

1. Diagrammarten

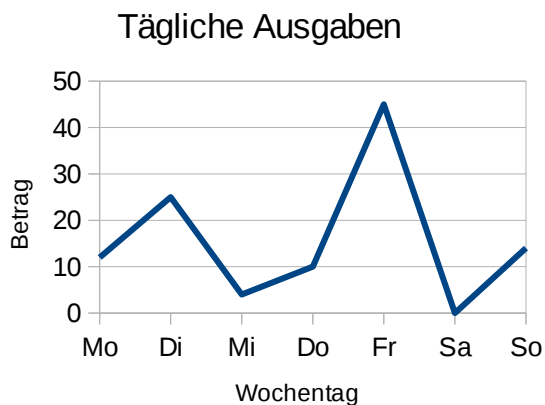
Gegeben sind die folgenden Diagramme.

4 Punkte

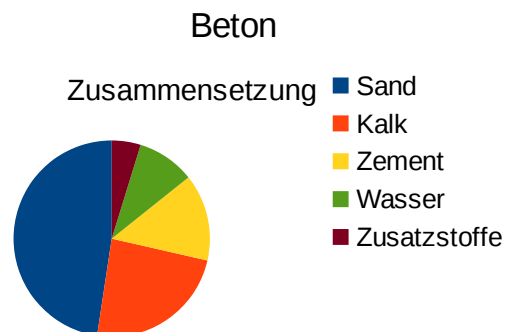
Aufgabe:

Entscheide, bei welchen Darstellungen der gewählte Diagrammtyp ungeeignet ist. Welcher Typ ist besser geeignet? Begründe.

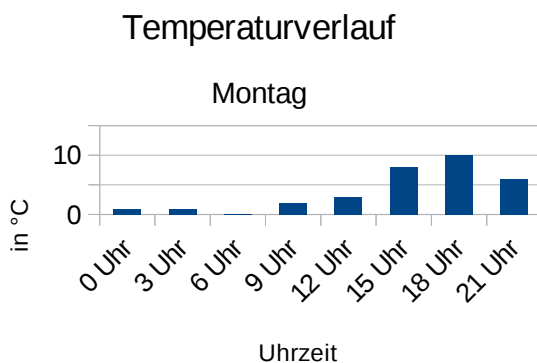
a)



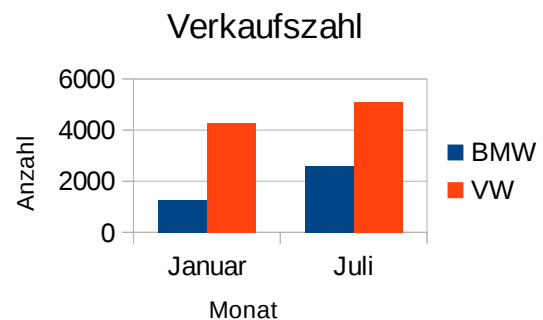
b)



c)



d)



Vorname: _____ Name: _____ Klasse: _____

2. Zahlensysteme

Das Oktalsystem (von lateinisch octo „acht“) ist ein Stellenwertsystem mit der Basis 8 (daher auch Achtersystem genannt). Es kennt acht Ziffern zur Darstellung einer Zahl: 0, 1, 2, 3, 4, 5, 6 und 7.

Seine Ursprünge finden sich im Schweden des 17. Jahrhunderts. Als Urheber kommen König Karl XII., der Wissenschaftler Emanuel Swedenborg oder der Erfinder Christopher Polhem in Frage.

Dezimal	1	2	3	4	5	6	7	8	9
Oktal	1	2	3	4	5	6	7	10	11

Für die Dezimalzahl 8 wird im Oktalsystem eine zweite Stelle benötigt, da es die Ziffer 8 dort nicht gibt und die Zahl wird als 10 dargestellt.

Aufgaben:

- a) Stelle die folgenden Zahlen im Oktalsystem dar. 3 Punkte

$$17 = \underline{\hspace{2cm}} \qquad 63 = \underline{\hspace{2cm}}$$

$$139 = \underline{\hspace{2cm}}$$

- b) Für welche Dezimalzahlen stehen die folgenden Oktalzahlen? 3 Punkte

$$20 = \underline{\hspace{2cm}} \qquad 56 = \underline{\hspace{2cm}}$$

$$261 = \underline{\hspace{2cm}}$$

3. Datenkomprimierung

Das bunte Bild wurde durch ein Programm codiert. Der Code ist rechts daneben zu sehen, er besteht aus Buchstabenfolgen. Die Codierung erfolgt zeilenweise. Gleichfarbige Felder werden zusammengefasst.

Aufgaben:

a) Wie lautet der Code in der 3. Zeile?

1 Punkt

X	X	O	O	O	X	X	bxcobx
X	O	O	O	O	O	X	axeoax
O	O	i	i	i	i	O	
X	O	X	i	X	O	X	axaoaxiaxaoax
X	X	i	O	O	X	X	bxaibobx

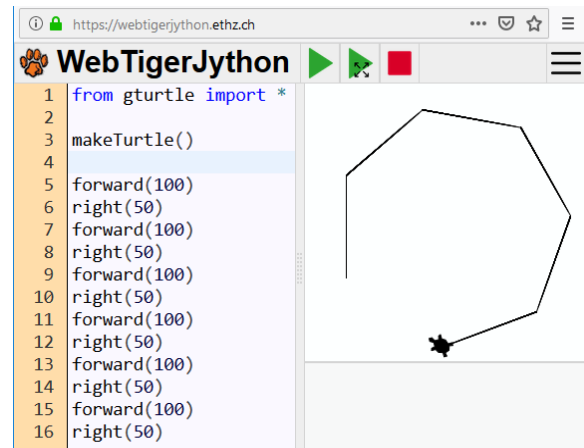
b) Wie sieht das Bild aus, dessen Code rechts gegeben ist? Verwende auch 3 verschiedene Farben.

5 Punkte

							axbiaobiax
							bxcibx
							aibxaibxai
							coaxco
							biaoaxaobi

4. Algorithmierung

Computerprogramme erfordern zu deren Erstellung Kenntnisse einer Programmiersprache. Python ist eine davon und zudem leicht zu erlernen. Du kannst damit dein System erweitern, sogar in Spielen wie minecraft findet python Verwendung. Dabei ist wie bei allen Programmiersprachen die genaue Schreibweise sehr wichtig. Hier sollte die Schildkröte ein 6-Eck zeichnen. Leider ging etwas schief.



Aufgaben:

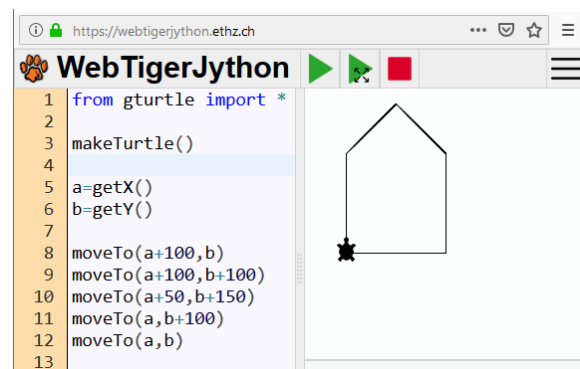
a) Was muss man ändern? Begründe.

2 Punkte

b) Mit den Methoden `forward()` und `right()` oder `left()` lässt sich das Haus vom Nikolaus schlecht zeichnen. Welches Problem entsteht dabei?

1 Punkt

c) Mit dem Beispiel rechts geht es einfacher: Die Schildkröte startet an den X-Y Koordinaten $(0,0)$. Mit dem Befehl `get` wird diese Koordinatenposition ermittelt. Bei der Methode `moveTo()` ändert sich die Blickrichtung der Schildkröte nicht.



Vervollständige den Code um die Schildkröte den Rest des Hauses zeichnen zu lassen.

3 Punkte

1. Tabellenkalkulation

Lege in Deinem vom Wettbewerbsleiter angegebenen Arbeitsverzeichnis einen Ordner 9_<NAME>_<VORNAME> an. Kopiere zuerst alle Dateien aus dem Vorgaben-Verzeichnis in diesen Ordner. Speichere darin auch alle Ergebnisse der folgenden Aufgaben.
Öffne die Datei `bruttostromerzeugung.*`.

Quelle: Umweltbundesamt auf Basis AG Energiebilanzen, Sondertabelle Bruttostromerzeugung in Deutschland von 1990 bis 2018

Aufgaben:

- a) Gestalte die Tabelle nach folgenden Kriterien. 5 Punkte
- Spaltenbreite optimieren, Lesbarkeit der Werte garantieren
 - Spaltenüberschriften hervorheben
 - Einstellung Querformat, gesamte Tabelle auf einer Seite
 - Bestimme die Jahressummen der Bruttoenergieerzeugung (gelbe Zellen), arbeite effektiv


	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	AA	AB	AC
1	Bruttostromerzeugung in Deutschland nach Energieträgern in TWh																												
2																													
3		##	##	##	##	##	##	##	##	##	##	##	##	##	##	##	##	##	##	##	##	##	##	##	##	##	##	##	##
4	Braunkohle	##	##	##	##	##	##	##	##	##	##	##	##	##	##	##	##	##	##	##	##	##	##	##	##	##	##	##	##
5	Kernenergie	##	##	##	##	##	##	##	##	##	##	##	##	##	##	##	##	##	##	##	##	97	97	92	85	76	76		
6	Steinkohle	##	##	##	##	##	##	##	##	##	##	##	##	##	##	##	##	##	##	##	##	##	##	##	##	##	##	##	##
7	Gase	41	46	48	51	52	49	56	56	63	63	73	75	78	89	81	89	86	76	68	61	62	81	87	83				
8	Mineralöle	9	8	7	7	6	6	6	9	10	11	12	11	10	10	10	9	7	8	7	6	6	6	6	5				
9	Erneuerbare Ene	25	23	24	26	29	38	39	46	46	57	64	73	89	94	96	##	##	##	##	##	##	##	##	##	##	##	##	
10	Übrige Energietr	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	
11	Bruttostromerzeugung gesamt																												
12																													
13	Quelle: Umweltbundesamt auf Basis AG Energiebilanzen, Sondertabelle Bruttostromerzeugung in Deutschland von 1990 bis 2018																												

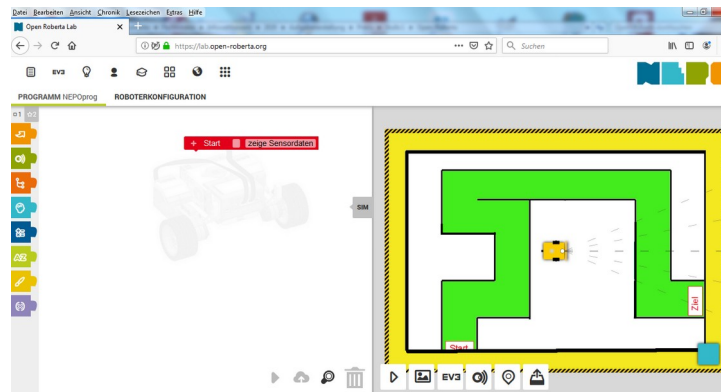
- b) Der Anteil der einzelnen Energieträger an der Stromerzeugung hat sich in den vergangenen Jahrzehnten stark verändert. Stelle diese Entwicklungen in Diagrammen gegenüber. 6 Punkte
- Entwicklung Braunkohle und Kernenergie (1995 bis 2018)
 - Entwicklung Braunkohle und Erneuerbare Energien (1995 bis 2018)
 - Entwicklung Kernenergie und Erneuerbare Energien (1995 bis 2018)
- c) Vergleiche die Anteile der Energieträger zur Stromerzeugung der Jahre 1995 und 2018. 2 Punkte
- d) Stelle die Entwicklung der Bruttostromerzeugung *aller* Energieträger in den Jahren 1995 bis 2018 dar. Verwende dabei nur die Werte der *geraden* Jahreszahlen. 2 Punkte


2. Programmierung

Aufgaben:

- a) Öffne die Seite <https://lab.open-roberta.org> mit Mozilla Firefox oder Google Chrome. Wenn sich die Seite geöffnet hat, wähle als System *Open Roberta Sim*. Klicke auf der rechten Seite SIM an, damit du die Umgebung deines Roboters sehen kannst.

Klicke unter der Umgebung das Symbol  für das Laden eines eigenen Hintergrundbildes an und lade aus dem Ordner VORGABEN das Bild hg_labyrinth hoch. Wenn du das Bild dann nicht siehst, klicke auf Roboterkonfiguration und anschließend erneut auf Programm NEOProg (siehe Screenshot). Falls du diese Umgebung nicht so siehst, informiere deinen Lehrer. 1 Punkt



- b) Verschiebe den gelben Roboter an die Ausgangsposition (Nutze die Schaltfläche ) und anschließend mit der Maus an die Position Start. Das untere Rad berührt dabei die Startlinie. Achte darauf, dass du dies nach jedem Testlauf deines Programms tun musst.

Der Roboter soll nun den grünen Weg zum Ziel fahren, ohne die schwarzen Linien zu berühren. Nutze dazu die Befehle, welche du im linken Bereich des Fensters auswählen kannst. Starte das Programm, indem du unter dem Labyrinth die Starttaste drückst. Der Roboter kann in die Ausgangslage gesetzt werden, indem die entsprechende Taste unter dem Labyrinth gedrückt wird. 7 Punkte

- c) Speichere dein Programm mit dem Namen `loesung_b`, indem du oben links im Bearbeitungs-Menü „exportiere Programm“ anwählst. Wird dabei nicht nach dem Zielordner gefragt, speichert der Browser die Datei im Ordner DOWNLOAD. Kopiere die Datei dann in den vorgegebenen Ordner. 1 Punkt
- d) Setze das blaue Hindernis (rechts unten in der Ecke) an eine beliebige Stelle der langen Strecke oben im Hintergrundbild. Ändere das Programm aus Aufgabe b) so ab, dass der Roboter von allein kurz vor dem Hindernis stoppt. Speichere das Ergebnis wie in Aufgabe c) beschrieben als `loesung_d` ab. 4 Punkte