

# Prüfung B: Schleifen, Modularer Programmentwurf, 4AB

Donnerstag, 28. Oktober 2021

Zeit: 40 Minuten

max. Punktezahl: 19

Hilfsmittel: keine, ohne Laptop, Farbtabelle wird bei Bedarf abgegeben

Name: Lösungen

Total Punkte:

Note:

 yellow	 red	 navy	 green	 white
 gold	 magenta	 blue	 dark green	 grey
 orange	 purple	 cyan	 sienna	 black

## Aufgabe 1 (3 Punkte)

Theorie-Fragen:

- ② a) Gib die drei Divisionsoperatoren an und erkläre deren Bedeutung anhand eines Beispiels.

Normale Division  $/$ :  $5/3$  ergibt 1.666...

Ganzzahliger Teil der Division  $//$ :  $5//3$  ergibt 1

Ganzzahliger Rest der Division  $\%$ :  $5\%3$  ergibt 2

Bsp je 0.5

0.5

- ① b) Wer hat von Programmiersprachen primär einen Nutzen: Menschen oder Computer? Begründe Deine Antwort.

Primär für Menschen. Denn der Computer hat direkt Befehle, welche für uns weit weg von einer normalen Sprache sind.

Deswegen wurden Programmiersprachen erschaffen, die für uns die Kommunikation mit dem Computer vereinfachen.

Begründung 0.5

## Aufgabe 2 (3 Punkte)

Notiere, was die Codezeile ausgibt. Nutze pro Zeichen (Symbole, Leerzeichen etc.) genau ein Häuschen.

a) `1 print("Y "*3 + 3*" Y")`

→ Y Y Y Y Y Y

\* 0.5  
+ 0.5  
Abstände 0.5

b) `1 print("9**2 ist \ngleich", 9**2)`

→ 9\*\*2 ist  
gleich 81

"9\*\*2 ..." als String 0.5  
\n 0.5  
9\*\*2=81 0.5

### Aufgabe 3 (4 Punkte)

Zeichne die Graphik, welche durch folgenden Code gezeichnet wird rechts daneben in den freien Platz. Die Richtungen und Grössenverhältnis sollen dabei möglichst korrekt sein. Nutze das Lineal oder Geodreieck für die Zeichnung.

```
1 from turtle import *
2
3 makeTurtle()
4
5 setPenColor("black")
6 setPenWidth(1)
7
8 right(45)
9 repeat 4:
10     forward(30)
11     right(45)
12     forward(10)
13
14     setPenColor("red")
15     setPenWidth(3)
16
17     repeat 3:
18         forward(20)
19         left(120)
20
21     setPenColor("black")
22     setPenWidth(1)
23
24     penUp()
25     forward(20)
26     penDown()
27     forward(10)
28     left(45)
29
30 hideTurtle()
```

Farbe / Breite

(0.5)

Rechts Dreieck

(1)

(falscher Winkel 0.75)

repeat 4

(1)

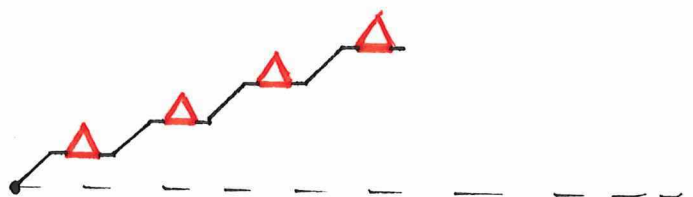
(nicht auf ganzes  
beugen → 0.5)

Winkel & Länge

(1)

Lösung

(0.5)



## Aufgabe 4 (5 Punkte)

Bearbeite folgende Aufgaben zu folgendem Code

- 2 a) Der Code beinhaltet Fehler, weshalb das Programm gar nicht korrekt läuft. Behebe diese. Den ersten findest Du, wenn Du die folgende Fehlermeldung beachtest:

```
3 def test:
4 Parameter erwartet aber '!' gefunden.
5 repeat 90:
```

3 2 Pkt  
2 1.25 Pkt  
1 0.75 Pkt

- 2 b) Ergänze die Lücken -----, sodass die nebenstehende Graphik entsteht.

Schritt (0.75) Winkel (1) Länge (0.5)

- 1 c) Gib zwei möglichst notwendige Verbesserungen an zur Art und Weise wie der Code geschrieben ist (Formatierung, Effizienz etc.). Beachte: Kommentare musst Du keine hinzufügen.

je (0.5)

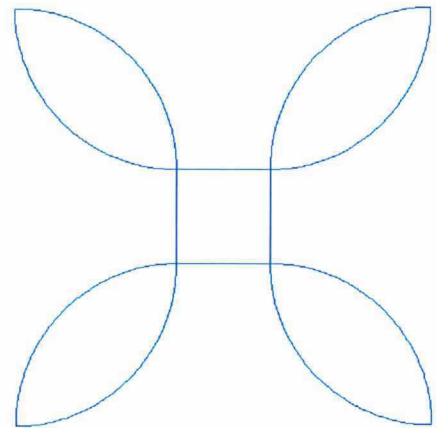
```
1 from turtle import *
2
3 def test:
4     left(45)
5     repeat 45:
6         forward(3)
7         right(2)
8         right(90)
9     repeat 45:
10        forward(3)
11        right(2)
12    right(135)
13
14 makeTurtle()
15 hideTurtle()
16 repeat 4 :
17 forward(100)
18 left(45)
19 test()
20 right(135)
```

Name: blatt():

repeat 2:  
repeat 45:  
forward(3)  
right(2)  
right(90)

oder

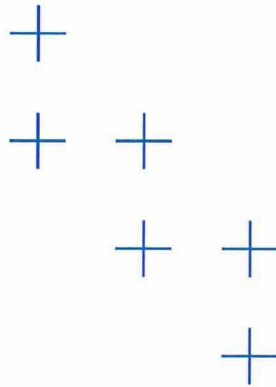
def halbblatt():  
repeat 45:  
:





### Aufgabe 5 (4 Punkte)

Notiere möglichst guten Programmcode, damit folgende Figur gezeichnet wird.



Die Breite und die Höhe der Kreuze, sowie der horizontale und vertikale Abstand zwischen den Kreuzen beträgt jeweils 20.

Beachte für diese Programmieraufgabe:

1. Benutze die Häuschen um eine saubere Formatierung zu erreichen.
2. Kommentare müssen **keine** gesetzt werden.

```
from turtle import *
```

```
def kreuz():  
    forward(20)  
    back(10)  
    left(90)  
    forward(10)  
    back(20)  
    forward(10)  
    right(90)  
    back(10)
```

```
makeTurtle(1)  
hideTurtle()  
repeat 3:  
    kreuz()  
    penUp()  
    back(40)  
    penDown()  
    kreuz()  
    penUp()  
    right(90)  
    forward(40)  
    left(90)  
    penDown()
```

Rahmen: import, makeTurtle()

(0.5)

Def kreuz (1)

repeat 3 inkl. 2x kreuz() (1)

Positionswechsel (1)

Los (0.5) (p4, p01, p2)

Mehr Platz auf der nächsten Seite.

