

Einführung in die Programmierung - WS 2016/2017

3. Übungsblatt

Abgabe: 18.11.2016 - 10:00

Aufgabe 1: CodingStyle

(4 Punkte)

Sie erhalten von Ihrem Übungspartner folgenden Code:

```
1 import math, numpy
2
3 variable1 = 2
4 a* = 3
5 3a = a*a*a;
6
7 string = "This placeholder text is gonna be HUGE. You are disgusting. I know words. I ↵
      have the best words. I think the only difference between me and the other placeholder↵
      text is that I am more honest and my words are more beautiful. Lorem Ipsum is ↵
      unattractive, both inside and out. I fully understand why it is former users left it ↵
      for something else. They made a good decision. You know, it really does not matter ↵
      what you write as long as you have got a young, and beautiful, piece of text."
8
9 String = "another string"
10
11 len=len(String)
12 print(len)
13 x = 12,1
14 y = 1.02+ 12.2+
15 23.1 +x+
16 0.123
17
18 //prints "hello World"
19 print( string + String)
20
21 z = 1
22 z=z+1 #increment z
23
24 foo
25 bar = 1
26 final_Name_for_Something = "That other text? Sadly, it is no longer a 10."
27 if not foo is None:
28     print(final_Name_for_Something)
29 if(bar == True):
30     foo = bar
31     if(bar is foo):
32         print(foo)
```

Was ist am Code Ihres Übungspartners falsch und was entspricht nicht dem Coding-Style, der durch **PEP8** gegeben ist? Finden Sie jeweils zwei **Syntax**-, **Semantik**-, **Pragmatik**- und **Coding-Style**-Verstöße.

Aufgabe 2: Zahlen sortieren

(4 Punkte)

Lesen Sie drei Ganzzahlen ein und speichern Sie sie in drei Variablen **a**, **b** und **c**. Anschließend sollen die Zahlen nach aufsteigendem Wert sortiert ausgegeben werden. Sortieren Sie die Zahlen dazu mit Hilfe von **if**- und **else**-Ausdrücken. Wie viele **if**-Abfragen benötigen Sie maximal? Wie viele Möglichkeiten gibt es die Variablen anzuordnen?

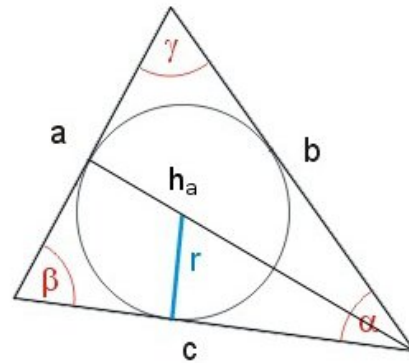
Hinweis: Die Verwendung eines Sortieralgorithmus ist nicht notwendig. Hier soll ein naiver Ansatz verwendet werden.

Aufgabe 3: Inkreis

(6 Punkte)

Der Inkreis eines Dreiecks ist definiert als der größte Kreis, der komplett innerhalb der Fläche des Dreiecks (A_D) liegt. Der Kreis berührt dabei alle drei Seiten des Dreiecks.

Ihre Aufgabe besteht darin, drei Programme zu schreiben, die jeweils den Umfang und den Flächeninhalt des Kreises bestimmen.



Bekannte Formeln zur Bestimmung des Radius des Inkreises sind:

$$r = \frac{2A_D}{a + b + c} \quad (1)$$

$$r = \sqrt{\frac{(s-a)(s-b)(s-c)}{s}} \quad \text{mit} \quad s = \frac{a+b+c}{2} \quad (2)$$

$$r = \frac{a}{\cot(\frac{\beta}{2}) + \cot(\frac{\gamma}{2})} = \frac{b}{\cot(\frac{\alpha}{2}) + \cot(\frac{\gamma}{2})} = \frac{c}{\cot(\frac{\alpha}{2}) + \cot(\frac{\beta}{2})} \quad (3)$$

Der Benutzer soll bei den jeweiligen Programmen unterschiedliche Variablen zu Beginn festlegen können:

- a) Gegeben a, b und c
- b) Gegeben h_a, β und γ
- c) Gegeben h_a, β und a

Alle Längen sollen in cm und alle Winkel in Grad angegeben werden.

Aufgabe 4: Wahrheitstabellen

(6 Punkte)

In der Vorlesung haben Sie bereits die beiden boolschen Werte **True** und **False** sowie deren Verknüpfung mit sogenannten boolschen Operatoren kennengelernt. Die Verknüpfungen kann man anhand von *Wahrheitstabellen* vereinfacht darstellen. Unten finden Sie ein Beispiel, das verdeutlichen soll, dass die Verknüpfung **not(a or b)** denselben Wert wie **(not a) and (not b)** ergibt:

a	b	not a	not b	a or b	not(a or b)	(not a) and (not b)
true	true	false	false	true	false	false
true	false	false	true	true	false	false
false	true	true	false	true	false	false
false	false	true	true	false	true	true

Gegeben sei folgender Ausdruck:

(not(a and b) and (a or b)) or ((a and b) or not(a or b))

- a) Schreiben Sie eine Wahrheitstabelle für den Ausdruck.

- b) Neben den *De Morganschen Gesetzen* gibt es noch weitere Gesetze in der booleschen Algebra. Dazu gehören unter anderem die Distributiv-, Neutralitäts- und Extremalgesetze. Zeigen Sie mit Hilfe der Gesetze, dass der Ausdruck immer den Wert **True** annimmt.

Hier finden Sie eine Sammlung der Gesetze (in Python-Notation):

```
1 # Kommutativgesetz
2 a or b == b or a
3 a and b == b and a
4
5 # Assoziativgesetz
6 a and (b and c) == (a and b) and c
7 a or (b or c) == (a or b) or c
8
9 # Distributivgesetz
10 a and (b or c) == (a and b) or (a and c)
11
12 # Idempotenzgesetz
13 a and a == a
14 a or a == a
15
16 # Neutralitätsgesetz
17 a and 1 == a
18 a or 0 == a
19
20 # Extremalgesetz
21 a and 0 == 0
22 a or 1 == 1
23
24 # Doppelnegationsgesetz
25 not (not a) == a
```