

Computergestützte Mathematik zur linearen Algebra – 3. Übungsblatt

Vergessen Sie nicht Ihren Quelltext sinnvoll zu kommentieren!

Aufgaben 9 und 10 müssen Sie erst nächste Woche vorzeigen.

Aufgabe 9: (for-Schleife)

Befehle: `input`, `int()`, `for`, `if`, `range`

Erstellen Sie ein PYTHON-Skript `meine_erste_schleife.py`:

- Erstellen Sie eine Funktion `summe()`, die über die Eingabekonsole eine ganzzahlige Zahl N einliest und mit Hilfe einer `for`-Schleife die Summe $\sum_{n=1}^N n^2$ berechnet und zurückgibt.
- Testen Sie Ihre Funktion mit den Zahlen 2, 9, -2 und 23

Aufgabe 10: (for-Schleifen und if-Abfragen)

Befehle: `for`, `if`, `elif`, `else`, `continue`

Erstellen Sie ein PYTHON-Skript mit dem Namen `zahlen_analyse.py`:

- Schreiben Sie eine Funktion `analyse(eingabe)`, die eine Liste oder ein Tupel `eingabe` übergeben bekommt und mittels einer `for`-Schleife jedes Element folgendermaßen „analysiert“:
 - Als erstes soll ein Text ausgegeben werden der zeigt, welche Zahl analysiert wird.
 - Sollte das Element 0 sein, so wird es nicht weiter analysiert (was ausgegeben wird) und die Schleife geht sofort zum nächsten Element (gucken Sie sich dafür den Befehl `continue` an).
 - Dann soll ausgegeben werden ob das Element positiv oder negativ ist.
 - Danach soll geprüft werden, ob die Zahl durch 2 teilbar ist, was auch ausgegeben wird.
 - Für die Übersichtlichkeit soll am Ende der Analyse eine neue Zeile ausgegeben werden.

Für `analyse((1,0,-2))` könnte die Ausgabe beispielsweise so aussehen:

```
1 wird analysiert:
```

```
1 ist positiv
```

```
1 ist nicht durch 2 teilbar
```

```
0 wird analysiert:
```

```
0 wird in der Analyse ignoriert
```

```
-2 wird analysiert:
```

```
-2 ist negativ
```

```
-2 ist durch 2 teilbar
```

WICHTIG: Verwenden Sie sinnvoll alle Befehle die oben stehen, aber nicht die Funktion `range`.

- Testen Sie `analyse` mit `a=(2,-1,0,6.8,-0.59)`.

Aufgabe 11: (Funktionsparameter)

Erstellen Sie ein Skript mit dem Namen *Aufgabe11.py*:

- (a) Schreiben Sie eine Funktion `plotte_funktion(f,a,b,N)` welche die Funktion $f(x)$ auf dem Intervall $[a, b]$ mit N Punkten plottet. `f` ist hierbei das Functionhandle von $f(x)$, `a`, `b` und `N` sind Zahlen.
- (b) Erweitern Sie den Kopf von `plotte_funktion` so, dass $N = 300$ gesetzt wird, falls N nicht übergeben wird.
- (c) Testen Sie Ihre Funktion mit

$$f(x) = 0,3x^2 - 1,7\sin(4x),$$

wobei $a = -1$, $b = 4$ und $N = 10$ bzw. ohne N .

Aufgabe 12: (Die Hilfe verwenden)

Befehle: `if`, `help()`

Erstellen Sie ein Skript mit dem Namen *Aufgabe12.py*:

- (a) Schreiben Sie eine Funktion `(v,o)=zylinder_neu(r,h)`, die als Eingabeparameter den Radius `r` und die Höhe `h` eines Zylinders übergeben bekommt, das Volumen und die Oberfläche des Zylinders berechnet und diese Werte zurückgibt (also Aufgabe 6).
- (b) Erweitern Sie `zylinder_neu(r,h)` so, dass (mit `print`) eine Warnung ausgegeben wird, falls `r` oder `h` kleiner als 0 sind. In diesem Fall sollen die Rückgabewerte 0 sein.
- (c) Testen Sie `zylinder_neu(r,h)` mit den Werten $r = 5.39$ und $h = 4.73$. Benutzen Sie den `format`-Befehl (oder `%`) um folgende Ausgabe zu erhalten (ohne Zeilenumbruch und mit den richtigen Werten):

Ein Zylinder mit dem Radius r und der Höhe h hat ein Volumen von v und eine Oberfläche von o .

v soll hierbei in *fixed-point*-Darstellung mit drei Nachkommastellen und o in *Exponenten*-Darstellung mit vier Nachkommastellen angezeigt werden.

Hinweis: Mit dem Befehl `help()` (ohne Parameter) können Sie in der Hilfe stöbern.

Hinweis: Da Dienstag und Mittwoch aufgrund von Feiertagen an diesen beiden Tagen keine Übungen stattfinden, können die betroffenen Studenten ihre Lösungen in dieser Woche in einer anderen Übungsgruppe oder in ihrer Übung in der Woche danach (zusammen mit Blatt 4) vorzeigen.

Besprechung in den Übungen vom 30.10.-3.11. 2017.