

1. PNK-Übungsblatt

Algorithmisches Denken und Programmiergrundlagen

Aufgabe 1 Algorithmisches Denken

Beschreiben Sie den Beginn Ihres Tagesablaufs, zum Beispiel Ihr Frühstück oder die Zeit zwischen Aufstehen und Erreichen des Vorlesungssaals (Stichpunkte genügen). An Hand der Beschreibung soll lückenlos vom Leser nachvollzogen werden können was Sie tun. Wählen Sie den Detailgrad so, dass Sie circa 20 Stichpunkte erhalten.

Hinweis: Achten Sie darauf Angefangenes auch zu beenden: Zwischen den Punkten

- Einsteigen in die Straßenbahn
- Betreten des Vorlesungssaals

sollten Sie aus der Straßenbahn wieder aussteigen.

Aufgabe 2 Visualisieren von Algorithmen

Programmablaufpläne dienen der grafischen Darstellung von Programmabläufen.

- Ein unausgefüllter Punkt gibt den Start des Ablaufs an
- ausgefüllte Punkte geben Enden des Ablaufs an
- Rechtecke geben Tätigkeiten an
- Rauten geben Entscheidungen (die Auswahl zwischen mehreren Möglichkeiten) an
- Pfeile verbinden diese Elemente miteinander entsprechend der zeitlichen Reihenfolge

a) Erstellen Sie einen Programmablaufplan, der Ihren Tagesablauf aus Aufgabe 1a visualisiert.

b) Sie haben in der Übung **Selectionsort** kennengelernt. Dabei wird eine ungeordnete Liste **u** von Zahlen in eine geordnete Liste **g** überführt. Zu Beginn enthält **u** alle zu sortierenden Zahlen und **g** ist leer. Der Algorithmus sucht und entnimmt das kleinste Element aus **u** und fügt es (hinten) an **g** an. Dies wird solange wiederholt, bis **u** leer ist. Als Ergebnis wird **g** ausgegeben.

Erstellen sie einen Programmablaufplan, der den Selectionsort-Algorithmus visualisiert.

Aufgabe 3 Programmiervorbereitung und Rechnen in Python

a) Installieren Sie auf Ihrem Rechner einen Python-Interpreter Ihrer Wahl, zum Beispiel Python(x,y) mit Spyder, wie ausführlich erklärt in folgendem Tutorial <https://www.youtube.com/watch?v=FC7vjuegK88>

b) Was tun in Python die folgenden Eingaben?

- `2 + 2`
- `'2' + '2'`
- `'2' + 'text'`
- `2 * 3`
- `2 ** 3`
- `liste = []`
- `liste.append('element')`
- `min(-5,3)`
- `zahlen = range(1,101)`
`sum(zahlen)`
- `if(2<3):`
 `7//2`
 `elif(5<10):`
 `7**2`
 `else:`
 `print('Fehler')`
- `if(7=7):`
 `print('funktioniert')`
 `else:`
 `print('funktioniert nicht')`
- `i=j=3`
`while(i<16.5):`
 `i *= 2`
 `i = i - 2.5`
 `j += i`
 `print(j)`

Hinweis: Achten Sie besonders auf Hochkommata, neue Zeilen und Einrückungen.
Hilfreiche Tutorials zum Programmieren in Python sind im Netz zu finden, unter Anderem unter:

<http://www.learn-to-program.net/using-numbers/>

<https://docs.python.org/2/tutorial/introduction.html>