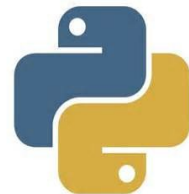


Struktogramme

Ihr habt vor einigen Wochen die Programmiersprache *Python 3* kennengelernt. Damit habt ihr gleichzeitig die ersten Schritte in die objektorientierte Programmierung getan. Ein paar kleine Programme hast du schon geschrieben. Bei der Arbeit mit der Turtle bestanden die Aufgaben meistens darin, der *Turtle* zu sagen was er tun soll. Man nennt diese Befehle in der Programmierung *Anweisungen*. Mehrere Anweisungen können auch zu *Funktionen* zusammengefasst werden. Funktionen werden mit dem Schlüsselwort `def` eingeleitet



```
def funktionsname(Parameterliste):  
    Anweisung(en)
```

Die Parameterliste besteht aus einem oder mehreren Bezeichnern, die durch Kommata getrennt sind.

Beispiel:

```
#!/usr/bin/python  
  
def add(x,y):  
    """Gib x plus y zurueck."""  
    return x + y  
  
print add(2,3)
```

Ein Programm besteht aus einer Abfolge von mehreren Anweisungen. Um mit einem Programm ein Problem zu lösen, müssen mitunter viele Anweisungen und Methoden verwendet werden. Man bezeichnet die Menge aller Anweisungen auch als *Algorithmus*.

Definition: Unter einem *Algorithmus* (auch Lösungsverfahren) versteht man eine genau definierte Handlungsvorschrift zur Lösung eines Problems oder einer bestimmten Art von Problemen in endlich vielen Schritten.

Zum Beispiel ist ein Kochrezept ein Algorithmus für die Fertigung einer Speise.

----- ✂ Schnipp Schnapp ✂ -----
Aufgabe: Nenne Beispiele für Algorithmen aus deinem täglichen Leben. Formuliere den Algorithmus mit umgangssprachlichen Mitteln.
----- ✂ Schnipp Schnapp ✂ -----

Um die Programmierung zu erleichtern gibt es aber noch mehr Elemente in einer Programmiersprache.

Dazu gehören:

1. Schleifen (Endlosschleifen, Zählschleifen, Solange-Schleifen)
2. Kontrollstrukturen (Auswahl oder Entscheidung)

----- ✂ Schnipp Schnapp ✂ -----
Aufgabe: Wiederhole die Syntax der Kontrollstruktur und der Schleifen.
----- ✂ Schnipp Schnapp ✂ -----

Um aber ein Programm zu schreiben muss man zuerst den Algorithmus formulieren. Wir haben das im Unterricht die auch *Programmidee* genannt. Die Programmidee kann entweder in normaler Sprache oder in Pseudocode formuliert werden. Es gibt aber noch eine sehr übersichtliche Weise der Darstellung. Diese nennt man *Struktogramm*.

Für die einzelnen Programmelemente gibt es jeweils eine eigenes Struktogramm.

1. Für die Darstellung mehrerer Anweisungen wird das folgende Element verwendet...

1. Befehl
2. Befehl
...
letzter Befehl

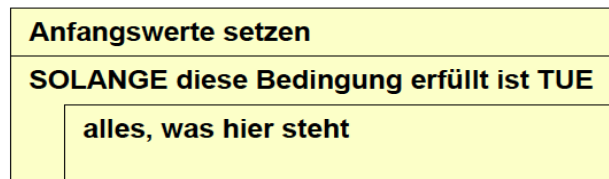
2. ... und für die Kontrollstruktur das folgende. Wenn man nur eine einfache Auswahl verwenden will, so bleibt das Feld bei der Alternative (d.h. für „falsch“) einfach frei.

Diese Aussage ist	
wahr	falsch
hier steht, was geschieht, wenn die Bedingung erfüllt ist	hier steht, was geschieht, wenn die Bedingung <i>nicht</i> erfüllt ist (ggf. nichts)

3. Eine Zählschleife sieht als Struktogramm so aus:

FÜR zählvariable VON anfangswert BIS endwert TUE
alles, was hier steht

4. Für eine Solange- (oder While-) Schleife nimmt man das:

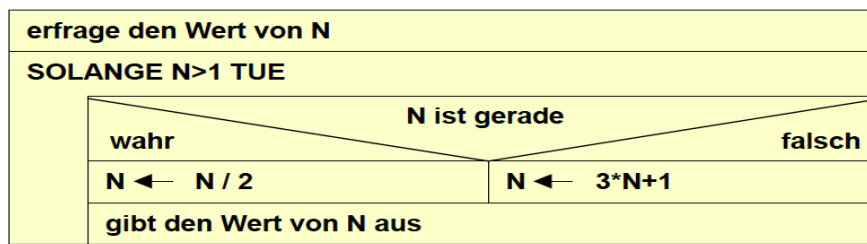


Hinweis: Es gibt neben dieser sogenannten vorprüfenden Solange-Schleife auch noch eine nachprüfende Schleife. Dort wird erst eine Anweisung ausgeführt und dann erst die Bedingung für den Abbruch geprüft.

Neben diesen wichtigen Elementen eines Struktogrammes gibt es noch weitere. Aber diese sollen erstmal für unsere Zwecke reichen.

-----✂ Schnipp Schnapp ✂-----

Aufgabe 1: Gegeben ist das folgende Struktogramm.



- Mache einen „Trockentest“ des Programms und finde heraus was das Programm tut.
- Schreibe das Programm mit Python.
- Teste das Programm für unterschiedliche Anfangswerte von N. Was stellst du fest?

Aufgabe 2: Zeichne ein Struktogramm zu dem Problem: „Finde alle Teiler einer beliebigen ganzen Zahl und gib die Anzahl der Teiler als Ergebnis aus.“