

# Prüfung **B**: Variablen, Verzweigungen, 4AB

Donnerstag, 13. Januar 2022

Zeit: 40 Minuten

max. Punktezahl: 21.5

Hilfsmittel: keine, ohne Laptop

Name: Lösungen

Total Punkte:

Note:

## Aufgabe 0 (25% der Punkte = 5.5 Punkte)

Bewertung des Mandala-Projektes:

- 0.5 P. Vorgabe: Zwei repeat
- 1 P. Vorgabe: Zwei Befehle mit Parametern
- 1 P. Vorgabe: Zwei Variablen und ein Benutzerwert
- 0.25 P. Vorgabe: Min. zwei Farben,
- 0.75 P. Umfang, Schwierigkeit, Kommentare
- 2 P. Vorgabe sauberer, modularer Code
- Abzüge (Abgabe verspätet/unvollständig; Umfang)

Benutzerwert fehlt -0.5

Vorgegebene Bild fehlt <sup>je</sup> -0.25

## Aufgabe 1 (2 Punkte)

Erkläre den Unterschied zwischen = und == in der Programmiersprache Python.

=: Damit wird einer Variable auf der linken Seite der Wert auf der rechten Seite zugewiesen. 0.75 +0.25 für weisung

==: Damit wird der Wert zweier Ausdrücke links und rechts verglichen und True bzw. False ausgegeben. 0.75 +0.25 für weisung



## Aufgabe 2 (6 Punkte)

### Original-Code:

```

1 zahl = input("Gib eine Zahl ein?")
2 if zahl > 10:           <math>10 > 10</math>
3     print("Gross.")
4     if zahl > 20:
5         print("Wirklich")
6     if zahl <= 20:
7         print("Naja")
8 if zahl < 0:
9     print("Negativ");   <math>10 < 0</math>
10 if zahl <= 10:
11     if zahl >= 0:
12         print("---")   nur für Zahlen <math>\geq 0, \leq 10</math>

```

### Dein verbesserter Code:

```

zahl = input("Gib eine Zahl ein?")
if zahl > 10:
    print("Gross.")
    if zahl > 20:
        print("Wirklich")
    else:
        print("Naja")
elif zahl < 0:
    print("Negativ")
else:
    print("---")

```

0.5  
0.5  
0.5  
Rest 0.5

- Was gibt das Programm für die Benutzereingabe 18 aus?
- Verbessere den Code. Schreibe Deine Version oben neben den Original-Code hin.
- Erkläre kurz: Weshalb ist der umgeschriebene Code besser als die Vorlage?
- Erstelle für Deine Version des Codes ein Flussdiagramm.

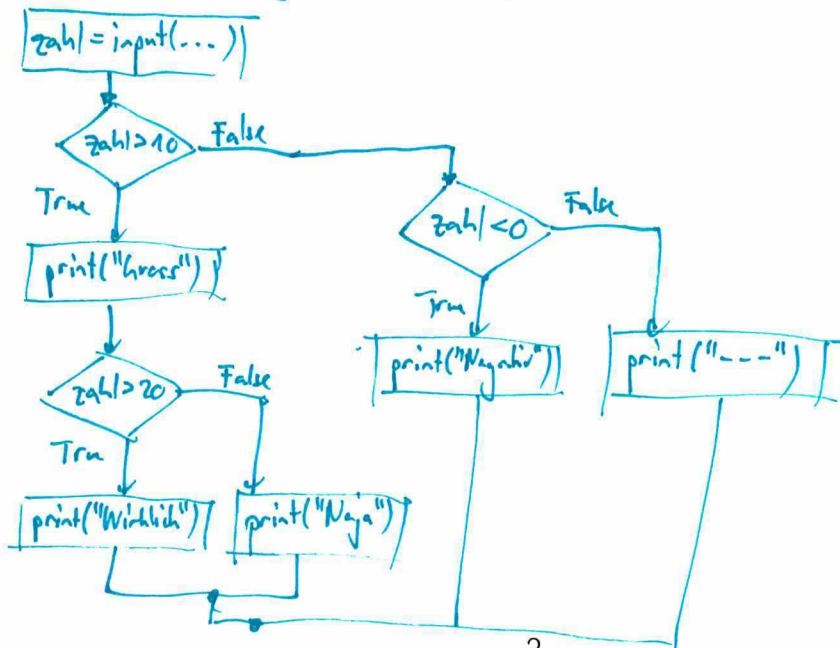
a) Gross.

Naja

Pro Fehler 0.5

c) Die Bedingungen  $>20, \leq 20$  in Zeilen 4 & 6 schließen sich gegenseitig aus, womit nur ein if nötig ist. Dies reduziert die Arbeit und erhöht die Lesbarkeit. Analog sind die Bedingungen aus Zeile 2, 8 und 10, 11 gegenseitig ergänzend.

d)



Vorzugsweise 0.5  
Zusammenführen 0.3  
Beschriftung True/False 0.2  
Anweisungen/Format 0.5



### Aufgabe 3 (6 Punkte)

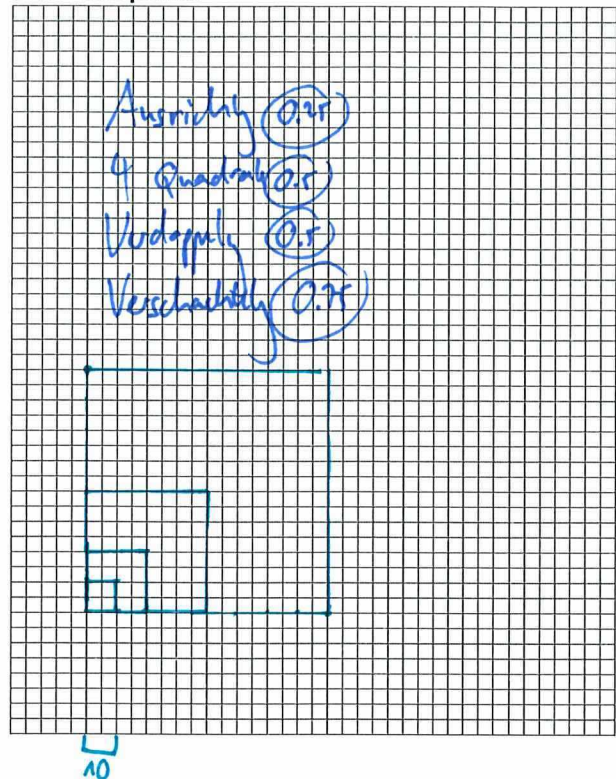
Code:

```

1 from turtle import *
2
3 def m(a):
4     repeat 4:
5         forward(a)
6         right(90)
7
8 def p(a,b,c):
9     z = a
10    repeat c:
11        m(z)
12        z *= b
13
14 makeTurtle()
15 hideTurtle()
16 p(10,2,4)

```

Die Graphik:



- ③ a) Erstelle eine Zeit-Tabelle für den Aufruf  $p(10,2,4)$ . Für  $m(a)$  muss keine Tabelle erstellt werden.
- ② b) Zeichne die Graphik, welche der Code erzeugt.
- ① c) Eine Variable  $a$  kommt im Befehl  $m(a)$  und  $p(a,b,c)$  vor. Begründe, weshalb es möglich ist, zweimal die gleiche Variable zu nutzen.

a) Zeit-Tabelle  $p(10,2,4)$

Durchlauf	1	2	3	4	Ende	
a	10	10	10	10	10	0.25
b	2	2	2	2	2	0.25
c	4	4	4	4	4	0.25
z	10	20	40	80	160	1

Struktur 0.5

c) Weil jeder Funktionsaufruf eine neue Tabelle erzeugt, was unabhängig von allen anderen Tabellen wieder eine neue Variable  $a$  geföhrt werden darf.



#### Aufgabe 4 (2 Punkte)

Code:

```

1 def b(x,y):
2     if x>y:
3         b(x-y,y)
4     elif x<y:
5         b(x,y-x)
6     else:
7         print(x)
8
9 b(21,60)

```

a) Was gibt der Code aus? Begründe Dein Resultat mit Zwischenresultaten.

b) Was berechnet  $b(x,y)$  allgemein?

a)

$b(21,60)$	:	$x < y$	$b(21,60-21)$
$b(21,39)$	:	$x < y$	$b(21,39-21)$
$b(21,18)$	:	$x > y$	$b(21-18,18)$
$b(3,18)$	:	$x < y$	$b(3,18-3)$
$b(3,15)$	:	$x < y$	$b(3,15-3)$
$b(3,12)$	:	$x < y$	$b(3,12-3)$
$b(3,9)$	:	$x < y$	$b(3,9-3)$
$b(3,6)$	:	$x < y$	$b(3,6-3)$
$b(3,3)$	:	$x == y$	$\Rightarrow \text{print}(3) \Rightarrow \boxed{3}$

Zwischenschritt ①  
1. Aufruf  $b(21,39)$  ① dann

Resultat ①

b) Der grösste gemeinsame Teiler von  $x$  und  $y$ .

①

