

Themen

	Seite	
Datenmenge	6	1
Zeichenvorrat, Codewörter	9	2
Barcode (EAN)	11	3
QR-Code	14	4
Paritätsbit	17	5
Prüfziffer	20	6
Vorwärtsfehlerkorrektur	24	7
Fehlerkorrekturverfahren bewerten	27	8
Datenstruktur Liste	32	9
Datenstruktur Baum	34	10
Datenstruktur Graph	36	11

weitere Themen siehe Seite 5

Themen

	Seite	
Sortieren und Filtern in Tabellen	39	12
Mit Formeln und Funktionen arbeiten	42	13
Daten graphisch darstellen	45	14
Sensoren und Aktoren im Smartphone	50	15
Pocket Code App	53	16
Sensoren und Aktoren ansteuern	57	17
Sensordaten auslesen	62	18
Sensordaten auswerten	66	19

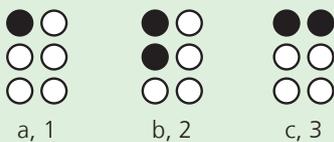
Zeichenvorrat, Codewörter

Codes werden verwendet, um eine Information für einen Anwendungsfall optimal darzustellen.

Dabei muss man zwei Varianten von Codes unterscheiden:

- Alphabete für besondere Anwendungen. Sie benötigen einen großen Zeichenvorrat, um Informationen möglichst detailliert darstellen zu können.
- Codes zur Darstellung umfangreicher Informationen auf kleinen Flächen. Sie lassen aufgrund der begrenzten Zeichenzahl nur eine bestimmte Anzahl unterschiedlicher Codewörter zu.

Die **Brailleschrift** ist ein Beispiel für ein spezielles Alphabet. Sie ermöglicht es Blinden, Texte und sogar ganze Bücher zu lesen. Die Zeichen der Brailleschrift bestanden ursprünglich aus sechs Punkten, die mit ihren beiden Möglichkeiten (flach/erhaben) einen Zeichenvorrat von 2^6 (also 64) Zeichen ergeben.



Dieser Zeichenvorrat erwies sich bald als zu klein. Deshalb wurde in den 1980er Jahren das 8-Punkt-Computerbraille entwickelt. Die 8 Punkte ergeben einen Vorrat von 2^8 (also 256) Zeichen. Neben den 64 Zeichen der ursprünglichen Brailleschrift standen damit weitere 192 Zeichen zur Verfügung, die nun auch Groß- und Kleinschreibung und Zahlen ohne vorangestellte Hilfszeichen ermöglichten.

Der ursprüngliche **ASCII-Code** bestand aus 7 Bit. Er erlaubte die Darstellung von $2^7 = 128$ Zeichen. Zahlreiche Zeichen wie die deutschen ß oder ä, die im Englischen nicht vorkommen, waren mit dem ASCII-Code nicht darstellbar. Der ANSI-Code – eine Erweiterung des ASCII-Codes – besteht aus 8 Bit. Mit ihm lassen sich $2^8 = 256$ Zeichen, also auch zahlreiche Sonderzeichen, darstellen.

Seit 1991 gibt es den auf Basis des ASCII-Codes entwickelten Unicode-Standard, einen 16-Bit-Code, der neben den europäischen Zeichen auch kyrillische, indische, chinesische und japanische Schriftzeichen enthält. Insgesamt können mit dem Unicode $2^{16} = 65\,536$ Zeichen codiert werden.

Das **Kfz-Kennzeichen** ist ein Beispiel für einen Code, mit dessen Hilfe Informationen auf einer begrenzten Fläche dargestellt werden. Links von TÜV-Plakette und Zulassungsplakette ist der Landkreis bzw. die kreisfreie Stadt codiert, in der das Fahrzeug zugelassen ist. Dafür stehen maximal 3 Zeichen zur Verfügung.



Danach folgen ein oder zwei Buchstaben und maximal vier Ziffern. Diese Buchstaben und Ziffern sind entscheidend dafür, wie viele unterschiedliche Kennzeichen (Codewörter) in einem Landkreis bzw. in einer kreisfreien Stadt möglich sind.

Unter der Voraussetzung, dass alle Kombinationen von Buchstaben und Ziffern (außer 0, 00 usw.) vergeben werden, lässt sich maximale Anzahl pro Landkreis ermitteln.

Für Kennzeichen mit einem Buchstaben und zwei Ziffern wie in unserem Beispiel gibt es $26 \text{ Buchstaben} \times 99 \text{ Zahlen} = 2\,574$ Möglichkeiten.

Zeichenvorrat, Codewörter

Aufgabe 1

Der französische Offizier Charles Barbier de la Serre (1767–1841) entwickelte die so genannte Nachtschrift, einen Vorläufer der Brailleschrift. Die Zeichen der Schrift bestanden aus jeweils 12 erhabenen Punkten, die ertastet werden konnten.

Wie viele Zeichen konnten mit dieser Schrift codiert werden?

Der Code hat 12 Punkte mit jeweils 2 Möglichkeiten (flach oder erhaben).

Daraus resultieren $2^{12} = 4096$ Zeichen.

Aufgabe 2

Der Baudot-Code ist ein 5-Bit-Zeichencode, der 1870 von Jean-Maurice-Émile Baudot (1845–1903) für ein von ihm entwickeltes Telegrafengerät erfunden wurde. Jedes Bit kann den Wert 1 oder 0 haben.

a) Wie viele Zeichen konnten mit dem Baudot-Code codiert werden?

b) Reicht der Code für die 26 Buchstaben des Alphabets und die zehn Ziffern aus?

a) Der Code hat 5 Punkte mit jeweils 2 Möglichkeiten (1 oder 0).

Daraus resultieren $2^5 = 32$ Zeichen.

b) Die Zeichenzahl reicht nicht aus, um alle 26 Buchstaben des Alphabets und 10 Ziffern darzustellen.

Aufgabe 3

Im Zulassungsbezirk Berlin waren zum 1. Januar 2019 1 524 484 Fahrzeuge und Fahrzeuganhänger zugelassen.¹⁾

Wie viele weitere Fahrzeuge könnten in Berlin noch zugelassen werden, bevor der Berliner Zulassungsbehörde die Kennzeichen ausgehen?

Kennzeichen mit einem Buchstaben und maximal vier Ziffern erlauben

26 Buchstaben x 9 999 Zahlen
= 259 974 unterschiedliche Kennzeichen.

Kennzeichen mit zwei Buchstaben und maximal vier Ziffern erlauben

26 x 26 Buchstaben x 9 999 Zahlen
= 6 759 324 unterschiedliche Kennzeichen.

Es könnten noch 5 494 814 weitere Fahrzeuge und Fahrzeuganhänger zugelassen werden.

Aufgabe 4

Die kreisfreie Stadt Suhl in Thüringen ist der Zulassungsbezirk in Deutschland mit der kleinsten Anzahl zugelassener Fahrzeuge.

Zum 1. Januar 2019 waren dort 26 922 Kraftfahrzeuge und Fahrzeuganhänger zugelassen.¹⁾

Würde die mögliche Anzahl an Kennzeichen für den Fahrzeugbestand ausreichen, wenn die Zulassungsstelle in Suhl nur Kennzeichen mit einem Buchstaben und maximal 3 Ziffern ausgeben würde?

Kennzeichen mit einem Buchstaben und maximal drei Ziffern erlauben

26 Buchstaben x 999 Zahlen
= 25 974 unterschiedliche Kennzeichen.

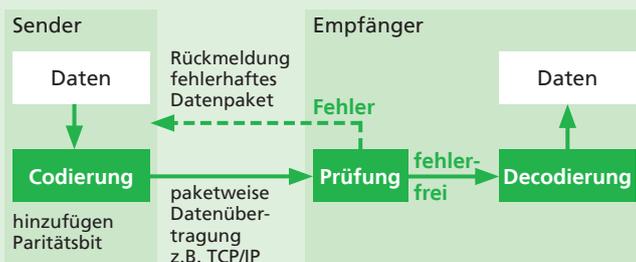
Die Anzahl möglicher Kennzeichen reicht nicht aus. Die Zulassungsstelle in Suhl muss auch Kennzeichen mit zwei Buchstaben oder vier Ziffern ausgeben.

¹⁾ Kraftfahrt-Bundesamt: Bestand am 1. Januar 2019 nach Zulassungsbezirken und Gemeinden
https://www.kba.de/DE/Statistik/Fahrzeuge/Bestand/ZulassungsbezirkeGemeinden/zulassungsbezirke_node.html
(Stand September 2019)

Vorwärtsfehlerkorrektur

Durch technische Probleme oder äußere Einflüsse können bei der Übertragung von Daten Fehler auftreten. Sie können zur Veränderung einzelner Bits oder ganzer Datenpakete führen. Zu einer sicheren Datenübertragung gehören daher auch Maßnahmen zur Fehlererkennung und -korrektur.

Seit den frühen 1970er Jahren beruht die Datenübertragung im Internet auf dem Protokoll TCP/IP. Sender und Empfänger stehen während der Datenübertragung über dieses Protokoll in ständigem Kontakt zueinander. Beim Empfänger werden ankommende Daten auf Fehler überprüft. Dabei werden Paritätsbits genutzt, die beim Sender im Zuge der Codierung zu den Datenpaketen hinzugefügt wurden. Wird ein Fehler festgestellt, erfolgt eine Rückmeldung an den Sender und das betreffende Datenpaket wird erneut übertragen.

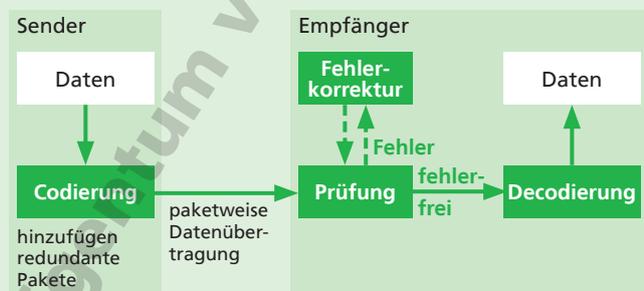


Ablaufschema Rückwärtsfehlerkorrektur

Da die Fehlerkorrektur eine Rückmeldung an den Sender erfordert, nennt man diese Art der Fehlerkorrektur auch Rückwärtsfehlerkorrektur. Sie benötigt kaum zusätzliche Übertragungskapazität, ist durch die mehrfache Übertragung fehlerhafter Pakete aber langsam.

Für Anwendungen, in denen es auf schnelle Datenübertragung ankommt, wie beim digitalen Antennenfernsehen (DVB), beim Mobilfunk oder aber auch beim Abspielen einer gewöhnlichen Audio-CD ist die Rückwärtsfehlerkorrektur daher nicht geeignet. Hier wird stattdessen die so genannte Vorwärtsfehlerkorrektur eingesetzt.

Statt eines Paritätsbits werden den Datenpaketen bei der Vorwärtsfehlerkorrektur beim Codieren zusätzliche, redundante Datenpakete hinzugefügt. Werden beim Empfänger Datenfehler festgestellt, können die fehlerhaften Datenpakete direkt beim Empfänger anhand dieser redundanten Daten wiederhergestellt werden.



Ablaufschema Vorwärtsfehlerkorrektur

Da bei der Vorwärtsfehlerkorrektur die Rückmeldungen an den Sender und die mehrmalige Übertragung einzelner Datenpakete entfallen, ist dieses Verfahren deutlich schneller. Die zusätzlich übertragenen redundanten Datenpakete benötigen jedoch zusätzliche Übertragungskapazität.

Redundanzpaket erzeugen und verlorene Pakete wiederherstellen mit dem XOR-Operator

Für das Erzeugen der Redundanzpakete wird z. B. der XOR-Operator verwendet. XOR ist die Kurzform für „eXclusive OR“ (deutsch exklusiv oder) und zählt zu den logischen Operatoren. Verknüpft man zwei Bits miteinander, ist das Ergebnis eine 1, wenn die Bits unterschiedlich sind.

$$0 \text{ XOR } 0 = 0 \quad 1 \text{ XOR } 0 = 1$$

$$0 \text{ XOR } 1 = 1 \quad 1 \text{ XOR } 1 = 0$$

Auch zwei Datenpakete lassen sich auf diese Weise Bit für Bit mit dem XOR-Operator verknüpfen:

P1	1	0	1	0	1	0	1	0
P2	0	0	1	1	0	0	1	1
P1 XOR P2	1	0	0	1	1	0	0	1

Das wird für das Erzeugen von Redundanzpaketen aus mehreren verknüpften Datenpaketen genutzt.

Die drei Datenpakete P1, P2 und P3 werden beispielsweise um ein Redundanzpaket PR ergänzt, indem sie über die Formel $PR = (P1 \text{ XOR } P2) \text{ XOR } P3$ verknüpft werden.

P1	1	0	1	0	1	0	1	0
P2	0	0	1	1	0	0	1	1
P3	0	0	0	0	1	1	1	1
P1 XOR P2	1	0	0	1	1	0	0	1
$PR = (P1 \text{ XOR } P2) \text{ XOR } P3$	1	0	0	1	0	1	1	0

Kommt eines der drei Datenpakete beim Empfänger nicht an und ist bekannt, welches der drei Pakete fehlt, lässt es sich aus den übrigen drei Paketen wiederherstellen.

$$P1 = (PR \text{ XOR } P3) \text{ XOR } P2 \quad P2 = (PR \text{ XOR } P1) \text{ XOR } P3$$

$$P3 = (PR \text{ XOR } P2) \text{ XOR } P1$$

Vorwärtsfehlerkorrektur

Aufgabe 1

Beschreibe den Ablauf der Vorwärtsfehlerkorrektur.

Beim Codieren werden den Datenpaketen zusätzliche, redundante Datenpakete hinzugefügt.

Beim Empfänger werden die Daten auf Fehler geprüft. Werden Datenfehler festgestellt, werden die fehlerhaften Datenpakete anhand der redundanten Daten wiederhergestellt.

Aufgabe 2

Vergleiche die Rückwärtsfehlerkorrektur und die Vorwärtsfehlerkorrektur in Bezug auf die benötigte Übertragungskapazität und die Übertragungsgeschwindigkeit.

	Rückwärtsfehlerkorrektur	Vorwärtsfehlerkorrektur
Übertragungskapazität	kaum zusätzliche Übertragungskapazität nötig	zusätzliche Kapazität für die Übertragung der Redundanzpakete nötig
Übertragungsgeschwindigkeit	gering aufgrund mehrmaliger Übertragung fehlerhafter Pakete	hoch, da alle Pakete nur einmal übertragen werden müssen

Aufgabe 3

Warum ist die Datenübertragung mittels Vorwärtsfehlerkorrektur schneller als mittels Rückwärtsfehlerkorrektur?

Rückmeldungen an den Sender und mehrmalige Übertragung fehlerhafter Datenpakete führen bei der Rückwärtsfehlerkorrektur zu einer geringen Übertragungsgeschwindigkeit.

Durch die Wiederherstellung fehlerhafter Datenpakete direkt beim Empfänger sind keine Rückmeldungen an den Sender nötig und alle Datenpakete müssen nur einmal übertragen werden. Dadurch ist die Übertragungsgeschwindigkeit höher als bei der Rückwärtsfehlerkorrektur.

Vorwärtsfehlerkorrektur

Aufgaben 4

Wie lautet das Redundanzpaket PR, das aus den folgenden Datenpaketen mit Hilfe des XOR-Operators ermittelt wird?

P1	1	0	0	1	1	0	0	1
P2	1	1	1	0	1	1	1	0
P3	0	1	0	0	1	1	0	0

P1 XOR P2	0	1	1	1	0	1	1	1
PR = (P1 XOR P2) XOR P3	0	0	1	1	1	0	1	1

Aufgabe 5

Bei der Übertragung ging das Datenpaket P3 verloren. Stelle es mit Hilfe des Redundanzpakets PR und des XOR-Operators wieder her.

P1	1	0	0	1	0	0	0	1
P2	1	1	1	0	1	0	1	1
P3								
PR	0	0	0	1	0	0	0	1

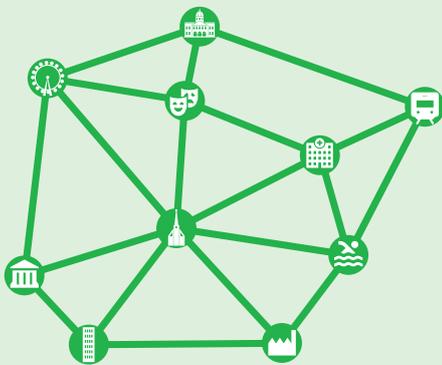
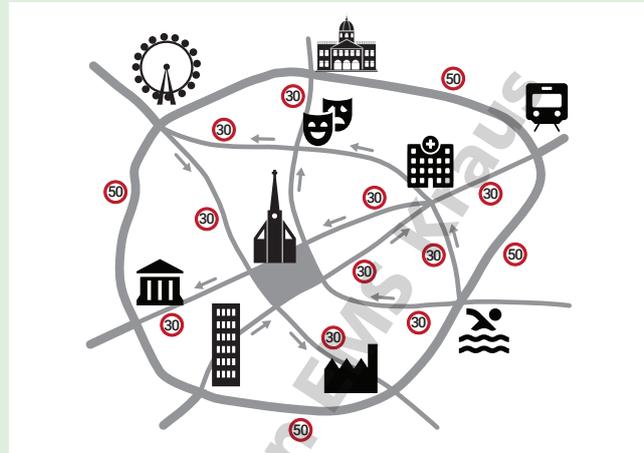
PR XOR P2	1	1	1	1	1	0	1	0
P3 = (PR XOR P2) XOR P1	0	1	1	0	1	0	1	1

Datenstruktur Graph

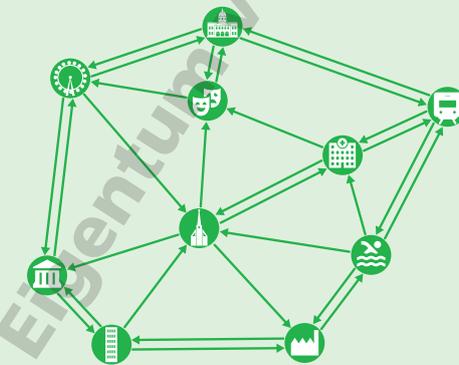
Der Graph ist eine dynamische Datenstruktur, mit der sich vernetzte Strukturen wie Straßenverbindungen, Rohrleitungs- und Telefonnetze oder auch soziale Netzwerke abbilden lassen.

Graphen bestehen aus Knoten und Kanten, die jeweils zwei Knoten miteinander verbinden. Innerhalb eines Graphen führt von jedem Knoten ein Weg zu jedem anderen Knoten.

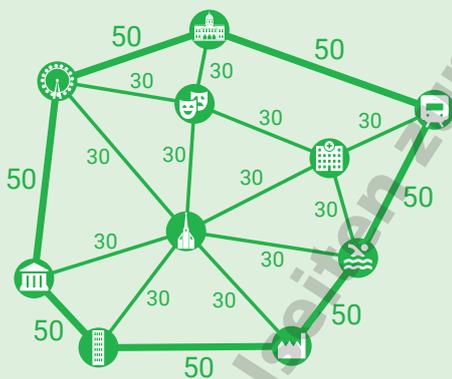
Man unterscheidet vier grundlegende Arten von Graphen:



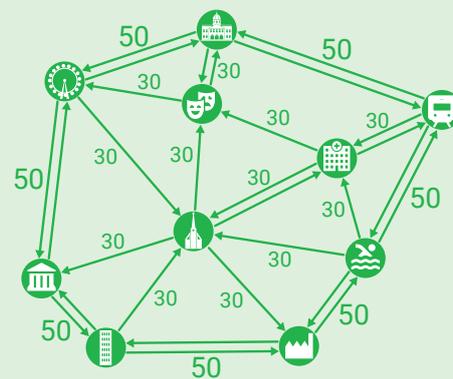
Ungerichtete Graphen zeigen nur die bestehenden Verbindungen der Knoten. In unserem Beispiel sind das die Straßen, die von einer Sehenswürdigkeit unserer kleinen Stadt zur anderen führen.



Der gerichtete Graph enthält zusätzlich Informationen über die Richtung der Verbindungen. In unserem Beispiel gibt es Einbahnstraßen und Straßen mit Gegenverkehr, die im Graph als einzelne und doppelte Pfeile dargestellt sind.



In einem ungerichteten, gewichteten Graph lassen sich Eigenschaften der Wegstrecken abbilden, die durch eine Kante repräsentiert werden. Das können beispielsweise Entfernungen in einem Straßennetz oder Durchflussmengen in einem Rohrleitungsnetz sein. In unserem Beispiel lassen sich so die unterschiedlichen zulässigen Höchstgeschwindigkeiten darstellen.

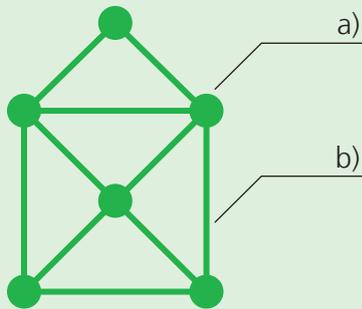


Ein gerichteter, gewichteter Graph vereint beide Möglichkeiten. In ihm lassen sich sowohl die Richtung der Verbindungen als auch Eigenschaften der Wegstrecken (Kanten) abbilden. In unserem Beispiel lassen sich auf diese Weise die Einbahnstraßen und die zulässigen Höchstgeschwindigkeiten abbilden.

Datenstruktur Graph

Aufgabe 1

Benenne die Teile des Graphen.



- a) Knoten
b) Kante

Aufgabe 2

Worin besteht der Unterschied zwischen einem Baum und einem Graph?

Innerhalb eines Graphen führt von jedem Knoten ein Weg zu jedem anderen Knoten. Es gibt keine hierarchischen Beziehungen zwischen den Knoten eines Graphen. Alle Knoten sind gleichwertig.

In einem Baum besteht eine hierarchische Beziehung vom Wurzelknoten bis hinunter zu einem Blatt. Innerhalb eines Baumes gibt es von der Wurzel zu jedem Knoten einen eindeutigen Pfad.

Aufgabe 3

Nenne Beispiele aus dem Alltag, die sich mit Hilfe von Graphen darstellen lassen, und zwar als

- a) ungerichteter Graph
Autobahnnetz,
Stromnetz,
Soziales Netzwerk
- b) gerichteter Graph
Abwassernetz mit Fließrichtung,
Straßennetz mit Einbahnstraßen
Fluchtwegplan in einem Gebäude
- c) gewichteter Graph
Autobahnnetz mit Entfernungen,
Rohrleitungsnetz mit unterschiedlichen Rohrquerschnitten
Busliniennetz mit Fahrzeiten

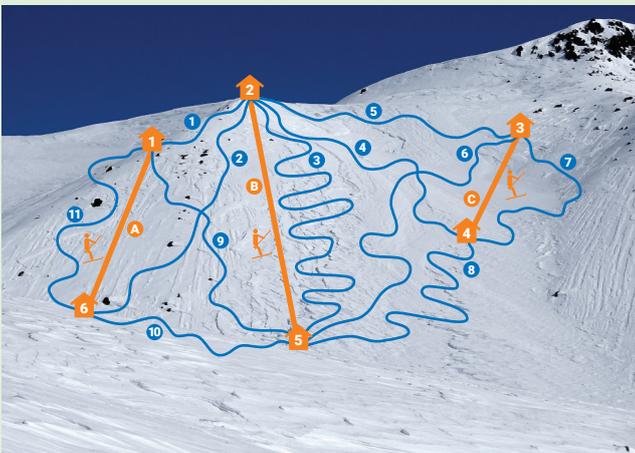
Datenstruktur Graph

Aufgabe 4

Zeichne unser kleines Skigebiet im Bild als Graph, und zwar als

- a) ungerichteter Graph
- b) gerichteter Graph
- c) gewichteter Graph

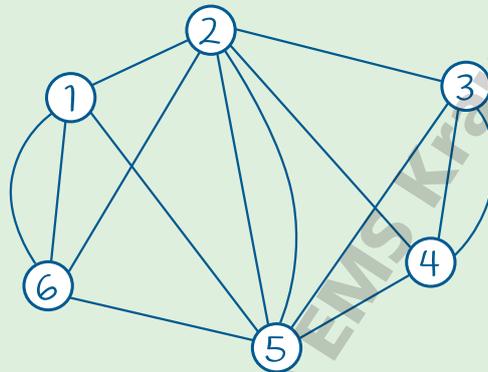
Dabei gehen wir davon aus, dass mit den Schleppliften nur bergauf und auf den Pisten nur bergab gefahren wird.



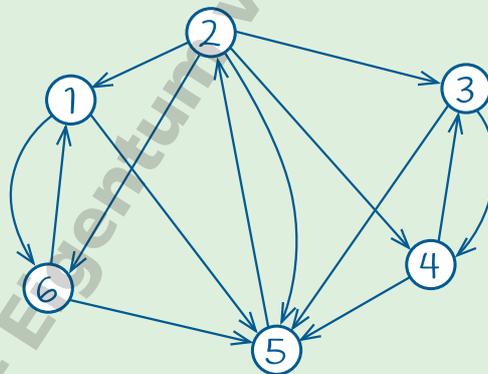
1 Lichtmoosabfahrt 800 m	8 Familienabfahrt 2500 m
2 Kitzsteinabfahrt 3300 m	9 Jägerabfahrt 2800 m
3 Sonneckabfahrt 3800 m	10 Grafenwiese 1400 m
4 Zirbentalabfahrt 3100 m	11 Schafalpeabfahrt 2700 m
5 Fuchsbergabfahrt 2600 m	A Schafalpelift 1300 m
6 Angertalabfahrt 3500 m	B Kitzsteinlift 2700 m
7 Breitspitzabfahrt 1500 m	C Breitspitzlift 800 m

Foto: Natalia Kollegova (Pixabay)

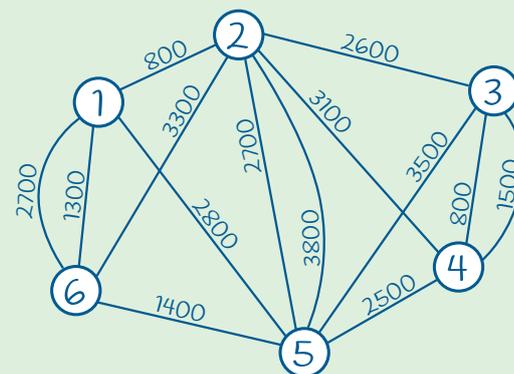
a)



b)



c)



Daten graphisch darstellen

Daten lassen sich in Tabellen kompakt und übersichtlich zusammenfassen. Doch umfangreiche Tabellen sind mühsam zu lesen und häufig ist es schwer, Zusammenhänge zwischen den Werten zu erkennen.

Diagramme helfen, die Tabellendaten anschaulich darzustellen, so dass Verläufe, Trends und Größenvergleiche leichter zu erfassen sind.

Tabellenkalkulations-Software unterstützt die Darstellung der Tabellendaten in Diagrammform durch vordefinierte Diagrammtypen.

Im Menü „Einfügen“ gelangt man – je nach Software – über eines dieser Symbole zu einem Menü, das bei der Auswahl eines geeigneten Diagrammtyps hilft.



Jahr	Kegelrobben ¹⁾
2015	213
2016	301
2017	422
2018	383
2019	451
2020	587
2021	913
2022	1086

Für die Darstellung der Entwicklung des Kegelrobben-Bestandes im Wattenmeer Niedersachsens und Hamburgs bietet sich beispielsweise ein XY-Diagramm an, das auch Streu- oder Punktdiagramm genannt wird.

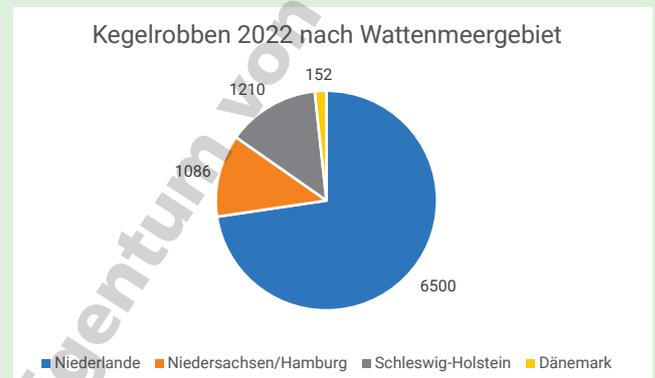
Im Diagramm-Assistenten oder über dieses Symbol lassen sich weitere Diagrammelemente wie eine Legende, Achsenbeschriftungen oder eine Überschrift zum Diagramm hinzufügen.



Das Vergleichen unterschiedlicher Mengen gelingt am anschaulichsten mit Kreisdiagrammen.

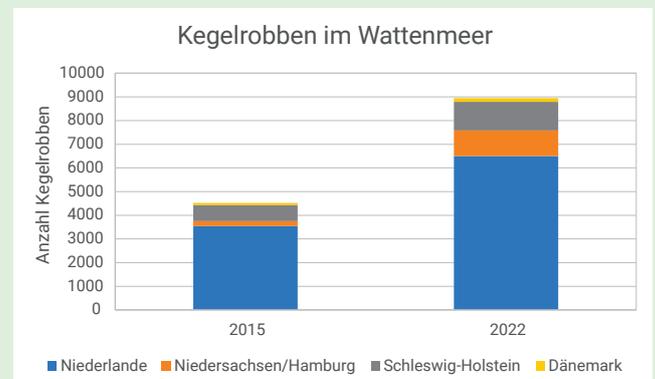
Wattenmeergebiet	Kegelrobben ¹⁾ 2015	2022
Niederlande	3544	6500
Niedersachsen/Hamburg	213	1086
Schleswig-Holstein	676	1210
Dänemark	88	152

Dieses Kreisdiagramm zeigt beispielsweise auf Anhieb, dass fast drei Viertel der Kegelrobben im niederländischen Teil des Wattenmeeres leben.



In Kreisdiagrammen sind die Daten immer als Teile eines Ganzen, also in prozentualer Verteilung dargestellt. Es eignet sich daher nicht, um die in unterschiedlichen Jahren im Wattenmeer gezählten Kegelrobben miteinander zu vergleichen.

So genannte gestapelte Säulendiagramme zeigen sowohl die Mengenverteilung innerhalb jeder Säule als auch den Vergleich der Gesamtmengen anhand der Säulenhöhe. Sie bieten sich daher an, die 2015 und 2022 im Wattenmeer gezählten Kegelrobben miteinander zu vergleichen.



¹⁾ Monitoring-Ergebnisse der Kegelrobbenzählungen im Wattenmeer in Niedersachsen und Hamburg <https://www.waddensea-secretariat.org/de/seehunde> (Stand April 2023)

Daten graphisch darstellen

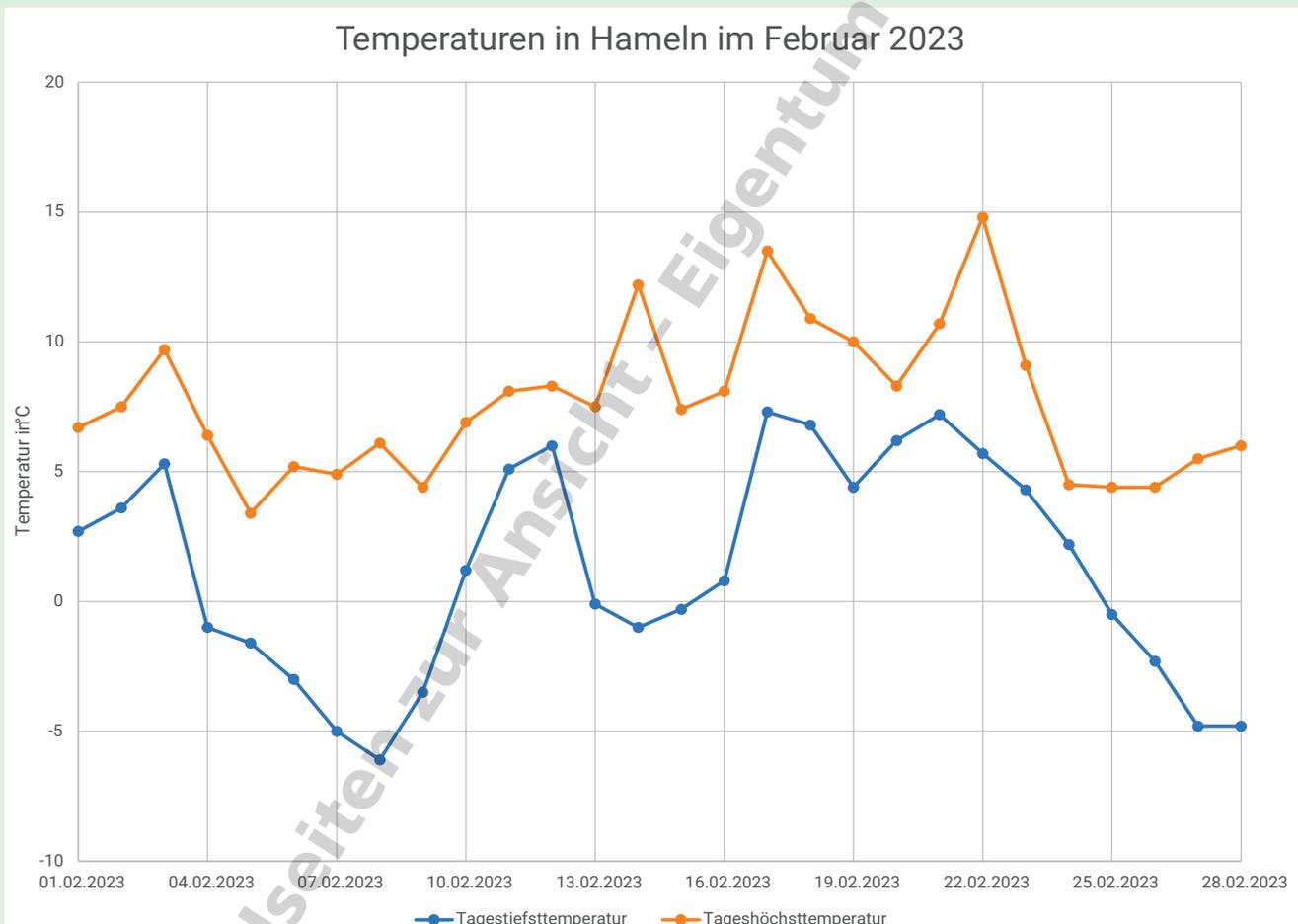
Aufgabe 1

Öffne die Datei Temperatur_Hameln_02-2023.xlsx¹⁾.

- Erstelle ein XY-Diagramm, das die Tagestiefst- und die Tageshöchsttemperaturen für Hameln im Februar 2023 darstellt.
- Füge eine Überschrift und einen Achsentitel an der Y-Achse ein.

Zusatzaufgaben

- Formatiere die Y-Achse so, dass die horizontale Achse sie bei -10 °C schneidet.
- Formatiere die X-Achse so, dass sie mit dem 1. Februar beginnt und mit dem 28. Februar endet.
Beachte dabei, dass Microsoft Excel anstelle des Datums den so genannten DATWERT anzeigt. Das ist eine Funktion, die vom 1. Januar 1900 (DATWERT = 1) ausgehend fortlaufend die Tage zählt. Der 1. Februar 2023 hat den DATWERT 44958 und der 28. Februar den DATWERT 44985.



Datenquellen:

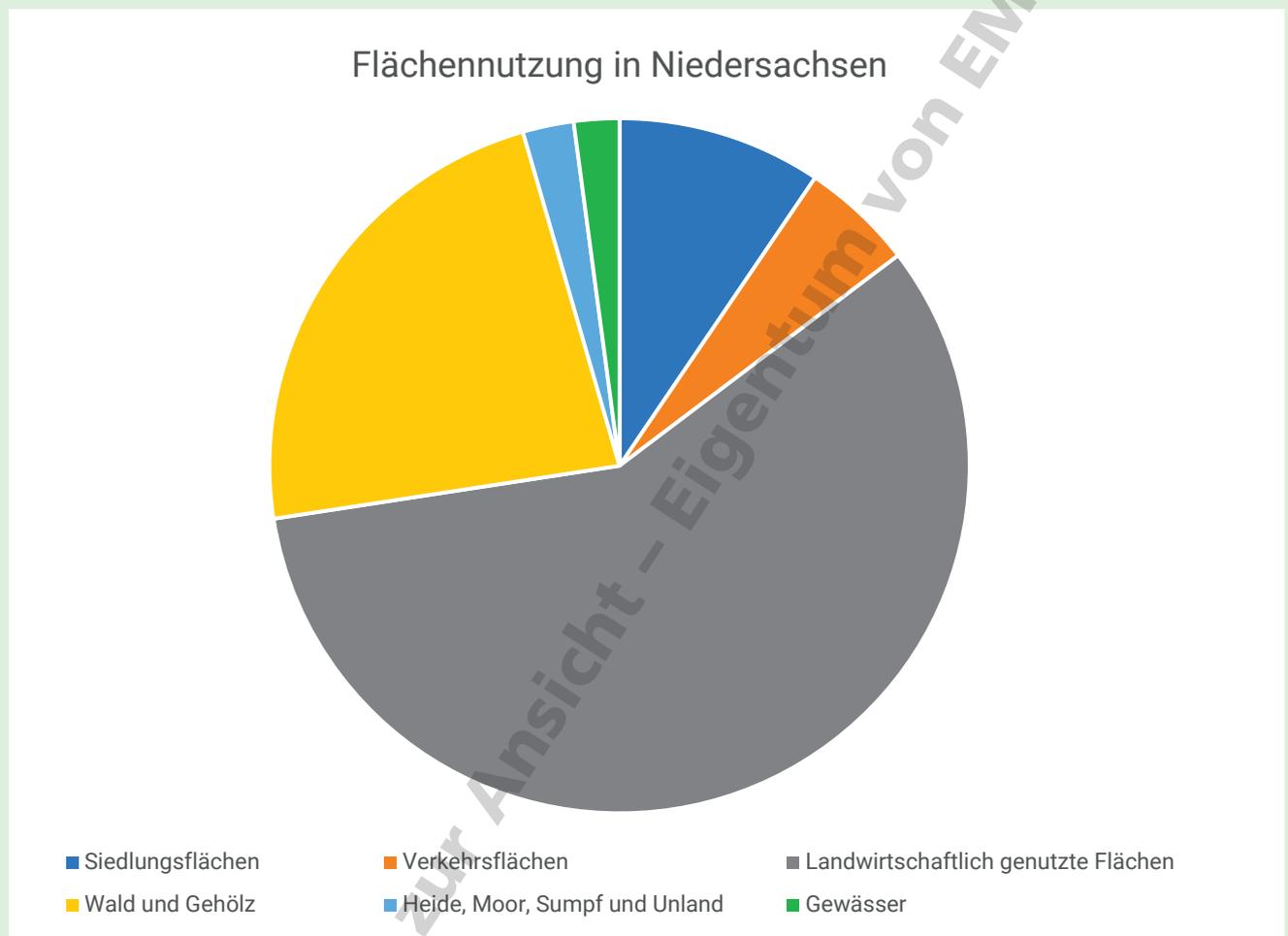
- <https://meteostat.net/de/place/de/hameln?s=D3675&t=2023-02-01/2023-02-28> (Stand April 2023)
- [https://www1.nls.niedersachsen.de/statistik/html/default.asp; 12411 - Fortschreibung des Bevölkerungsstandes; Bevölkerung nach Altersgruppen \(23\) und Geschlecht \(Gemeinde\) \(Stand Februar 2023\)](https://www1.nls.niedersachsen.de/statistik/html/default.asp; 12411 - Fortschreibung des Bevölkerungsstandes; Bevölkerung nach Altersgruppen (23) und Geschlecht (Gemeinde) (Stand Februar 2023))
- Flächenerhebung nach Art der tatsächlichen Nutzung, <https://www.statistik.niedersachsen.de/flaechenerhebung/flaechenerhebung-nach-art-der-tatsaechlichen-nutzung-statistische-berichte-87671.html> (Stand April 2023)

Daten graphisch darstellen

Aufgabe 2

Öffne die Datei Flaechennutzung_NI.xlsx³⁾, die Daten zur Nutzung von Flächen in Niedersachsen enthält.

- Erstelle ein Kreisdiagramm für die Flächennutzung in Niedersachsen.
- Füge eine Überschrift und eine Legende hinzu.



Datenquellen:

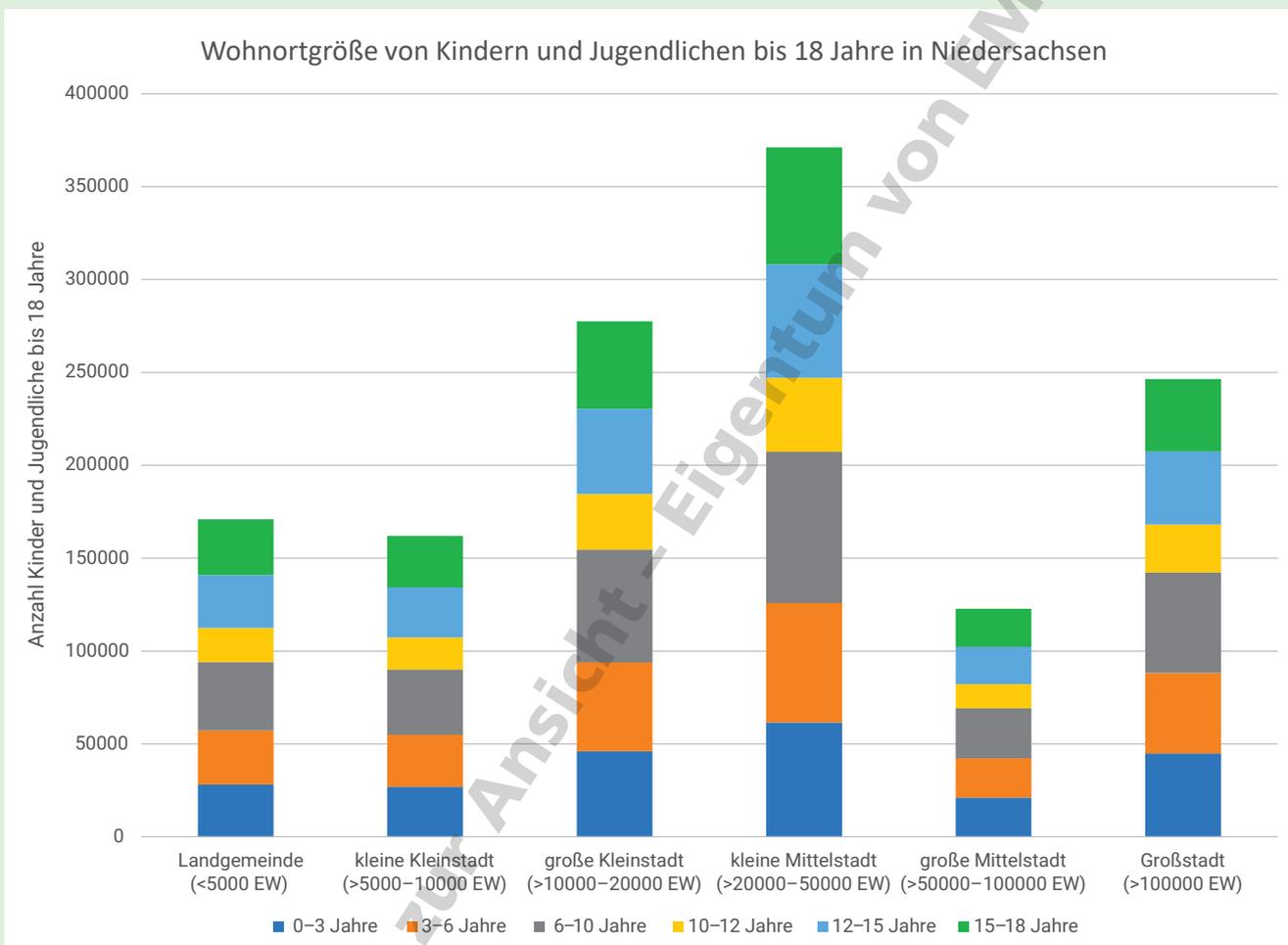
- <https://meteostat.net/de/place/de/hameln?s=D3675&t=2023-02-01/2023-02-28> (Stand April 2023)
- [https://www1.nls.niedersachsen.de/statistik/html/default.asp; 12411 - Fortschreibung des Bevölkerungsstandes; Bevölkerung nach Altersgruppen \(23\) und Geschlecht \(Gemeinde\) \(Stand Februar 2023\)](https://www1.nls.niedersachsen.de/statistik/html/default.asp; 12411 - Fortschreibung des Bevölkerungsstandes; Bevölkerung nach Altersgruppen (23) und Geschlecht (Gemeinde) (Stand Februar 2023))
- Flächenerhebung nach Art der tatsächlichen Nutzung, <https://www.statistik.niedersachsen.de/flaechenerhebung/flaechenerhebung-nach-art-der-tatsaechlichen-nutzung-statistische-berichte-87671.html> (Stand April 2023)

Daten graphisch darstellen

Aufgabe 3

Öffne die Datei Wohnortgroesse.xlsx²⁾, die Daten zur Größe der Wohnorte von Kindern und Jugendlichen bis 18 Jahren in Niedersachsen enthält.

- Erstelle ein gestapeltes Balkendiagramm, in dem für jede Wohnortgröße die Anzahl der dort lebenden Jugendlichen in den Altersgruppen aufgetragen ist.
- Füge eine Überschrift und einen Achsentitel an der vertikalen Achse ein.



Datenquellen:

¹⁾ <https://metostat.net/de/place/de/hameln?s=D3675&t=2023-02-01/2023-02-28> (Stand April 2023)

²⁾ [https://www1.nls.niedersachsen.de/statistik/html/default.asp; 12411 - Fortschreibung des Bevölkerungsstandes; Bevölkerung nach Altersgruppen \(23\) und Geschlecht \(Gemeinde\) \(Stand Februar 2023\)](https://www1.nls.niedersachsen.de/statistik/html/default.asp; 12411 - Fortschreibung des Bevölkerungsstandes; Bevölkerung nach Altersgruppen (23) und Geschlecht (Gemeinde) (Stand Februar 2023))

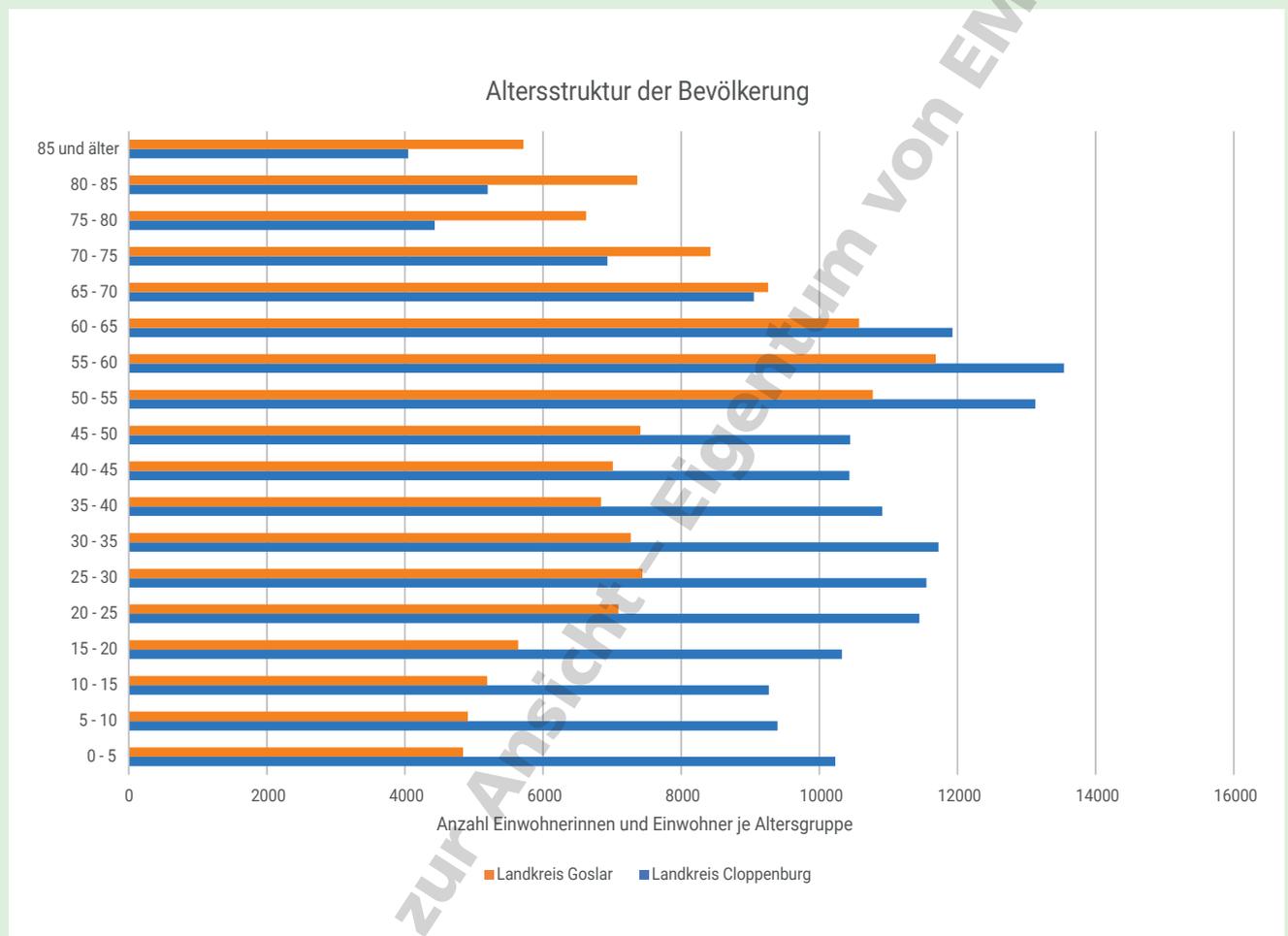
³⁾ Flächenerhebung nach Art der tatsächlichen Nutzung, <https://www.statistik.niedersachsen.de/flaechenerhebung/flaechenerhebung-nach-art-der-tatsaechlichen-nutzung-statistische-berichte-87671.html> (Stand April 2023)

Daten graphisch darstellen

Aufgabe 4

Öffne die Datei Altersstruktur_CLP_GS.xlsx²⁾, die Daten zur Altersstruktur der Bevölkerung der Landkreise Cloppenburg und Goslar enthält.

- Erstelle ein Balkendiagramm, das die Altersstruktur beider Landkreise enthält.
- Füge eine Überschrift und einen Achsentitel an der horizontalen Achse ein.



Datenquellen:

- <https://meteostat.net/de/place/de/hameln?s=D3675&t=2023-02-01/2023-02-28> (Stand April 2023)
- [https://www1.nls.niedersachsen.de/statistik/html/default.asp; 12411 - Fortschreibung des Bevölkerungsstandes; Bevölkerung nach Altersgruppen \(23\) und Geschlecht \(Gemeinde\) \(Stand Februar 2023\)](https://www1.nls.niedersachsen.de/statistik/html/default.asp; 12411 - Fortschreibung des Bevölkerungsstandes; Bevölkerung nach Altersgruppen (23) und Geschlecht (Gemeinde) (Stand Februar 2023)
- Flächenerhebung nach Art der tatsächlichen Nutzung, <https://www.statistik.niedersachsen.de/flaechenerhebung/flaechenerhebung-nach-art-der-tatsaechlichen-nutzung-statistische-berichte-87671.html> (Stand April 2023)

Sensoren und Aktoren ansteuern

Positionen und Bewegungen

Die gesamte Programmierung in Pocket Code erfolgt durch Bausteine. Sobald eine Figur in ein Projekt eingefügt wird, fügt die App selbst die ersten Bausteine eines Skripts ein, mit dem die Figur auf dem Bildschirm platziert wird.

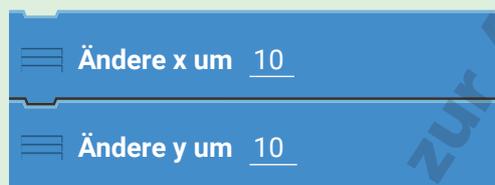


Die voreingestellten Werte 100 und 200 der beiden Parameter des zweiten Bausteins sind dabei schon an die Position der Figur angepasst.



Im weiteren Verlauf eines Skripts kann man mit diesen Bausteinen eine Figur anweisen, sich zu bestimmten x- oder y-Koordinaten zu bewegen.

Dabei können absolute Positionen definiert werden oder relative Koordinaten mit Bezug zur aktuellen Position, indem im Formeleditor aus den Eigenschaften die aktuelle „Position x“ oder die aktuelle „Position y“ eingesetzt werden.



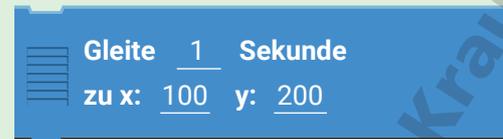
Mit diesen Bausteinen lassen sich entsprechend Figuren in x- und y-Richtung bewegen.

Geschwindigkeit

Die Geschwindigkeit, mit der sich eine Figur bewegt, wird mit diesem Baustein vorgegeben.



Alternativ kann man mit diesem Baustein vorgeben, welche Strecke eine Figur in einer bestimmten Zeit zurücklegt.



Soll sich eine Figur drehen, kann die Rotationsgeschwindigkeit mit diesen Bausteinen eingestellt werden.



Für alle diese Bausteine gilt, dass die voreingestellten Parameter im Formel-Editor durch andere Zahlenwerte, Variablen, Funktionen oder die Messwerte von Sensoren ersetzt werden können. Dabei können unterschiedliche Größen auch durch logische Operatoren miteinander verknüpft werden.

Wiederholungen und Abfragen

Häufig sollen bestimmte Anweisungen nicht nur einmal ausgeführt, sondern wiederholt werden.



Mit diesem Baustein wird definiert, wie oft die darunter platzierten Anweisungen wiederholt werden sollen.



Mit diesem Baustein erreicht man, dass die Anweisungen fortlaufend wiederholt werden, solange das Programm aktiv ist.

Den Block „Ende der Schleife“ fügt die Pocket Code App bei allen Wiederholungen automatisch ein. Er ist wichtig, da nur die Bausteine zwischen Start und Ende einer Wiederholungsschleife wiederholt werden.

Sensoren und Aktoren ansteuern

Aufgabe 1

Hintergrund



holz.jpg

Figur

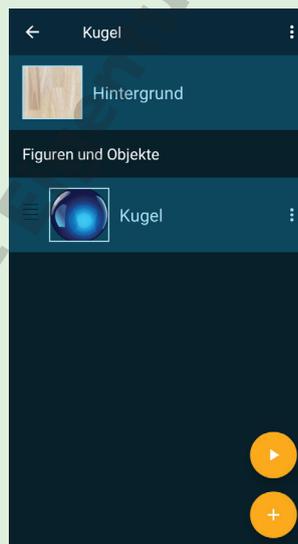


kugel.png

Story

Die Kugel erscheint bei Programmstart in der Mitte. Wenn das Smartphone geneigt wird, soll sich die Kugel wie beim Balancieren auf einem Brett in die jeweilige Richtung bewegen. Je stärker die Neigung, desto schneller soll sich die Kugel bewegen. Sie soll dabei am Rand abprallen.

- Erstelle ein neues Projekt „Kugel“ mit vertikaler Ausrichtung. Lade als Hintergrundbild die Datei holz.jpg. Füge eine Figur namens „Kugel“ hinzu und wähle das Bild kugel.png aus. Platziere die Figur visuell in der Mitte des Bildschirms.
- Öffne das Menü der Figur und korrigiere die Anfangsposition auf $x = 0 / y = 0$.

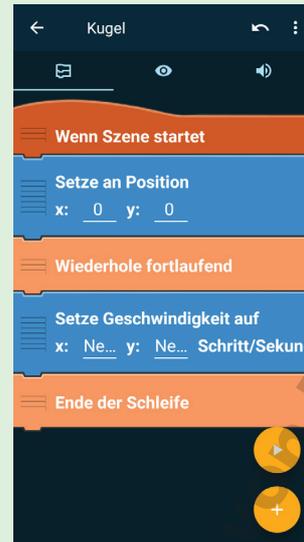


- Füge aus der Kategorie „Steuerung“ den Baustein „Wiederhole fortlaufend“ ein. Schiebe ihn unter den Baustein, der die Anfangsposition setzt.



Sensoren und Aktoren ansteuern

- d) Füge aus der Kategorie „Bewegung“ den Baustein „Setze Geschwindigkeit auf“ ein. Schiebe ihn zwischen die Blöcke „Wiederhole fortlaufend“ und „Ende der Schleife“. Ändere die Parameter für x und y im Formel-Editor. Wähle dort die Rubrik „Sensoren“ und darin „Neigung x“ für den x-Parameter bzw. „Neigung y“ für den y-Parameter.



- e) Füge aus der Kategorie „Bewegung“ zwei weitere Bausteine ein: „Ändere x um“ und „Ändere y um“ und schiebe sie unter den Geschwindigkeits-Baustein.

Ändere die Parameter für x und y im Formel-Editor. Wähle dazu aus der Rubrik „Sensoren“ wieder „Neigung x“ und „Neigung y“, setze in den beiden Formel aber jeweils ein Minus (-) davor.



- f) Füge als letzten Baustein innerhalb der Schleife aus der Kategorie „Bewegung“ den Baustein „Pralle vom Rand ab“ ein.



Sensoren und Aktoren ansteuern

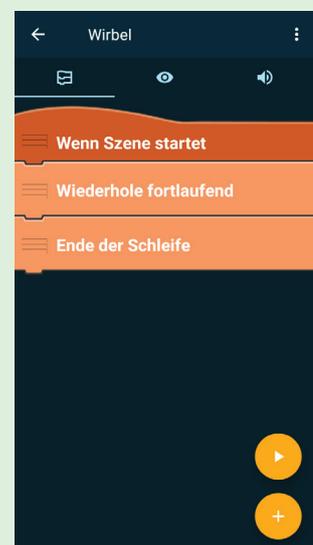
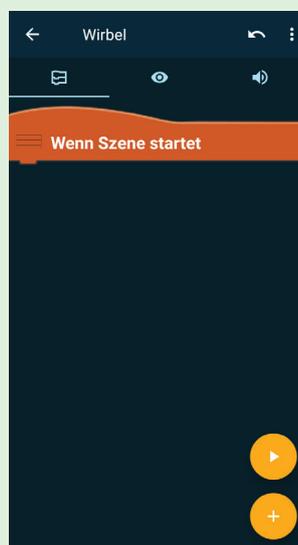
Aufgabe 2

Hintergrund	Figur
	 wirbel.png
<p>Story</p> <p>Der Wirbel erscheint bei Programmstart in der Mitte. Je nach verwendetem Sensorsignal dreht er sich umso schneller rechts herum, je</p> <ul style="list-style-type: none"> ▪ lauter es in der Umgebung ist. ▪ weiter oben oder unten der Bildschirm berührt wird. 	
schwarz.jpg	

- Erstelle ein neues Projekt „Lärm“ mit vertikaler Ausrichtung. Lade als Hintergrundbild die Datei schwarz.jpg.
- Füge eine Figur „Wirbel“ hinzu und wähle das Bild wirbel.png aus. Lass die Figur automatisch in der Mitte des Bildschirms platziert.

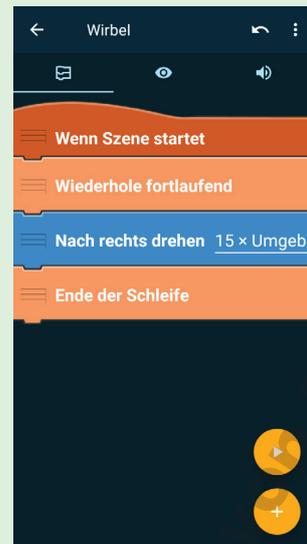


- Füge aus der Kategorie „Ereignisse“ den Baustein „Wenn Szene startet“ ein.
- Füge aus der Kategorie „Steuerung“ den Baustein „Wiederhole fortlaufend“ ein.



Sensoren und Aktoren ansteuern

- e) Füge aus der Kategorie „Bewegung“ den Baustein „Nach rechts drehen ... Grad/Sekunde“ ein. Setze den Parameter auf „15 x Umgebungslautstärke“.



Alternative zur Umgebungslautstärke:

- Setze den Parameter auf „Absoluter Wert(Bildschirm berührt y)“.

