

PROBEKLAUSUR ZU PROGRAMMIEREN 1 IM WINTERSEMESTER 2011/12



Geschrieben am ---.---.----

Vorname _____
Nachname _____
Matrikelnummer _____
Geburtsdatum _____
Studiengang _____

Klausur für 9 Creditpoints werten? ☐

Vom Prüfer auszufüllen:

Aufgabe	1	2	3	4	5	6	7	8	9	10	Σ
Punkte	10	10	10	10	10	10	10	10	10	10	100
Davon erreicht											

Klausurpunkte _____
+ Bonuspunkte _____
 Σ _____
Note _____

Generelle Klausurhinweise:

1. Geben Sie auf jedem Blatt (oben rechts) ihre Matrikelnummer an. Blätter ohne Matrikelnummer können nicht gewertet werden.
2. Schreiben Sie bitte *leserlich*!
3. Kontrollieren Sie ihre Klausur auf Vollständigkeit. Die Seitenzahlen befinden sich unten rechts.
4. Verwenden Sie die Rückseiten der Klausur ausschließlich für eigene Notizen - diese werden **nicht** gewertet. Die letzte Seite der Klausur ist als "*Schmierpapier*" vorgesehen, falls der Platz zum Beantworten einer Frage nicht ausreicht (Verweis bei der Aufgabenstellung und deutliche Kennzeichnung auf dem Schmierblatt). Benötigen Sie weiteres Papier, melden Sie sich bei der Aufsicht. Selbst mit gebrachtes Papier wird als Täuschungsversuch gewertet!
5. Außer einem dokumentenechten Stift - kein Bleistift - (**nicht** Rot) sind keine weiteren Hilfsmittel zugelassen, wie Handy, Taschenrechner, Laptop etc. (ein betriebsbereites Handy wird als Täuschungsversuch gewertet).
6. Die Prüflinge können während der Klausur einzeln die Toilette besuchen. Vor Verlassen des Klausorraumes haben diese bei der Aufsicht ihren Namen anzugeben.
7. Für die Bearbeitung der Klausur stehen 120 Minuten zur Verfügung. Eine halbe Stunde (30 Minuten) vor Abgabe ist es den Prüflingen untersagt den Raum zu verlassen, um unnötige Unruhe zu vermeiden.

Aufgabe 1: Multiple Choice**Punkte: __ / 10****(a) (2 Punkte) Pythonanalyse**

Was berechnet folgende Funktion?

```
1 def unknown(a):  
2     for i in range(2, int(a**0.5) + 1):  
3         if a % i == 0:  
4             return i  
5     return None
```

- ☐ Die Wurzel von a für eine Zahl ≥ 2
- ☐ Das Quadrat von a für eine positiven Zahlen
- ☐ Einen ganzzahligen Teiler von a
- ☐ Den größten ganzzahligen Teiler von a

(b) (2 Punkte) Python AusdrückeWelche der folgenden Ausdrücke werden von Python (Version 3.x) zu **True** ausgewertet?

- ☐ $5//2 * 2 == 5.0$
- ☐ `not 4%2 == 1`
- ☐ `bool(4<<1>>3)`
- ☐ keine der obigen

(c) (2 Punkte) Funktionen & Prozeduren

Funktionen unterscheiden sich von Prozeduren durch

- ☐ ihre Signatur.
- ☐ das vorhanden sein eines Rückgabewertes.
- ☐ formale Parameter.
- ☐ reale Parameter.

(d) (2 Punkte) Abstrakte Datentypen

Welche Aussagen treffen auf Abstrakte Datentypen zu?

- ☐ Stellen Schnittstellen bereit
- ☐ Sind mit Datenstrukturen gleichzusetzen
- ☐ Besitzen einen abgedeckten Wertebereich
- ☐ Reduzieren ggf. die Mächtigkeit einer Datenstruktur


(e) (2 Punkte) Synchronisationsmechanismen

Mit Semaphoren kann

- ☐ ein wechselseitiger Ausschluss erzielt werden.
- ☐ eine atomare Aktion sichergestellt werden.
- ☐ eine Aktion abgebrochen werden.
- ☐ ein Betriebsmittel geschützt werden.

Aufgabe 2: Zahlendarstellung**Punkte:** __ / 10

- (a) (2 Punkte) Geben Sie das Bitmuster und den Rechenweg der Zahl $z = -21,5$ gemäß *IEEE-754* mit einfacher Genauigkeit an.



Aufgabe 3: Python-Datentypen**Punkte: __ / 10**

- (a) (2 Punkte) Rekonstruieren Sie den Graphen G anhand der Adjazenzliste A .

```
A = [ ['A', ['B', 'D', 'E']],  
      ['B', ['C', 'E']],  
      ['C', ['A']],  
      ['D', ['A']],  
      ['E', []] ]
```



(b) (4 Punkte) Gegeben seien die folgenden Python Codezeilen:

```
1 a = 17 + 4 ** 3 // 2
2 b = 17 << 1 + 4 // 3
```

Geben Sie die Auswertungsreihenfolge der Operatoren von links (zuerst) nach rechts (zuletzt) an.

Priorität

1 2 3

Auswertungsreihenfolge der Operatoren aus Zeile 1

--	--	--

Auswertungsreihenfolge der Operatoren aus Zeile 2

--	--	--

(c) (2 Punkte) Von welchem `type` sind die Variablen `a` und `b`?

(d) (2 Punkte) Vervollständigen Sie die folgenden Aussagen:

1. Wie viele Zustände lassen sich mit n-Bit kodieren?

1. _____

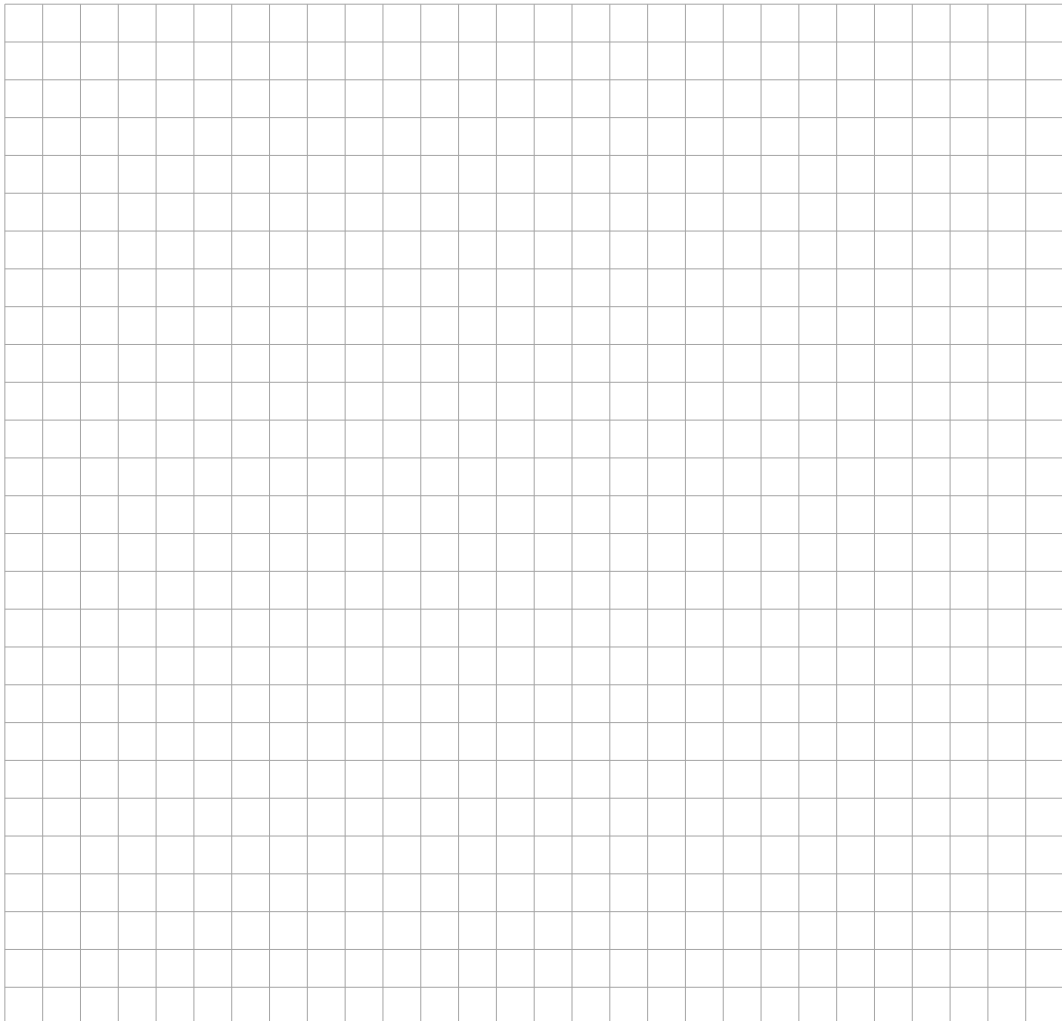
2. Wie werden *veränderliche* Datentypen in Python genannt?

2. _____

Aufgabe 5: OOP & OOAD**Punkte: __ / 10**

- (a) (6 Punkte) Erstellen Sie aus dem gegebenen Python-Code das zugehörige UML-Klassendiagramm.

```
1 class Compress(object):
2     def __init__(self): self.__compression = 0.0
3     def encode(self, string=''): return ''
4     def decode(self, string=''): return ''
5     def verbose(self): return ''
6
7 class Huffman(Compress):
8     def __init__(self):
9         Compress.__init__(self)
10        self.frequency = {}
11        self.character_count = 0
12    def H(self, char=''): return 0.0
13    def I(self, char=''): return 0.0
14    def encode_text(self, text=''): return ''
15    def compute_entropy(self, length=0): return 0.0
16    def count_frequency(self, char=''): return 0
17
18 class RunLengthEncoding(Compress):
19     def write_file(self, path=''): return None
```



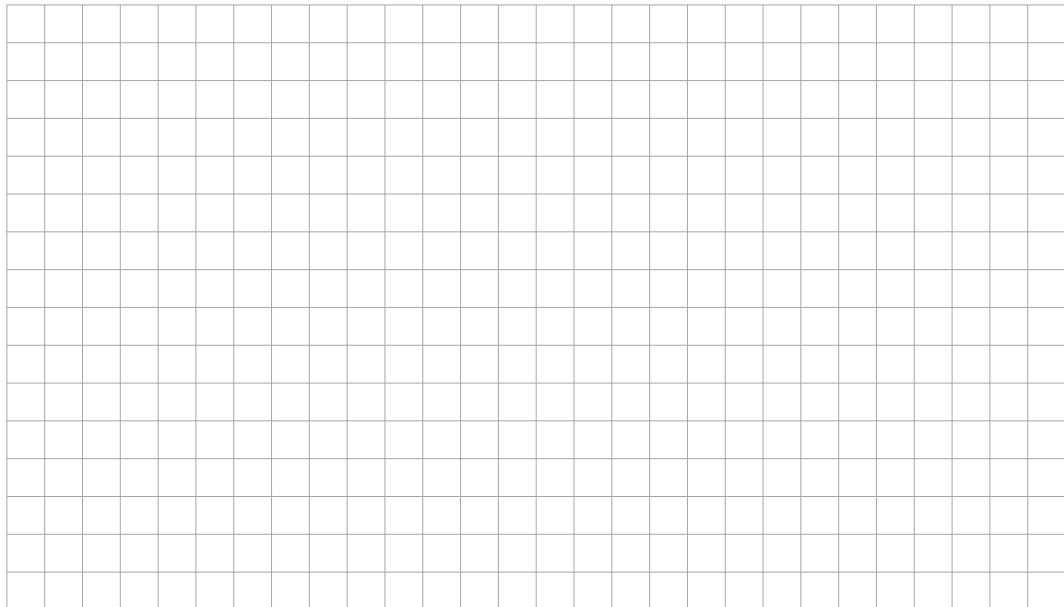
(b) (2 Punkte) Erläutern Sie den Begriff *Introspektion*.

(c) (2 Punkte) Erläutern Sie den Begriff des *Ableiten* im Kontext des Objektorientierung.

Aufgabe 6: Datenstrukturen**Punkte:** __ / 10Gegeben sei die folgende Adjazenzmatrix \mathcal{A} des Baumes \mathcal{B} .

	A	B	C	D	E	F	G	H	I	J
A	0	1	1	0	0	0	0	0	0	0
B	1	0	0	1	1	0	0	0	0	0
C	1	0	0	0	0	1	0	0	0	0
D	0	1	0	0	0	0	1	1	0	0
E	0	1	0	0	0	0	0	0	0	0
F	0	0	1	0	0	0	0	0	1	1
G	0	0	0	1	0	0	0	0	0	0
H	0	0	0	1	0	0	0	0	0	0
I	0	0	0	0	0	1	0	0	0	0
J	0	0	0	0	0	1	0	0	0	0

- (a) (4 Punkte) Zeichnen Sie den durch \mathcal{A} beschriebenen Baum \mathcal{B} mit A als Wurzel. Verwenden Sie den **linken** Pfad für Knoten die nur einen Nachfahren besitzen.



- (b) (6 Punkte) Traversieren Sie den Baum und nennen Sie die Reihenfolge, in der die Knoten betrachtet werden für:

Preorder:

Inorder:

Postorder:

Aufgabe 7: Daten – Information – Wissen

Punkte: __ / 10

Gegeben sei die folgende Nachricht:

TRUG TIM EINE SO ALTE HOSE GERN?

Gehen Sie im Folgenden davon aus, dass alle Zeichen des zugrundeliegenden Alphabets in der Nachricht vorkommen und dass die Nachricht repräsentativ für die Auftrittswahrscheinlichkeit eines jeden Zeichens ist.

- (a) (4 Punkte) Wie definierte Shannon die erwartete Information (*Entropie* H) einer Nachrichtenquelle (*Informationsquelle*) X über einem Zeichenvorrat Ω ?

- (b) (4 Punkte) Wie groß ist der Informationsgehalt I für die Zeichen A und I? Geben Sie auch den Rechenweg an.

[illegible]

- (c) (2 Punkte) Mit was kann die Entropie eines Zeichens einer Informationsquelle gleichgesetzt werden?

Aufgabe 8: Debuggen**Punkte: __ / 10**

(a) (6 Punkte) Finden Sie in folgendem Code alle Fehler.

```
1 def add_queen(new_row, col, old_solution)
2     new_solution = ()
3     foreach solution in old_solutions:
4         for new_col in range(0:len(col):-1):
5             if self.attacks(new_row, new_col, solution);
6                 new_solution.append(Solution % [new_col])
7     return new_solution
```

Fehler in Zeile 1:

Fehler in Zeile 2:

Fehler in Zeile 3:

Fehler in Zeile 4:

Fehler in Zeile 5:

Fehler in Zeile 6:

Fehler in Zeile 7:

(b) (4 Punkte) Gegeben Sei die folgende *Python*-Funktion

```
1 def f(x, y, z):    # S
2     r = None
3     if y > z:       # B1
4         r = y       # A1
5         if 5 < z:    # B2
6             r = x    # A2
7     if x <= y:       # B3
8         r = y        # A3
9     return r        # E
```

und die Testmenge \mathcal{T} , bei der jeder Testfall die Form (x, y, z) besitzt:

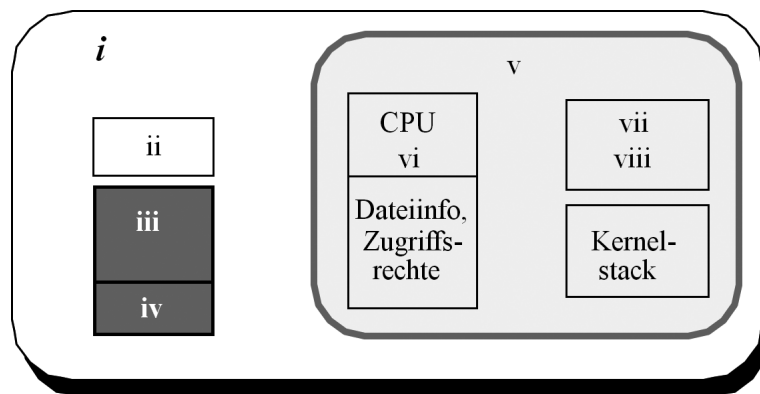
$$\mathcal{T} := \{(6, 5, 5), (0, 1, 1), (6, 6, 5), (8, 7, 6)\}$$

Erweitern Sie die gegebene Testmenge \mathcal{T} um nötige Testfälle, so dass eine *minimale Testmenge* zur Ablaufpfadüberdeckung entsteht.

Aufgabe 9: Prozesse & Synchronisation**Punkte: __ / 10**

- (a) (2 Punkte) Erklären Sie in eigenen Worten was der „wechselseitiger Ausschluss“ in der Informatik bewirkt?

- (b) (8 Punkte) Benennen Sie die gekennzeichneten Elemente (*i-viii*) der Zeichnung?



- i* _____
- ii* _____
- iii* _____
- iv* _____
- v* _____
- vi* _____
- vii* _____
- viii* _____

Aufgabe 10: Algorithmenentwurf**Punkte: __ / 10**

- (a) (10 Punkte) Erläutern Sie die systematische Vorgehensweise von *Divide and Conquer*-, *Greedy*- und *Backtracking*-Algorithmen und gehen Sie auf deren Vor- und Nachteil ein:

Divide and Conquer

Greedy

Backtracking

