

# Prüfung A: 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:

- a) Was versteht man unter dem "modularen Programmentwurf" in der Programmierung? Nenne zwei Vorteile dieses Konzepts? (vgl. Seite 29)

Ein Programm aus sinnvollen Bausteinen <sup>(0.5)</sup> zusammensetzen

- eindeutige Namen für überschaubare Programmkomponenten
- übersichtlicher
- einfacher modifizieren/korrigieren
- einfacher Code wiederverwenden

Zwei Vorteile  
je (0.5)

- b) Welche Datentypen kennen wir bisher in Python? Illustriere diese mithilfe von Beispielen.

integer → Ganzzahl -123

float → Kommazahlen 1.23

string → Zeichenketten "Text123"

je (0.5)

## Aufgabe 2 (3 Punkte)

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

1.5 a)

```
1 print("HH" *3 + "\nJJ"*2)
```

→ HH HH HH  
JJ JJ

\*  
+  
ln } je 0.5

1.5 b)

```
1 print("sqrt(9) ist \ngleich", sqrt(9))
```

sqrt(9) ist  
gleich 3

sqrt(9) als str 0.5  
ln 0.5  
sqrt(9) = 3 0.5

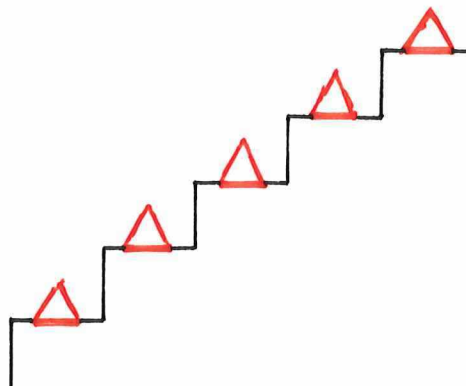
### Aufgabe 3 (4 Punkte)

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 repeat 5:
9     forward(30)
10    right(90)
11    forward(10)
12
13    setPenColor("red")
14    setPenWidth(3)
15
16    repeat 3:
17        forward(20)
18        left(120)
19
20    setPenColor("black")
21    setPenWidth(1)
22
23    penUp()
24    forward(20)
25    penDown()
26    forward(10)
27    left(90)
28
29 hideTurtle()
```

Farbe/Breite (0.5)  
Rechts Dreieck (1)  
repeat 5 (1)  
Winkel/Länge (1)  
Lösung (0.5)

(falsch Winkel 0.75)  
(nicht auf ganzes  
bezug  $\rightarrow 0.5$ )



## 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:

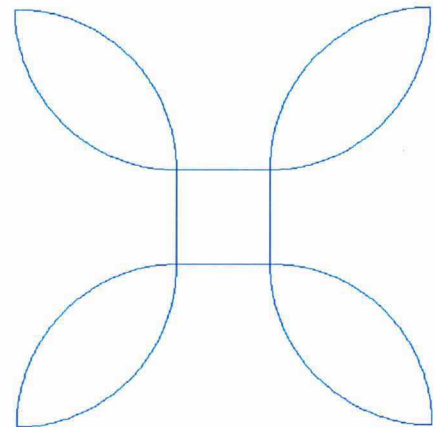
```
12 makeTurtle()
13 Der Name 'makeTurtle' ist nicht definiert oder falsch geschrieben.
14 Hast du das 'gturtle'-Modul geladen?
15 forward(100)
```

- ② b) Ergänze die Lücken -----, sodass die nebenstehende Graphik entsteht.  
Schritt (0.5), Länge (0.5), Winkel (1.25)

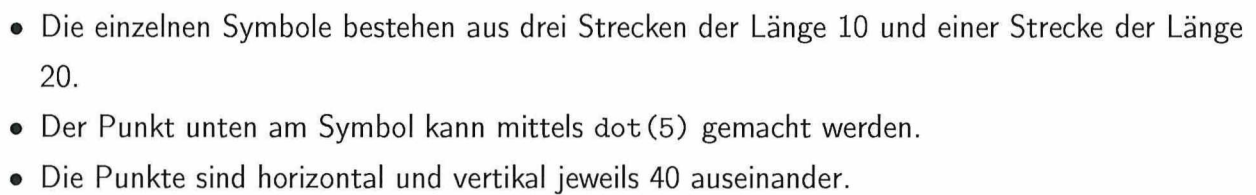
- ① c) Gib zwei möglichst notwendige Verbesserungen an zur Art und Weise wie der Code geschrieben ist (Formatierung, Effizienz etc.). Beachte: Kommentare musst Du keine hinzufügen.  
Name, Schritt zsf.

from gturtle import \*

```
1 def test():
2     left(45)
3     repeat 30:
4         forward(3)
5         right(3)
6     right(90)
7     repeat 30:
8         forward(3)
9         right(3)
10    right(135)
11
12 makeTurtle()
13 repeat 4:
14     forward(100)
15     left(45)
16     test()
17     right(135)
```



Notiere möglichst guten Programmcode, damit folgende Figur gezeichnet wird.



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

Def Flagge (1)  
repeat 3 (1)  
Positionswinkel (1)  
Lsg (0.5) (1)  $\text{pul}(), \text{pd}()$

```
make Turtle()
hide Turtle()
repeat 3:
    flagge()
    penUp()
    back(40)
    penDown()
    flagge()
    right(90)
    penUp()
    forward(40)
    penDown()
    left(90)
```

