Operatoren (1/2)

Aufgabe 1
Welche Ausgaben sind nach Ablauf dieser Programmzeilen im Ausgabefenster zu erwarten?

a) `print(100 + 25)` .....

b) `print(2 ** 5)` .....

c) `print(80 // 3)` .....

d) `print(25 % 4)` .....

e) `print(3 * 5 + 8)` .....

f) `print(3 ** 3 % 5)` .....

g) `print(60 / 5 - 2)` .....

h) `print(3 * 15 // 4)` .....

Aufgabe 2
Welche Ausgaben sind nach Ablauf dieser Programmzeilen im Ausgabefenster zu erwarten?

a) `print(5 in [2, 4, 6, 8])` .....

b) `print("AB" in "RHABARBER")` .....

c) `print(10 not in [10, 20, 30, 40])` .....

d) `print("T" in ["H", "O", "R", "B"])` .....

e) `print("N" not in "STUTTGART")` .....

f) `print(1 in [-2, -1, 0, 1, 2])` .....

g) `print(10**2 not in [500, 1000, 1500])` .....

h) `print(10 % 3 in [3, 6, 9])` .....

Operatoren (2/2)

Aufgabe 3
Welche Ausgaben sind nach Ablauf dieser Programmzeilen im Ausgabefenster zu erwarten?

a) `print(7>5 and 14>7)` .....

b) `print(3<8 or 3>8)` .....

c) `print(not 10<12)` .....

d) `print(20>10 and not 10>5)` .....

Aufgabe 4
Welche Ausgabe ist nach Ablauf dieses Programms im Ausgabefenster zu erwarten?

```
berg = "Feldberg"
hoehe = 1277
einheit = "Meter"
gebirge = "Schwarzwald"

print("Der",berg,"im",gebirge)
print("ist"+str(hoehe)+einheit+"hoch.")
```

.....

.....

Aufgabe 5
Schreibe ein Programm, das den Namen und den Wohnort abfragt und beides in der Form „X wohnt in Y“ wieder ausgibt.

Aufgabe 6
Schreibe ein Programm, das

- fünf Variablen mit den folgenden Werten anlegt und
- sie mit dem +-Operator verkettet und ausgibt

Texte:	Im Jahr	Zahlen:	2020
	leben		7.8
	Milliarden Menschen auf der Erde		

for-Schleifen (1/1)

Aufgabe 1
Welche Ausgaben sind nach Ablauf dieser Programmzeilen im Ausgabefenster zu erwarten?

a) `print(list(range(1,10,2)))` .....

b) `print(list(range(-2,-10,-4)))` .....

c) `print(list(range(3)))` .....

d) `print(list(range(15,20)))` .....

e) `print(list(range(-5,0)))` .....

f) `print(list(range(12,4,-2)))` .....

g) `print(list(range(10,90,15)))` .....

Aufgabe 2
Schreibe ein Programm, mit dessen Hilfe die Lottozahlen 6 aus 49 ermittelt und ausgegeben werden können.

Aufgabe 3
Lass den Computer im übertragenen Sinne 6000 mal würfeln und anschließend ausgeben, wie häufig die einzelnen Augenzahlen gefallen sind.

- Weise dazu sechs Variablen den Wert 0 zu.
- Verwende eine for-Schleife, um die Anzahl der gewürfelten Zahlen zu zählen.
- Erhöhe nach jedem Würfeln den Wert der Variable, die zur jeweiligen Augenzahl gehört.
- Gib nach dem Durchlaufen der Schleife aus, wie häufig die einzelnen Ziffern gefallen sind

Aufgabe 4
Schreibe Programme, die die Zeilen- und Spaltenzahl abfragen und anschließend – mit zwei for-Schleifen – die folgenden Muster ausgeben.

In jeder Runde soll dabei nur ein Zeichen angefügt werden.

Die Programme sollen beliebige Zeilen- und Spaltenanzahlen erlauben und entsprechend größere oder kleinere Figuren erzeugen.

a)
```
OOOOO
OOOOO
OOOOO
OOOOO
OOOOO
```

b)
```
O----
-O---
--O--
---O-
----O
```

c)
```
OOOOO
-OOOO
--OOO
---OO
----O
```

d)
```
OOOOO
O---O
O---O
O---O
OOOOO
```

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))
```

.....

.....

.....

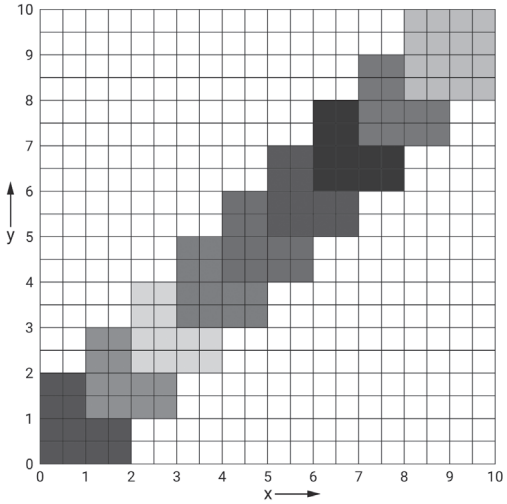
.....

.....

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.)



a) Importiere dazu die benötigten Bibliotheken, setze den Anfangswert für den Zufallsgenerator und lass ein Grafikpanel mit den Parametern wie im Bild zeichnen.

b) Definiere eine Funktion, die ein buntes Quadrat zeichnet. Die RGB-Farbe soll aus zufälligen Werten für Rot, Grün und Blau zusammengesetzt werden.

c) Im Hauptprogramm soll zunächst der untere linke Eckpunkt des ersten Quadrats auf 0, 0 gesetzt werden.

d) 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.

a) Importiere dazu die benötigte Bibliothek und setze den Anfangswert für den Zufallsgenerator.

b) Definiere eine Funktion, die einen zufälligen Buchstaben von a bis z auswählt und als Rückgabewert an das Hauptprogramm übergibt.

c) Im Hauptprogramm soll mit Hilfe der Funktion drei Variablen nacheinander jeweils ein ausgewählter Buchstabe zugeordnet werden.

d) Der Nutzer soll zur Eingabe eines Wortes aufgefordert werden. In der Aufforderung sollen die drei Buchstaben enthalten sein, beispielsweise „Wort mit abc eingeben“.

e) 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",
   "LimeGreen","RoyalBlue","BlueViolet",
   "DeepPink","WhiteSmoke"] 4
6  for i in range(0,8):
7      farbe = farben[i] 5
8      setColor(farbe)
9      radius = 9 - i 6
10     BogenZeichnen(radius)
```

1 .....

.....

.....

.....

.....

.....

2 .....

.....

3 .....

.....

4 .....

.....

5 .....

.....

6 .....

.....

Programmierfehler (Bugs) (3/3)

Aufgabe 3

Das folgende Programm soll dieses Gesicht zeichnen.

a) Markiere die logischen Fehler im Programm, die das gewünschte Ergebnis verhindern.

b) Korrigiere das Programm.

c) 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?