was du zwischen diesen beiden Tasten eingibst, wird wiederholt.

Aufgabe 28: Programmiere den Pro-Bot mithilfe der Rpt-Funktion so, dass er ein Quadrat fährt und danach auch zeichnet. Schreibe die Befehle hier in dein Heft.