

### Aufgabe 1: Zahlendarstellung

- (a) (9 Punkte) Vervollständigen sie die folgende Tabelle so, dass in den Zeilen jeweils die gleichen Werte stehen.

Binär in Zweierkomplement (16Bit)	Oktal	Dezimal
0110011010100110		
1011111100000000		
	100703	
	4007	
		-42
		500

- (b) ( 2 Punkte) Welche Dezimalzahl wird durch das Bitmuster  
1 10000001 1001000000000000000000  
Gemäß IEEE 754 codiert?

- (c) (2 Punkte) Geben sie das Bitmuster und den Rechenweg für die Zahl  
-22,5 gemäß IEEE-754 ( 32 Bit) an!

## Aufgabe 2: Python-Ausdrücke

(a) (5 Punkte) Welche Ausgabe erzeugen folgende Ausdrücke ( Python 3.4) ?

Ausdruck	Ausgabe
<code>print ( 2 &lt;&lt; 1 / 5 + 7 )</code>	
<code>print ( 3 / 2 * 2 == 3 / 2. * 2 )</code>	
<code>print ( 2 ** 3 % 4 * 2.0</code>	
<code>print (17 &lt;&lt; 1 + 4 // 2 % 3 )</code>	
<code>print ( int ( 5.0 % 3 * ( 8 &gt;&gt; 3 ) + 0.5 )</code>	
<code>print (2 &lt;&lt; 3 &gt;&gt; 1 % 4</code>	
<code>print ( i :=4 , 2.3 )</code>	
<code>print ( 2 == 3 == 3)</code>	
<code>print ( 1 . 25 . as_integer_ratio())</code>	
<code>print ( ' hallo ' [ 2 :: -1 ]</code>	

(b) (2 Punkte) Vervollständigen Sie die `print`-Anweisung, des nachfolgenden Programms durch „slicing“ so, dass es den Text

E esre  
ausgibt.

**Programm:**

```
1 s = 'Reverse Engineering'  
2 print ( )
```

(c) ( 3 Punkte) Was ist die Ausgabe des folgenden Programms?

```
1 def dummystuff ( ) :  
2     global a  
3     anzahl = 0  
4     for i in range (1, len(a)):  
5         if anzahl < 8 and a[i] %2 == 0:  
6             print (a[i])  
7             anzahl += 1  
8 a = (4, 7, 6, 3, 8, 1, 2, 5, 9, 14, 4)  
9 dummystuff ( )
```

### Aufgabe 3: Objektorientierung

- (a) (1 Punkt) Was ist der Unterschied zwischen einem Klassenattribut und einem (Objekt-) Attribut? (1-3 Sätze)
- (b) (2 Punkte) Was besagt das Geheimnisprinzip der Objektorientierten Programmierung? Geben sie ein Beispiel wann es wichtig ist. ( 2-3Sätze)
- (c) ( 3Punkte) Zeichnen sie ein UML-Klassendiagramm entsprechend der nachfolgenden Beschreibung. Für die Darstellung der Klassen genügt die Angabe des Namens in der üblichen Darstellung. Attribute und Operationen sind in dieser Aufgabe **nicht** vorgesehen. Der Fokus liegt auf der Modellierung der Assoziationen. Geben sie dabei immer die Multiplizitäten an, auch wenn es sich um Standardwerte handeln sollte. Die Leserichtung der Assoziationen muss nicht angegeben werden. Achten Sie jedoch darauf, je nach Typ der Assoziation die richtigen Symbole zu verwenden.

*Ein Besitzer (**Owner**) kann beliebig viele Sammlungen (**Collection**) besitzen. Jede Sammlung gehört immer genau einem Besitzer. Eine Sammlung setzt sich aus Medien (**Medium**) zusammen, mindestens jedoch aus einem. Ein Medium ist abstrakt definiert und wird durch **CD** bzw. **LP** realisiert, bei denen es sich um Spezialisierungen von Medium handelt. Ein Medium setzt sich aus Tracks (**Track**) zusammen. Tracks können nicht allein für sich existieren. Ein Medium besteht aus mindestens einem, maximal aber aus 30 Tracks. Jeder Track entspricht letztlich genau einem Lied (**Song**).*

#### Aufgabe 4: Datenstrukturen

- (a) (2 Punkte) Betrachten Sie den in Abbildung 1 gezeigten Binärbaum. Wie viele innere Knoten und wie viele Blätter besitzt dieser Baum? Welche Knoten befinden sich auf der Stufe 2? Geben sie die Abfolge der Knoten gemäß der Preorder-Traversierung des Baumes an.

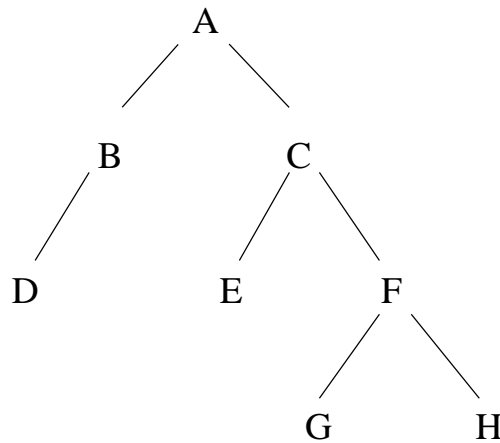


Abbildung 1: Binärbaum

#### Antwort

Anzahl innerer Knoten	
Anzahl Blätter	
<b>Namen</b> der Knoten auf Stufe 2	
Abfolge Preorder-Traversierung	

- (b) (2 Punkte) Erläutern Sie die grundlegende Funktion sowie die Gemeinsamkeiten und Unterschiede eines Stapelspeichers und einer Warteschlange ( 3-6 Sätze).

## Aufgabe 5: Programmierung in der Anwendung

- (a) (5 Punkte) Gegeben sei ein Grauwert-Bild image ( 30 mal 30 Pixel ) in einem 2-dimensionalen Python-Array. Darauf soll ein Filter

$$k = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

angewendet werden, welcher ebenfalls als 2D-Array vorliegt. Der neu berechnete Pixelwert muss dann noch mit der Filtergröße normalisiert werden. Nehmen Sie für Bereiche außerhalb des Bildes an, dass diese eine Intensität von 0 haben. Nach dem Filtern soll das Bild in das 2D-Array `filtered` geschrieben werden.

Vervollständigen sie das gegebene Python Programm an den durch Unterstriche gekennzeichneten Stellen. Füllen sie dazu die gegebene Tabelle aus.

```
1 image = [[...], [...], ..., [...]] # 30 times 30 pixels
2 filtered = []
3 k = [[0, 1, 0], [1, -4, 1], [0, 1, 0]]
4
5 for x in ____:
6     for y in ____:
7
8         new_value = ____
9         for i in range(-1,1):
10            for j in range(-1,1):
11
12                if x+i < 0:
13                    new_value += ____
14                elif x+i > 29:
15                    new_value += ____
16                elif y+j < 0:
17                    new_value += ____
18                elif y+j > 29:
19                    new_value += ____
20                else:
21                    new_value += ____
22
23 filtered[x][y] = ____
24
```

<b>Zeile</b>	<b>Code</b>
Zeile 5	
Zeile 6	
Zeile 8	
Zeile 13	
Zeile 15	
Zeile 17	
Zeile 19	
Zeile 21	
Zeile 23	

- (b) (3 Punkte) Nennen und Beschreiben sie zwei Modelle für Software Lebenszyklen. Gehen sie auf Vor- und Nachteile ein, je eine Nennung ist ausreichend.

## Aufgabe 6: Synchronisation

- (a) (5 Punkte) Was ist ein **korrespondierender** kritischer Abschnitt im Unterschied zu einem einfachen kritischen Abschnitt? Geben sie beide Definitionen und inhaltliche Unterschiede an.
- (b) (5 Punkte) Betrachten sie die folgenden fünf Threads, die gleichzeitig ausgeführt werden. Jeder Thread besteht aus vier Abschnitten, wobei Abschnitte der Typen A bis G jeweils **korrespondierend** kritisch sind:

T1	T2	T3	T4	T5
D	D	B	G	A
B	E	E	H	D
C	A	E	G	E
D	D	F	A	H

Welche Abschnitte müssen sie tatsächlich mit Semaphoren synchronisieren?  
Wie viele Semaphore benötigen Sie zur korrekten Synchronisation aller Threads insgesamt? Begründen sie Ihre Antworten.

## Aufgabe 7: Programmanalyse

- (a) (10 Punkte) Ein etwas unerfahrener Programmierer hat untenstehendes Python 3.4 Programm geschrieben. Leider sind sehr viele Fehler in diesem Programm, insgesamt 13. Wenn sie 10 Fehler gefunden haben, erhalten sie die volle Punktzahl (weitere ergeben Bonuspunkte!). Finden sie möglichst alle Fehler, klassifizieren sie diese ( durch ein Kreuz in der richtigen Spalte ) in der umseitig gegebenen Tabelle. Beschreiben und korrigieren Sie die Fehler in der jeweiligen Spalte. Beziehen sie sich auf die Nummer der Zeile, welche in der Spalte mit Titel # gegeben ist. ( Style: Ein Fehler, der unseren Programmierrichtlinien (Style Guide) widerspricht.)

```
1
2
3 import tkinter for tk
4
5
6 name_list = {}
7
8 def names():
9     "Read in Names and print them"
10
11     last_name = e1.get()
12     first_name = e2.get()
13     name_list.append((len(name_list) + 1, last_name, first_name))
14     print name_list
15
16 main = tk.tk()
17
18
19 tk.Label(main, text = "Vorname").grid(row = 0, column = 0)
20 tk.Label(main, text = "Nachname").grid(row = 1, column = 0)
21
22
23 e1 = tk.Entry( main )
24 e2 = tk.Entry(main)
25 button = tk.Button(main, text = "Input", command = names())
26
27
28 e1.grid(row = 0, column = 1)
29 e2.grid(row = 1, column = 1)
30 button.grid(row = 1, column = 1)
31
32
33
34 print ("Die Anwendung 'Namenstabelle' wurde geschlossen.")
35
```



#	Style	Syntax	Logik	Beschreibung	Korrektur
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					

(b)(4 Punkte) Erklären sie, was der Programmierer erreichen wollte. Was gibt das korrigierte Programm aus, wenn sie als ersten Namen „Mickey Mouse“ und als zweiten Ihren eigenen Vornamen und Nachnamen eingeben? Geben sie die Ausgabe nach Eingabe des zweiten Vornamens und Nachnamens an. Achten sie auf die Syntax der Ausgabe!