

Selbst definierte Funktionen (1/3)

Aufgabe 1

Nimm in diesem Programm die folgenden kleinen Veränderungen vor. Beschreibe, was daraufhin beim Ausführen des Programmes geschieht, und erkläre, warum das so ist.

```
def quadrat(seite):  
    ergebnis = seite * seite  
    return ergebnis  
  
seite = input("Seitenlänge")  
flaeche = quadrat(seite)  
print flaeche
```

a) Kommentiere die Zeile `return ergebnis` aus.

```
def quadrat(seite):  
    ergebnis = seite * seite  
    # return ergebnis  
  
seite = input("Seitenlänge")  
flaeche = quadrat(seite)  
print flaeche
```

.....

.....

.....

.....

.....

.....

.....

Selbst definierte Funktionen (2/3)

- b) Ändere die letzten beiden Zeilen des Hauptprogramms wie folgt:

```
quadrat(seite)  
print ergebnis
```

```
def quadrat(seite):  
    ergebnis = seite * seite  
    return ergebnis  
  
seite = input("Seitenlänge")  
quadrat(seite)  
print ergebnis
```

.....

.....

.....

.....

.....

- c) Fasse die letzten beiden Zeilen des Hauptprogramms zusammen zu
print quadrat(seite)

```
def quadrat(seite):  
    ergebnis = seite * seite  
    return ergebnis  
  
seite = input("Seitenlänge")  
print quadrat(seite)
```

.....

.....

.....

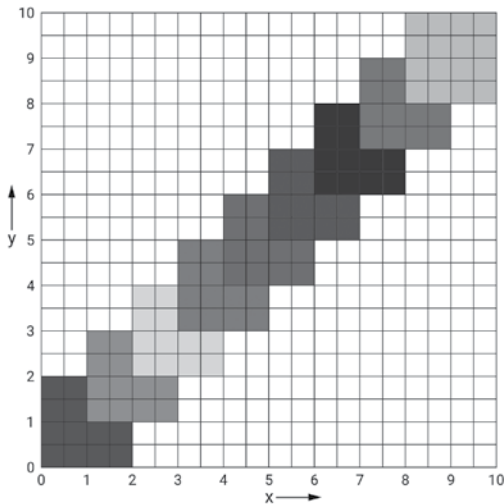
.....

.....

Selbst definierte Funktionen (3/3)

Aufgabe 2

Erstelle ein Programm, das neun in einer zufälligen RGB-Farbe gefärbte Quadrate auf diese Weise diagonal anordnet. (Das Kästchenraster dient nur der Orientierung.)



- Importiere dazu die benötigten Bibliotheken, setze den Anfangswert für den Zufallsgenerator und lass ein Grafikpanel mit den Parametern wie im Bild zeichnen.
- Definiere eine Funktion, die ein buntes Quadrat zeichnet. Die RGB-Farbe soll aus zufälligen Werten für Rot, Grün und Blau zusammengesetzt werden.
- Im Hauptprogramm soll zunächst der untere linke Eckpunkt des ersten Quadrats auf 0, 0 gesetzt werden.
- In einer Schleife sollen dann nacheinander die neun Quadrate gezeichnet werden.

Die oberen rechten Eckpunkte der Quadrate sollen jeweils ausgehend vom aktuellen linken unteren Eckpunkt definiert werden.

Nach dem Ausführen der Funktion sollen die Koordinaten des linken unteren Eckpunktes um 1 nach oben und nach rechts verändert werden.

Aufgabe 3

Erstelle ein Programm, das eingegebene Wörter auf darin enthaltene, zufällig ausgewählte Buchstaben prüft.

- Importiere dazu die benötigte Bibliothek und setze den Anfangswert für den Zufallsgenerator.
- Definiere eine Funktion, die einen zufälligen Buchstaben von a bis z auswählt und als Rückgabewert an das Hauptprogramm übergibt.
- Im Hauptprogramm soll mit Hilfe der Funktion drei Variablen nacheinander jeweils ein ausgewählter Buchstabe zugeordnet werden.
- Der Nutzer soll zur Eingabe eines Wortes aufgefordert werden. In der Aufforderung sollen die drei Buchstaben enthalten sein, beispielsweise „Wort mit abc eingeben“.
- Es soll geprüft werden, ob das eingegebene Wort alle drei Buchstaben enthält. Falls das so ist und auch falls das nicht so ist, sollen entsprechende Meldungen ausgegeben werden.

Programmierfehler (Bugs) (1/3)

Aufgabe 1

- a) Markiere alle Fehler in diesem Programm.
- b) Notiere jeweils die Fehlermeldung und die Art der Fehler.

```
from random import

from GPanel import *

seed(

makegpanel(0, 10, 0, 10)

def KreisZeichnen(r)

    rot = randint(0,255)
    grün = randint(0,255)

    blau = randint(0,255)
    setColor(rot,gruen,blau)
    fillCircle(R)

Hauptprogramm

x = 0
y == 0

for i in range1,5):

    x = x + i

    y = y + i
    r = i
    pos x,y)

    KreisZeichnen(r)
```

Programmierfehler (Bugs) (2/3)

Aufgabe 2

Im folgenden Programm sind sechs Zeilen markiert, deren Fehlen im weiteren Verlauf des Programms einen semantischen Fehler verursacht.

Beschreibe für jede dieser markierten Zeilen, was passiert, wenn sie fehlt und wo sich dieser Fehler auswirkt.

```
1 from gpanel import * ①
2 makeGPanel(-10, 10, -6, 14) ②
3 def BogenZeichnen(radius): ③
4     fillArc(radius,0,180)
5     farben = ["Red","DarkOrange","Gold",
6               "LimeGreen","RoyalBlue","BlueViolet",
7               "DeepPink","WhiteSmoke"] ④
8     for i in range(0,8):
9         farbe = farben[i] ⑤
10        setColor(farbe)
11        radius = 9 - i ⑥
12        BogenZeichnen(radius)
```

1

2

3

4

5

6

Programmierfehler (Bugs) (3/3)

Aufgabe 3

Das folgende Programm soll dieses Gesicht zeichnen.

- Markiere die logischen Fehler im Programm, die das gewünschte Ergebnis verhindern.
- Korrigiere das Programm.
- Übertrage deine Korrekturen in das Programm Aufgabe_13-3_Gesicht.py.
Liefert das Programm nun das gewünschte Ergebnis?

```
from gpanel import *
makeGPanel(-10, 10, -10, 10)

rot = 10
gruen = 150
blau = 200
setColor(blau, gruen, rot)
fillCircle(5)

pos(-2.5, 2)
pos(2.5, 2)
fillCircle(1)
fillCircle(1)
setColor("white")
fillTriangle(-0.75, -0.5, 0.75, -0.5, 0, 1)
pos(-1.5, 0)
fillArc(2.5, 0, 180)
```

Aufgabe 4

Das Programm Aufgabe_13-4_Kueken.py enthält in fast jeder Zeile einen Syntaxfehler.

Finde und korrigiere die Fehler.

Führe das Programm testweise aus.
Zeichnet es das Küken?

