

LS 1 – Hallo Python

Ausgangssituation

Die Entwicklung von Programmen spielt in der Automatisierungstechnik eine immer größere Rolle. Deshalb sollen Sie sich in die Programmiersprache Python einarbeiten und erste kleine Programme erstellen.



Aufgabe 1 – Einarbeitung in erste Entwicklungswerkzeuge

Machen Sie sich mit der Entwicklungsumgebung Python-IDLE (alternativ Visual Studio Code) und dem Zeichenprogramm Structorizer vertraut.



- Erstellen Sie ein **Taschenrechner** mit Python, welcher einen Operator (+, -, *, /) und zwei Zahlen einliest. Je nach Operator wird die entsprechende Rechnung durchgeführt und das Ergebnis angezeigt. Speichern Sie das Programm unter dem Namen **aufgabe_ls1_01.py** ab.
- Erstellen Sie einen **Temperaturumrechner** mit Python. Das Programm soll zwischen Grad Celsius und Grad Fahrenheit umrechnen können. Speichern Sie das Programm unter dem Namen **aufgabe_ls1_02.py** ab.
- „**Parallelschaltung**“ - Erstelle ein Programm, welches die Widerstandswerte zweier parallel geschalteter Widerstände einliest und anschließend den Gesamtwiderstand ausgibt. Speichere das Programm unter dem Namen **aufgabe_ls1_03.py** ab.
- „**Kugel**“ - Erstelle ein Programm, welches den Radius einliest und anschließend die Oberfläche und das Volumen einer entsprechenden Kugel ausgibt. Speichere das Programm unter dem Namen **aufgabe_ls1_04.py** ab.
- Erstellen Sie je ein **Struktogramm** oder einen **PAP** für die Aufgaben (a) und (d).

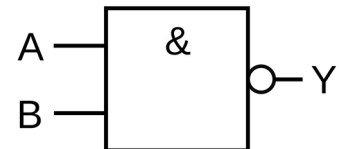
Aufgabe 2 – Reflexion der benutzten Programmierkonstrukte und Programmierwerkzeuge

- Mit welcher Anweisung können Sie in Python eine Ausgabe auf dem Bildschirm realisieren? Geben Sie beispielhaft den Text „LF5 ist der didaktische Höhepunkt der Woche aus.“
- Mit welcher Anweisung können Sie in Python einen Text einlesen?
- Was ist beim Einlesen einer Zahl zu beachten? Lesen sie beispielhaft den Radius (Kommazahl) ein und speichern Sie ihn in einer Variable `radius`.
- Was versteht man unter einer Variablen?
- Nennen Sie beispielhaft einige häufig genutzte Datentypen für Variablen?
- Was versteht man unter einer Anweisung?
- Am Anfang vieler Python-Programme finden Sie `import`-Anweisungen. Wozu dienen diese?
- Wozu dienen Kommentare? Wie können Sie Kommentare in Python kennzeichnen?
- Welche Anweisungen gibt es in Python, um Schleifen aufzubauen?
- Zählen Sie beispielhaft in einer `WHILE`-Schleife rückwärts von 10 bis 1.
- Zählen Sie beispielhaft in einer `FOR`-Schleife von 5 bis 25.

- l) Welche Anweisungen gibt es in Python, um Fallunterscheidungen aufzubauen?
- m) Wozu benutzt man bei der Programmierung einen (Text-)Editor?
- n) Was ist der Unterschied zwischen einem Interpreter und einem Compiler?
- o) Wird bei Python mit einem Compiler oder mit einem Interpreter gearbeitet?
- p) Nennen Sie zwei Entwicklungsumgebungen, die Sie bei der Erstellung von Python-Programmen nutzen können?
- q) Welche Entwicklungsumgebung wird direkt mit Python ausgeliefert?
- r) Worauf müssen Sie achten, wenn Sie Visual-Studio-Code zur Python-Entwicklung nutzen wollen?
- s) Was ist der Unterschied zwischen einem Textverarbeitungsprogramm und einem Text-Editor?
- t) Neben Python gibt es noch weitere Programmiersprachen. Nennen Sie drei weitere.
- u) Lesen Sie sich den einführenden Text über Python durch (FISI-LF5-LS1-Text1). Welche Anwendungsbereiche für Python werden dort genannt?
- v) Wofür steht die Abkürzung IDE?

Aufgabe 3 - Programm erweitern

Erweitern Sie das Hilfsprogramm zur Anzeige logischer Funktionen und führen Sie folgenden Arbeiten aus:



- a) Dokumentieren Sie das Original-Programm mit einem Struktogramm und einem Programmablaufplan (PAP).
- b) Erweitern Sie das Programm, so dass es auch Wertetabellen für die folgenden logischen Verknüpfungen darstellen kann:
 - NAND
 - NOR
 - ÄQUIVALENZ
 - ANTIVALENZ

Hinweise:

- Das Originalprogramm finden Sie in der Datei **aufgabe_ls1_05_orginal.py**.
- Speichere das Programm unter dem Namen **aufgabe_ls1_05.py** ab.

Aufgabe 4 - Mit Text- und CSV-Dateien arbeiten

- a) Schreiben Sie ein Python-Programm, welches eine Text-Datei einliest und deren Inhalt auf dem Bildschirm ausgibt. Speichere das Programm unter dem Namen **aufgabe_ls1_06.py** ab.
- b) Schreiben Sie ein Python-Programm, welches alle 2 Sekunden einen Zeitstempel und einen beliebigen Text auf dem Bildschirm ausgibt und in eine Text-Datei schreibt. Der Name der Text-Datei soll vorher erfragt werden. Speichere das Programm unter dem Namen **aufgabe_ls1_07.py** ab.
- c) In der CSV-Datei **widerstand.csv** stehen zeilenweise jeweils 3 Widerstandswerte in Ohm. Schreiben Sie ein Python-Programm, welches die drei Widerstände addiert und die 3 Widerstandswerte und die Summe wieder in eine CSV-Datei namens **gesamtwiderstand.csv** schreibt. Zur Kontrolle soll das Programm auf dem



Bildschirm die Namen der Ein- und Ausgabedatei, die Anzahl der Berechnungen, das aktuelle Datum und die Uhrzeit ausgeben. Speichere das Programm unter dem Namen **aufgabe_ls1_08.py** ab.

- d) Erstellen Sie jeweils ein **Struktogramm** oder einen **PAP** für die Aufgaben (a) und (d).

Aufgabe 5 – Reflexion Dateien



- a) Wofür steht die Abkürzung CSV?
- b) Wozu werden CSV-Dateien typischerweise genutzt?
- c) Welche Alternativen gibt es zu CSV-Dateien?
- d) Was ist der Unterschied zwischen einer Text- und einer Binär-Datei?
- e) Nennen Sie einige Beispiele für typische Binärdateien. Welche Endungen zu ihrer Kennung werden genutzt?
- f) Zeigen Sie, wie man eine Text-Datei mit Python einlesen kann?
- g) Welches Modul zum Umgang mit CSV-Dateien gibt es in Python?

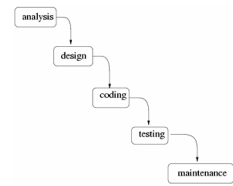
Aufgabe 6 – Visualisierung



- a) Wozu wird ein Kontext-Diagramm genutzt?
- b) Wozu werden Struktogramme und PAPs genutzt?
- c) Wofür steht die Abkürzung PAP?
- d) Wie werden Struktogramme auch genannt?
- e) Was versteht man unter strukturierter Programmierung?
- f) Was versteht man unter einfachen linearen Programmen? Wie wird die hierbei genutzte Struktur auch genannt?
- g) Was ist der Unterschied zwischen einer kopfgesteuerten und einer fußgesteuerten Schleife? Wie werden sie in Struktogrammen dargestellt?
- h) Welche Möglichkeiten zur Darstellung von Schleifen gibt es bei PAPs?
- i) Bei Programmieren mit Python werden häufig andere vorhandene Module genutzt. Welche Möglichkeiten zur Visualisierung gibt es?

Aufgabe 7 - Aufgaben zu Vorgehensmodellen

- a) Bei der Anwendungsentwicklung gibt es das klassische Wasserfallmodell als Vorgehensmodell. Skizzieren Sie dieses.
- b) Welche Kritikpunkte gegen die alten eher statischen Vorgehensmodelle wurden laut?
- c) Nennen Sie neben dem Wasserfallmodell noch andere eher klassische Vorgehensmodelle.
- d) Nennen Sie ein neues eher dynamisches Vorgehensmodell zur Anwendungsentwicklung.
- e) In den Vorgangsmodellen wird oft zwischen einer Analyse- und einer Design-Phase unterschieden. Was ist der Unterschied?

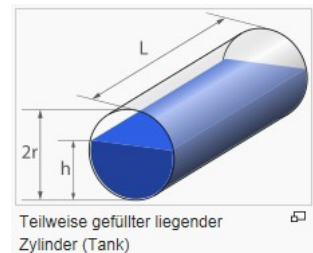
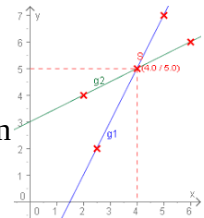


Aufgabe 8 – Zusatzaufgaben

Lösen die folgenden Aufgaben mit Python. Erstellen Sie geeignete Diagramme zur Visualisierung.



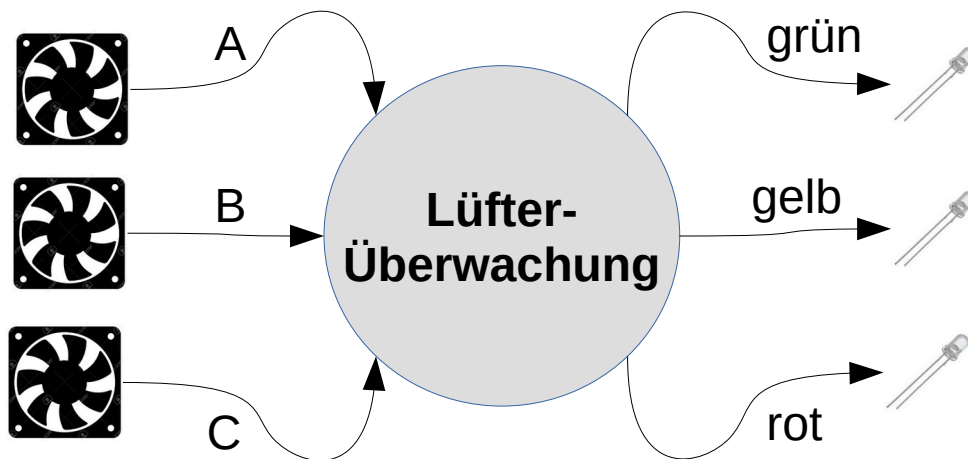
- a) Verschiedene Datenlogger erzeugen Textdateien, in denen Dezimalzahlen stehen. Pro Datei kann es viele Zahlen geben, wobei jede Zahl in einer eigenen Zeile steht. Bei manchen Dateien wird ein Punkt als Dezimalkennzeichen genutzt (3.14), bei manchen Dateien wird ein Komma als Dezimalkennzeichen genutzt (3,14). Erstellen Sie mit Python einen Konvertierer, der in beide Richtungen konvertieren kann. Speichern Sie das Programm unter dem Namen **aufgabe_ls1_09.py** ab.
- b) **„Geradenschnittpunkt“** - Erstellen Sie ein Programm zur Berechnung des Schnittpunkts zweier Geraden:
Zuerst werden die Koeffizienten der beiden Geraden eingelesen.
Anschließend berechnet das Programm den Schnittpunkt der beiden Geraden und gibt diesen aus. Speichere das Programm unter dem Namen **aufgabe_ls1_10.py** ab.
- c) **„Tankfüllung“** - Gegeben ist ein Tank in Form eines liegenden geraden Zylinders mit dem Radius r und der Länge l , sowie die Füllhöhe h . Erstelle ein Programm, welches den Radius, die Länge und die Füllhöhe einliest und anschließend das Volumen (Tankinhalt) ausgibt. Speichere das Programm unter dem Namen **aufgabe_ls1_11.py** ab.



Aufgabe 9 – „Lüfterüberwachung“

- In einem Gehäuse sind drei Lüfter A, B, C angeordnet. Jeder Lüfter hat einen Sensor, der ein Low-Signal (0) liefert, wenn der betreffende Lüfter arbeitet. Ist der Lüfter ausgefallen, liefert der Sensor ein High-Signal (1). Entwerfen Sie eine Überwachungsschaltung, die drei Kontroll-LEDs folgendermaßen ansteuert:
 - die grüne LED soll leuchten (High-Signal), wenn alle drei Lüfter arbeiten,
 - die gelbe LED soll leuchten (High-Signal), wenn einer der Lüfter (gleich welcher) ausgefallen ist,
 - die rote LED soll leuchten (High-Signal), wenn zwei oder alle drei Lüfter ausgefallen sind.

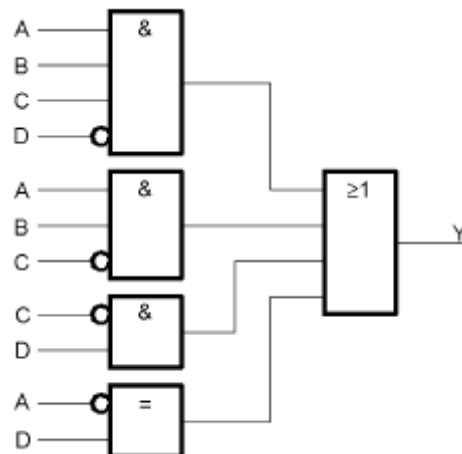


**Hinweise:**

- Die Sensor Signale der Lüfter sollen durch entsprechende Eingaben simuliert werden.
- Die LEDs sollen durch entsprechende Ausgaben simuliert werden.
- Speichere das Programm unter dem Namen **aufgabe_ls1_12.py** ab.

Aufgabe 10 – „Schaltungs-Realisierung (I)“

Schreiben Sie ein Python-Programm zur Realisierung der folgenden Schaltung:

**Hinweise:**

- Die Ein- und Ausgänge sollen durch entsprechende Eingaben und Ausgaben simuliert werden.
- Speichere das Programm unter dem Namen **aufgabe_ls1_13.py** ab.