

ABSCHLUSSKLAUSUR ZU PROGRAMMIERUNG 1 IM SOMMERSEMESTER 2013



Geschrieben am 06.09.2013

Nachname _____
Matrikelnummer _____
Geburtsdatum _____
Studiengang _____

Klausur für 9 Credit Points werten? Ja ☐

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, oder falls der Platz zum Beantworten einer Frage nicht ausreicht. Zur Benotung muss ein Verweis bei der Aufgabenstellung und eine deutliche Kennzeichnung auf dem Schmierblatt enthalten sein. 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, Smartphone, Taschenrechner, Laptop etc. Ein betriebsbereites Handy oder Smartphone 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 180 Minuten zur Verfügung. In der letzten halben 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

Eine Frage gilt dann als korrekt beantwortet, wenn alle richtigen und keine falschen Antworten angekreuzt sind. Mindestens eine Antwort ist immer richtig.

(a) (1 Punkt) **Algorithmus**Welche Eigenschaften besitzt jeder *deterministische* Algorithmus?

- ☐ Er ist Determiniert
- ☐ Er kann beliebig viel Speicher anfordern
- ☐ Zu jedem Ausführungszeitpunkt ist der nächste Schritt eindeutig
- ☐ Er bricht nach einer endlichen Zeit ab

(b) (1 Punkt) **Formale Sprachen**

Welche der folgenden Elemente sind Bestandteil einer formalen Grammatik?

- ☐ Terminalregeln
- ☐ Produktionsregeln
- ☐ Startsymbol
- ☐ Endsymbole

(c) (1 Punkt) **Rechner-Architektur**Welcher dieser Befehle führt zum „*Interpretieren der Befehle im Steuerwerk*“?

- ☐ DECODE
- ☐ FETCH OPERANDS
- ☐ EXECUTE
- ☐ UPDATE INSTRUCTION POINTER

(d) (1 Punkt) **Pythonanalyse**Was berechnet die Python Funktion `unknown(n)`?

```
1 def unknown(n):  
2     if n > 1:  
3         return n + unknown(n - 1)  
4     return 1
```

- ☐ Die Fakultät von n
- ☐ Die n -te Fibonacci Zahl
- ☐ Die Summe der Zahlen 1 bis n
- ☐ Die Produktsumme der Zahlen 1 bis n

(e) (1 Punkt) **Python Ausdrücke**Welche der folgenden Ausdrücke werden von Python (Version 3.x) zu **True** ausgewertet?

- ☐ `bool(12 % 6.0)`
- ☐ `bool(1 << 2 << 3)`
- ☐ `bool(False and True or not False)`
- ☐ `bool(not True and 12 * 12 / 12)`

(f) (1 Punkt) **Build-In Datentypen**Welche Eigenschaften haben Variablen vom Typ *Integer* in Python?

- ☐ Es können maximal 2^{32} unterschiedliche Werte abgebildet werden
- ☐ Es können maximal 2^{64} unterschiedliche Werte abgebildet werden
- ☐ Sie sind *immutable*
- ☐ Für die Variablen `i=1` und `j=1` gilt `id(i)==id(j)` ist `True`

(g) (1 Punkt) **Objektorientierung**Wie kann in Python aus der Klasse B auf die Methode `m` einer Basisklasse A zugegriffen werden?

- ☐ `super(A, self).m()`
- ☐ `super(B, self).m()`
- ☐ `super().m()`
- ☐ `A.m(self)`

(h) (1 Punkt) **Prozesse**

Woraus besteht der Prozesskontext unter Anderem?

- ☐ Stack
- ☐ Kernelstack
- ☐ Programmdateien
- ☐ Zugriffsrechte

(i) (1 Punkt) **Synchronisation**Welche notwendigen, beziehungsweise hinreichenden, Bedingungen müssen nach *Coffman et al.* für eine Verklemmung (*deadlock*) gegeben sein?

- ☐ busy wait
- ☐ circular wait
- ☐ hold and wait
- ☐ preemptive wait

(j) (1 Punkt) **Algorithmenentwurf**Wodurch hebt sich das *Backtracking*-Verfahren von einem *Greedy*-Algorithmus ab?

- ☐ Aheadpropagation
- ☐ Backpropagation
- ☐ Lookback
- ☐ Loopback

Aufgabe 3: Elementare Datentypen

Punkte: ____ / 10

- (a) (2 Punkte) Gegeben sind die folgenden Python Codezeilen:

```

1 a = 2 * 3 // 42 - 2
2 b = 2 % 42 + 15 / 3
3 c = 4 ^ 4 << 2 >> 4
4 d = 4 <= 1 - 2 & 3

```

Tragen Sie die Operatoren in der Reihenfolge in die Kästchen ein, in der Sie ausgewertet werden. Eine Zeile gilt als richtig gelöst, wenn alle drei Operatoren richtig eingetragen sind.

Der Operator wird als 1. 2. 3. ausgewertet.

Zeile 1	<input type="text"/>	<input type="text"/>	<input type="text"/>
Zeile 2	<input type="text"/>	<input type="text"/>	<input type="text"/>
Zeile 3	<input type="text"/>	<input type="text"/>	<input type="text"/>
Zeile 4	<input type="text"/>	<input type="text"/>	<input type="text"/>

- (b) (2 Punkte) Erklären Sie in eigenen Worten die Begriffe *dynamische Typisierung* und *statische Typisierung*.

- (c) (2 Punkte) Erklären Sie in eigenen Worten die Begriffe *starke Typisierung* und *schwache Typisierung*.

(d) (1 Punkt) Was verstehen wir in der Informatik unter dem Begriff „*Datentyp*“?

(e) (1 Punkt) Worin besteht der Unterschied zwischen *Casting* und *Coertion*?

(f) (2 Punkte) Mit welchen Befehlen können in Python (3.x) die folgenden Typkonvertierungen vorgenommen werden?

1. Den Wahrheitswert `x>y` als String

1. _____

2. Der String `s="0xBAADF00D"` als Ganzzahl

2. _____

3. Die Oktalzahl `0o177` als Dezimalzahl

3. _____

4. Den Buchstaben `c = "X"` als ganzzahligen Wert gemäß der ASCII Tabelle

4. _____

Aufgabe 5: Objektorientierung

Punkte: ____ / 10

- (a) (1 Punkt) Worin besteht der wesentliche Unterschied zwischen der objektorientierten und der prozeduralen Programmierung?

- (b) (1 Punkt) Was besagt das Geheimnisprinzip?

- (c) (4 Punkte) Welche Arten der Sichtbarkeit (Zugriffskontrolle) gibt es in der Objektorientierung?

Welche Bedeutung haben diese jeweils?

- (d) (1 Punkt) Python unterstützt Mehrfachvererbung. Mit welchem Konstrukt ist es möglich in Programmiersprachen ohne Mehrfachvererbung eine ähnliche Flexibilität zu erreichen?

- (e) (1 Punkt) Erklären Sie was mit „Operator überladen“ gemeint ist.

- (f) (2 Punkte) Was wird mit dem Begriff *Reflexion* bezeichnet?
Sind Programmierspachen die über Reflexion verfügen mächtiger als solche die kein Reflexion besitzen? **Begründen Sie ihre Antwort.**

Aufgabe 6: Aggregierte Datentypen

Punkte: ____ / 10

- (a) (1 Punkt) Was verstehen wir unter dem Begriff *aggregierter Datentyp*?

- (b) (2 Punkte) Was ist der Unterschied zwischen einer Liste und einem Array? Wie sieht ihr Speicherverbrauch aus?

- (c) (2 Punkte) Für Wörterbücher ist eine effiziente Implementierung unerlässlich. Auf welche Datenstruktur würden Sie zurückgreifen, wenn Sie ein effizientes Wörterbücher selbst implementieren wollen?

Aufgabe 7: Daten-Informationen-Wissen

Punkte: ____ / 10

(a) **Syntaxdiagramm & EBNF**

```

identifier ::= (letter|"_") (letter | digit | "_")*
letter     ::= lowercase | uppercase
lowercase  ::= "a"... "z"
uppercase  ::= "A"... "Z"
digit      ::= "0"... "9"

```

1. (2 Punkte) Wandeln Sie die oben gegebene formale Notation für die Erzeugung eines Python Bezeichners in EBNF um.

[illegible]

2. (2 Punkte) Stellen Sie die EBNF zusätzlich als Syntaxdiagramm dar.

This image shows a full page of blank graph paper. The grid consists of small, equal-sized squares formed by thin gray lines. There are 20 columns and 20 rows of squares, creating a total area of 400 square units. The grid is perfectly aligned and covers the entire page without any margins or additional markings.

- (b) (3 Punkte) Was beschreiben die folgenden drei Begriffe jeweils?

1. Syntax

2. Semantik

3. Pragmatik

(c) (3 Punkte) Gegeben sei die folgende 32 Zeichen lange Nachricht:

HESSE_LACHT_ZUR_FASSENACHT_HELAU

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

Wie groß ist der Informationsgehalt (Bit) der folgenden Zeichen?

Geben Sie auch den Rechenweg an!

1. H

2. T

3. N

Aufgabe 8: Prozesse & Synchronisation

Punkte: ____ / 10

- (a) (2 Punkte) Wie lauten die vier Strategien zur Lösung einer Verklemmung?

- (b) (1 Punkt) Was sind die Vorteile, was sind die Nachteile der *lightweight Threads*?

- (c) (1 Punkt) Was ist ein *kritischer Abschnitt*?

- (d) (1 Punkt) Was ist eine *atomare Aktion*?

- (e) (2 Punkte) Wie kann es zu einer Verletzung der *fairness-Bedingung* kommen?
Geben Sie ein Beispiel an.

- (f) Das **Erzeuger-Verbraucher** Problem

1. (1 Punkt) Wie viele Semaphoren werden zu Puffersynchronisation des Erzeuger-Verbraucher Problems benötigt?

2. (2 Punkte) Ergänzen Sie die Semaphoren anhand des Codebeispiels zum Lösen des Erzeuger-Verbraucher Problems?

```
while True: # Erzeuger-Thread  
    produce(item)
```

```
    putInBuffer(item)
```

```
while True: # Verbraucher-Thread
```

```
    getFromBuffer(item)
```

```
    consume(item)
```

Aufgabe 9: Algorithmenentwurf

Punkte: ____ / 10

- (a) 1. (1 Punkt) Was Berechnet der folgende Algorithmus?

2. (1 Punkt) Welchem Entwurfsmuster ist er zuzuordnen?

```

1 sets = [[v] for v in g.getvertices()]
2 X = Graph()
3 for e in g.getedges():
4     v1, v2 = e[0][0], e[0][1]
5     s1, s2 = findset(sets, v1), findset(sets, v2)
6     if s1 != s2:
7         X.addedge(v1,v2,e[1])
8         joinset(sets,s1,s2)
9     if len(sets) == 1:
10        break

```

- (b) 1. (3 Punkte) Formulieren Sie in Worten, oder Pseudocode, eine Funktion die mit *logarithmischer* Komplexität nach der Divide & Conquer Methode feststellt, ob ein bestimmter Wert x in einer Liste L enthalten ist.

[illegible]

2. (1 Punkt) Welche Voraussetzung muss für die Liste L gegeben sein, das Ihr Algorithmus korrekt arbeitet?

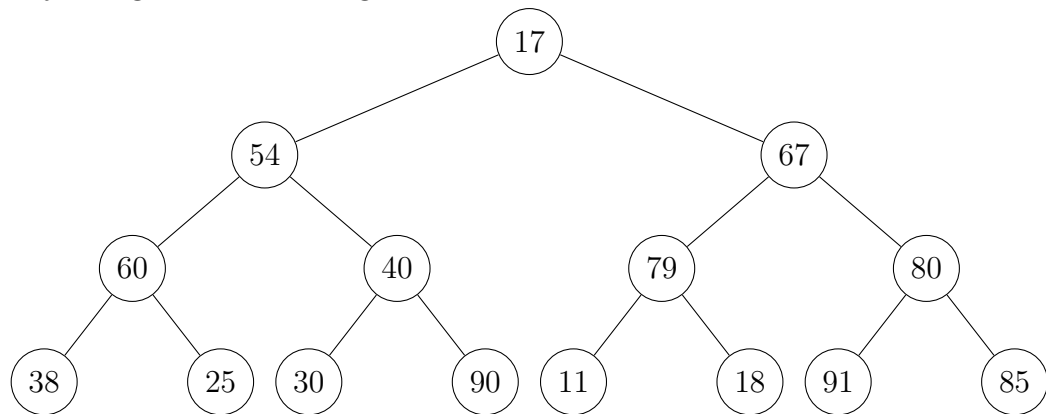
- (c) (1 Punkt) Wie ist die asymptotische Laufzeit für *Mergesort* bei einer Eingabe der Größe n abzuschätzen?

- (d) (1 Punkt) Was muss gegeben sein, um die *kombinatorische Explosion* bei Backtracking Algorithmen zu verhindern?

- (e) 1. (1 Punkt) Warum kann es vorkommen, dass ein Greedy-Algorithmus nicht das *globale Optimum* findet?

2. (1 Punkt) Wie können wir die Güte eines Greedy-Algorithmus verbessern? So dass dieser bessere Chance hat das *globale Optimum* zu finden.

- (d) Notieren Sie in welcher Reihenfolge die Knoten des nachstehenden Baumes bei den jeweiligen Traversierungen besucht werden.



1. (1 Punkt) Preorder:

2. (1 Punkt) Inorder:

3. (1 Punkt) Postorder:

4. (1 Punkt) Levelorder:

Matrikelnummer: _____

