Aufgabe 1

	Grundgebühr pro Monat	Verbrauchspreis pro kWh
Stromtarif "Watt für wenig"	13,50 €	0,75 €
Stromtarif "Billig Strom"	9,20 €	0,81 €

Schreibe eine Funktion, die den Monatsverbrauch in Kilowattstunden akzeptiert und den monatlichen Rechnungsbetrag beim Tarif "Billig Strom" berechnet. Prüfe die Ergebnisse.

Aufgabe 2

Schreibe eine Funktion, die Grundgebühr, Verbrauchspreis und Verbrauch eines Stromtarifs akzeptiert und den monatlichen Rechnungsbetrag für diesen Stromtarif berechnet. Prüfe die Ergebnisse.

Aufgabe 3

Wir wollen in Großbritannien einkaufen. Die Preise sind dort in britischen Pfund (GBP) angegeben. Wir müssen also umrechnen.

a) Schreibe eine Funktion, die britische Pfund in Euro umrechnet.
   (Kurs: 1 GBP = 1,21 EUR)
   1. Schreibe eine Kurzbeschreibung
   2. Mit welchen Daten soll die Funktion arbeiten?
   3. Definiere einen Namen für die Funktion, gib Eingabewert und Rückgabewert an
   4. ~~Gerüst~~
   5. Rumpf
b) Prüfe die Ergebnisse mit Hilfe einer Tabelle von 0 GBP bis 10 GBP in Schritten von 0.50 GBP.

Aufgabe 4

Wir haben in Großbritannien eingekauft. Die Preise unserer Einkäufe stehen in einer Liste:

Book: Guinness World Records	09.00 GBP
CD: Adele – 30	11.99 GBP
Poster: Butterflies	12.99 GBP
Bicycle Reflective Gilet	24.99 GBP

a) Schreibe eine Funktion, die die Preise der Einkäufe addiert und die Summe in Euro umrechnet. (Kurs: 1 GBP = 1,21 EUR)
   1. Schreibe eine Kurzbeschreibung
   2. Mit welchen Daten soll die Funktion arbeiten?
   3. Definiere einen Namen für die Funktion, gib Eingabewert und Rückgabewert an
   4. ~~Gerüst~~
   5. Rumpf
b) Drucke die Summe für alle 4 Einkäufe.
c) Die Weste (Gilet) ist zu teuer. Drucke die Summe für die ersten 3 Einkäufe.

Aufgabe 5

Schreibe ein Programm, das die Kosten für die beiden Tarife "Watt für wenig" und "Billig Strom"

a) berechnet
b) nebeneinander ausdruckt
c) in ein gemeinsames Diagramm plottet

Aufgabe 6

Laura prüft zwei Angebote für mobiles Internet.
Angebot A: Unbegrenztes Surfen für 8,99 € pro Monat
Angebot B: Preis pro GB: 6,50 €
In den letzten Monaten hat sie ihren Verbrauch notiert:

Mai 1,2 GB
Juni 1,5 GB
Juli 2 GB

a) Schreibe ein Programm, das die Kosten (Mai, Juni, Juli) für Angebot A berechnet
Schreibe ein Programm, das die Kosten (Mai, Juni, Juli) für Angebot B berechnet
b) Vergleiche die Ergebnisse mit Hilfe einer Tabelle
c) Vergleiche die Ergebnisse mit Hilfe eines Diagramms

Aufgabe 7

Herr Häberle hat ein neues Elektroauto. Nach der täglichen Nutzung lädt Herr Häberle den Akku des Elektroautos wieder auf, zu Hause an der Steckdose oder an einer Ladestation. Herr Häberle hat an 3 Tagen notiert, wie lange er gefahren ist und wie lange er geladen hat.

Tag	Was	Dauer in Stunden	Änderung des Ladezustands pro Stunde	Kommentar
Montag	Fahren	2	-33 %	Hohes Tempo
	Laden	13	+5 %	Steckdose
Dienstag	Fahren	5	-18 %	Mittleres Tempo
	Laden	16	+5 %	Steckdose
Mittwoch	Fahren	2	-33 %	Hohes Tempo
	Laden	1	+40 %	Ladestation

Am Montagmorgen ist Herr Häberle mit 100 % Ladezustand losgefahren. Erstelle ein Diagramm, das den Ladezustand des Akkus in Abhängigkeit von der Zeit darstellt.

a) Schreibe eine Funktion mit den Eingangswerten: alter Ladezustand, Dauer, Änderung und dem Rückgabewert: neuer Ladezustand.
   1. Schreibe eine Kurzbeschreibung
   2. Mit welchen Daten soll die Funktion arbeiten?
   3. Definiere einen Namen für die Funktion, gib Eingabewert und Rückgabewert an
   4. ~~Gerüst~~
   5. Rumpf
b) Übertrage die Dauer aus der Tabelle in eine Liste.
c) Beschreibe die Zeitachse durch eine Liste. Die Liste beginnt mit 0. Die folgenden Zeitpunkte berechnest du aus der angegebenen Dauer.
d) Übertrage die Änderung aus der Tabelle in eine Liste.
e) Beschreibe die Ladezustands-Achse durch eine Liste. Die Liste beginnt mit 100. Die folgenden Ladezustände berechnest du mit Hilfe der Funktion aus Teilaufgabe a).
f) Erstelle das Diagramm mit Hilfe der Funktionen des Moduls matplotlib.

Aufgabe 8

a) Erstelle die Klasse "Fahrrad" mit den Eigenschaften:
   - Besitzer
   - Farbe
   - Typ (Touring, Renn, Mountain o. ä.)
   - Anzahl Gänge

b) Erstelle eine Instanz der Klasse "Fahrrad" und gib die Eigenschaften aus.

Aufgabe 9

a) Erweitere die Klasse "Fahrrad". Erstelle die Methoden:
   - fahren
   - klingeln
   - abstellen
   - kilometerstand

b) Erstelle eine Instanz der Klasse "Fahrrad" und rufe die Methoden auf.

Aufgabe 10

a) Erstelle die Eltern-Klasse "Zweirad" mit den Eigenschaften:
   - Besitzer
   - Farbe
   - Typ
   - Anzahl Gänge

   und den Methoden:
   - fahren
   - klingeln
   - abstellen
   - kilometerstand

b) Erstelle die Kind-Klassen "Fahrrad" und "Pedelec", die alle Eigenschaften und Methoden der Eltern-Klasse "Zweirad" erben.

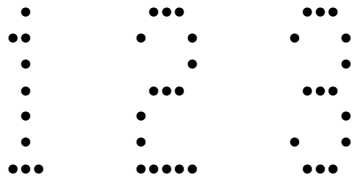
c) Die Klasse "Pedelec":
   - hat zusätzlich die Eigenschaft "Kapazität" (Wattstunden)
   - überschreibt die Methode "abstellen" (warum?)
   - hat zusätzlich die Methode "aufladen"

d) Erstelle eine Instanz der Klasse "Fahrrad" und eine Instanz der Klasse "Pedelec" und rufe alle Methoden auf.

e) Ausgabe der Eigenschaft "Kapazität" der Instanz des "Pedelec".

Aufgabe 11

Die Ziffern auf einem Kassenzettel sind aus Punkten zusammengesetzt. Mit 5x7 Punkten können die Ziffern 0 bis 9 gut lesbar dargestellt werden. Wir beschränken uns auf die Ziffern 1, 2 und 3.



Nicht alle Ziffern auf dem Kassenzettel sehen so aus wie oben. Gelegentlich fehlen Punkte!
Schreibe ein Programm, das erkennt, ob es sich bei einem Muster mit 5x7 Punkten um eine der Ziffern 1, 2 oder 3 handelt. Wenige fehlende Punkte sollen toleriert werden.
Das Programm soll die erkannte Ziffer (oder die erkannten Ziffern) angeben. Wird keine Ziffer erkannt, soll „keine“ angegeben werden.

a) Definiere die (ungestörten) Muster der Ziffern 1, 2 und 3 durch Listen 7 Strings. Jeder String enthält das Muster einer Zeile.
b) Schreibe eine Funktion b), die zwei beliebige Muster akzeptiert, und die passenden Zeilen zählt.
c) Schreibe eine Funktion c), die ein fragliches Muster und das Minimum der passenden Zeilen akzeptiert. Funktion c) soll Funktion b) aufrufen und das fragliche Muster mit dem ungestörten Muster der Ziffern 1, 2 und 3 vergleichen. Funktion c) soll einen String mit den erkannten Ziffern zurückgeben.
d) Definiere ein gestörtes Muster der Ziffer 2. Ein Punkt fehlt.
e) Schreibe ein Programm, das Funktion c) aufruft, und ein gestörtes Muster prüft.

Aufgabe 12

Schreibe einen Unittest für die Funktionen von Aufgabe 11.

Aufgabe 13

Schreibe einen Unittest für die Funktionen von Aufgabe 5.

Aufgabe 14

Schreibe einen Unittest für die Funktionen von Aufgabe 6.

Aufgabe 15

Ein einfacher Saugroboter hat drei Zustände: STANDBY – GERADEAUS_FAHREN – DREHEN.
Die Bedingungen für die Zustandsübergänge sind: an_aus – kollision_erkannt – genug_gedreht.
Zu Beginn soll der Saugroboter im Zustand STANDBY sein.

a) Welche Zustandsübergänge gibt es?
b) Welche Bedingung gehört zu welchem Zustandsübergang?
c) Zeichne ein Zustandsübergangsdiagramm mit den gegebenen Zuständen und Übergängen.
d) Schreibe ein Programm, das den Saugroboter steuert. Die Zustandsübergänge werden durch die Tasten a – k – g der PC-Tastatur gesteuert. Verwende die Bibliothek StateMachine.
    1. Schreibe eine Kurzbeschreibung
    2. Importiere die notwendigen Bibliotheken
    3. Schreibe eine Begrüßung mit einer Erklärung des Gebrauchs der Tasten
    4. Erzeuge das Objekt state_machine
    5. Vorbelegung des Zeichens von der Tastatur
    5a. Timer für periodische Ausführung
    6. Schreibe die Funktionen mit den print-Ausgaben:
        - standby()
        - geradeaus_fahren()
        - drehen()
    7. Verwende die Funktion taste_gedrueckt(zeichen) von Ampel.py
    8. Schreibe die Funktionen zur Steuerung der Zustandsübergänge:
        - an_aus()
        - kollision_erkannt()
        - genug_gedreht()
    9. Definiere die Zustände STANDBY – GERADEAUS_FAHREN – DREHEN
    10. Zustandsübergänge hinzufügen
    11. Loop

Aufgabe 16

Schreibe ein Programm mit GUI, das drei verschiedene Smileys untereinander anzeigt. Smileys findest du in folgenden Dateien: slightly_smiling_face.png, neutral_face.png, slightly_frowning_face.png.

Aufgabe 17

Schreibe ein Programm mit GUI, das die drei Smileys (von Aufgabe 16) nebeneinander anzeigt.

Aufgabe 18

Schreibe ein Programm mit GUI, das die drei Smileys (von Aufgabe 16) diagonal anzeigt.
Die Bilderzeugung und die Platzierung des Smileys sollen in eine Funktion ausgelagert werden.
Hinweis: Der Aufruf tk.PhotoImage() muss im Hauptprogramm stehen.

Aufgabe 19

Schreibe ein Programm mit GUI, das auf Knopfdruck ein Smiley (von Aufgabe 16) anzeigt.
Hinweis: Der Aufruf tk.PhotoImage() muss im Hauptprogramm stehen.

Aufgabe 20

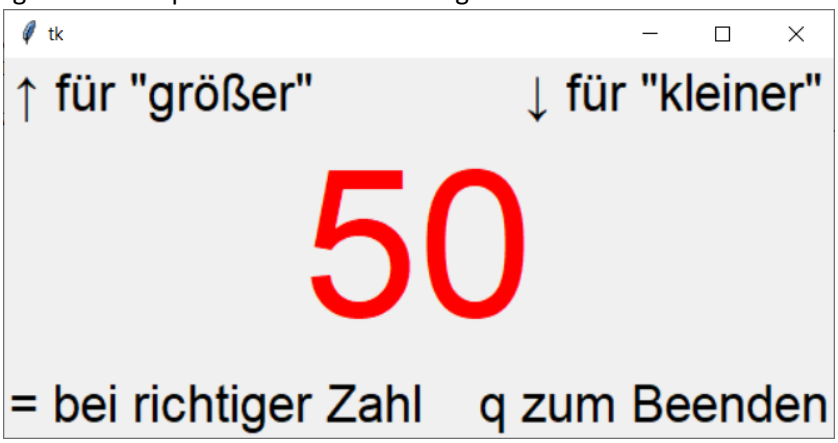
Schreibe ein Programm mit GUI, das drei Smileys in einem Gitter von 7 Zeilen und 11 Spalten anzeigt. Platziere:

- das erste Smiley in der oberen linken Ecke
- das zweite Smiley im Zentrum
- das dritte Smiley in der unteren rechten Ecke

Hinweis: Die Smileys sind 72 Pixel breit und 72 Pixel hoch.

Aufgabe 21

Schreibe das Programm "Computer errät Zahl" mit folgendem GUI:



a) Erzeuge ein Fenster und platziere Bedienungsanleitung-1 und -2.

b) Erzeuge eine Steuervariable für die geratene Zahl. Platziere die Ausgabe der geratenen Zahl zwischen Bedienungsanleitung-1 und -2.

c) Binde die Betätigung einer Computertaste an das Fenster. Wenn eine Taste betätigt wird, soll die Funktion onKeyPress aufgerufen werden.

d) Erzeuge die globalen Variablen:
untere
obere
versuche

e) Schreibe die Funktion rate_zahl() zur Berechnung der geratenen Zahl.

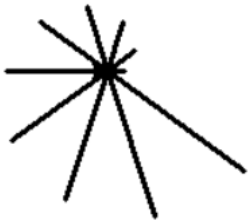
f) Schreibe die Funktion onKeyPress. Die Funktion soll auf die Tasten ↓ ↑ = q reagieren. Arbeite mit den globalen Variablen.
   1. Bei ↓ die obere Grenze anpassen, versuche hochzählen
   2. Bei ↑ die untere Grenze anpassen, versuche hochzählen
   3. Bei = einen Kommentar zeigen
   4. Bei q das Fenster schließen

g) Schreibe die Funktion zeige_Kommentar(). Die Funktion gibt die Anzahl der Versuche aus, die der Computer benötigt hat.

Aufgabe 22
Zeichne ein Sechseck. Verwende Turtle.

Aufgabe 23
Zeichne einen Stern, der nur aus Strahlen besteht. Verwende Turtle.

a) Beginne mit 10 Strahlen
b) Definiere eine Variable für die Anzahl der Strahlen. Berechne daraus den Winkel für den Stern.
c) Zeichne einen Stern mit Strahlen wachsender Länge.



Aufgabe 24
Zeichne ein Vieleck. Verwende Turtle.

a) Hauptprogramm: Frage den Benutzer nach der Anzahl der Ecken, 1 bis 10 sind erlaubt. Hinweis: Verwende die Methode numinput().
b) Funktion: Schreibe die Funktion vieleck. Eingabewert ist die Anzahl der Ecken.
c) Hauptprogramm: Rufe die Funktion vieleck auf.
d) Erweitere b): Der berechnete Winkel soll auf der Zeichenfläche erscheinen.

Aufgabe 25
Teste die Codierung der Textdatei.

a) Schreibe ein paar Zeilen Text mit ä, ö und ü in eine Textdatei. Verwende den Editor.
b) Speichere den Text in zwei Dateien: Einmal UTF-8 codiert, einmal ANSI-codiert.
c) Ändere das Programm zwei_textcodierungen.py so, dass es deine Dateien liest.

Aufgabe 26
Gib den Dateipfad an.

a) Starte das Programm datei_liegt_woanders.py. Was bedeutet die Fehlermeldung?
b) Korrigiere das Programm.

Aufgabe 27
Ändere den Inhalt der Datei.

a) Editiere das Programm lesen_ersetzen_schreiben.py. Ersetze ein weiteres Wort im ursprünglichen Text.
b) Ersetze mehrfach vorkommende Worte im ursprünglichen Text.

