

Probeklausur zur Vorlesung PRG-1 WS 2012/13

1	2	3	4	5	6	7	8	9	10
/10	/6	/6	/10	/8	/10	/6	/11	/4	/4

Summe: ____ / 75

Aufgabe 1: Multiple-Choice-Fragen und Lückentexte (10 Punkte)

1.1 Formale Sprachen

Gegeben sei die folgende Grammatik in EBNF (Startsymbol S):

S = "a", B, "b".

A = "a", {B}, "b", "c".

B = "a", "d" | "a", A, "d".

Welche Wörter gehören zu der durch die Grammatik definierten Sprache?

- ☐ aabcd
- ☐ aadb
- ☐ aaaB*bc
- ☐ aaabcd

1.2 Programmanalyse

Was gibt folgendes Programm aus?

```
s = "CDROM"
for c in s:
    print(chr(ord(c)+1))
```

1.3 Python-Ausdrücke (Version 3.x)

Mit welchen folgenden Python-Ausdrücken lässt sich bestimmen, ob eine beliebige Integerzahl n ungerade ist (d.h. der Wert ist `True`, wenn ungerade, und `False`, wenn gerade)?

- ☐ `n%2 == 1`
- ☐ `n//2*2 != n`
- ☐ `n//2 == 1`
- ☐ `n%2 != 0`
- ☐ keiner der obigen

1.4 Python-Import

Angenommen, eine Funktion `f` wurde aus einem Modul `test` mit `"from test import f"` importiert. Was gilt?

- ☐ die Funktion kann mit `test.f` angesprochen werden
- ☐ die Funktion kann mit `f` angesprochen werden
- ☐ es kann keine Funktion mehr mit dem Namen `f` definiert werden
- ☐ wird eine neue Funktion mit dem Namen `f` definiert, wird `f` überschrieben

1.5 Bäume

In welchem Verhältnis wächst die Höhe eines balancierten Binärbaums zur Anzahl seiner Knoten?

- ☐ logarithmisch
- ☐ linear
- ☐ quadratisch
- ☐ exponentiell

1.6 Bitte ergänzen Sie die Lücken im Text.

In der Informationstheorie ist die [] ein Maß für Unordnung.

Der Informationsgehalt einer Nachricht wächst []

mit der [] Häufigkeit des Auftretens.

1.7 Abstrakte Datentypen

Was ist für die Definition eines abstrakten Datentyps essentiell?

- ☐ verwendete Datentypen
- ☐ abgedeckter Wertebereich
- ☐ verwendete Konstruktoren
- ☐ definierte Methoden
- ☐ definierte Axiome
- ☐ abgedeckte Polymorphie

1.8 Testen

Welche dieser Vorgehensweisen gehören zum Black-Box-Test?

- ☐ Die Testfälle können parallel zur Implementierung entwickelt werden.
- ☐ Testfälle werden aus der Spezifikation eines Systems oder einer Komponente abgeleitet.
- ☐ Ist unabhängig von der Implementierung.
- ☐ Eingabewerte können sich an Kriterien wie Anweisungs-, Ablaufzweig- und Ablaufpfadüberdeckung orientieren.

1.9 Entwurfsmethoden

Welcher dieser Entwurfsmethoden benutzt der Mergesort-Algorithmus?

- ☐ Greedy
- ☐ Backtracking
- ☐ Divide-and-Conquer
- ☐ Scanning

1.10 Prozesse

Welche nebenläufigen Einheiten können sich bei Programmierfehlern gegenseitig die Daten im Hauptspeicher überschreiben?

- ☐ lightweight threads
- ☐ heavyweight threads
- ☐ Leichtgewichtsprozesse
- ☐ Normale Prozesse

Matrikelnummer: _____

Aufgabe 2: Zahlendarstellung (6 Punkte)

- (a) Wandeln Sie die folgenden Dezimalzahlen in Binärzahlen und die Ergebnisse in Hexadezimalzahlen um und ergänzen Sie die Tabelle: (2 Pkt)

Dezimal	Binär	Hexadezimal
251		
1266		

- (b) Was ist der Unterschied zwischen dem Einerkomplement und dem Zweierkomplement? Geben Sie ein Beispiel an. (1 Pkt)

- (c) Stellen Sie die folgenden Dezimalzahlen als Integerzahl im Zweierkomplement dar; verwenden Sie acht Bit: 127, -128, -23 (2 Pkt)

- (d) Welchen Wert haben die folgenden Gleitkommazahlen (IEEE-754, single precision) als Dezimalzahl. Sie können dies auch als Summe von Potenzen formulieren.

1 10010011 000111110000010000000000 (1 Pkt)

0 11111101 000000000000000000000001 (1 Pkt)

Aufgabe 3: Python-Ausdrücke, Programmanalyse (6 Punkte)

3.1 Welche Ausgabe erzeugen folgende Python-Ausdrücke ? (je 0,5 Pkt)

<code>print(2<<1//3 +2)</code>	Antwort: _____
<code>print(2**3%4*2.0)</code>	Antwort: _____
<code>print(int(5.0%3*(4>>2)+0.5))</code>	Antwort: _____
<code>print(3//2*2 == 3/2*2)</code>	Antwort: _____

3.2 Formulieren Sie folgenden mathematischen Ausdruck für eine reellwertige Funktion in Python. Sie dürfen keine Funktionen aus der Python-Klassenbibliothek verwenden. (2 Pkt)

$$y = 5x \left(\sqrt[5]{\frac{10x}{2,4}} + (1+x)^2 \right) - 1$$

3.3 Was ist die Ausgabe des folgenden Python-Skripts: (2 Pkt)

```
L = []
for i in range(21, 1, -2):
    if i%3 == 0:
        continue
    L.append(i)
print(L)
```

Aufgabe 4: Kontrollstrukturen, Rekursion (10 Punkte)

4.1 Gegeben sei folgendes Programm aus alten Zeiten:

```
10 a = 10; x = 2;
20 if x > 2 goto 50
30 a = a-1
40 x = x+1; goto 20
50 if a > 0 goto 30
60 print(x)
```

a) Was ist die Ausgabe des Programms? (1 Pkt.)

b) Schreiben Sie das Programm in strukturiertem Python-Code. (2 Pkt.)

4.2 Wie ist die mathematische Definition der Funktion, die durch folgenden Python-Code für die natürliche Zahl n beschrieben wird (2 Pkt.):

```
def f(n):
    if n == 0:
        return 1
    elif n == 1:
        return 4
    else:
        return 3 * f(n - 2) + f(n - 1)
```

4.3 Implementieren Sie die obige Funktion in einer iterativen Variante, wobei keine `if`-Klauseln verwendet werden dürfen. (5 Pkt.)

Aufgabe 5: Datentypen, Objektorientierung (8 Punkte)

5.1 Lokale und globale Variable (2 Punkte)

Gegeben sei folgender Code :

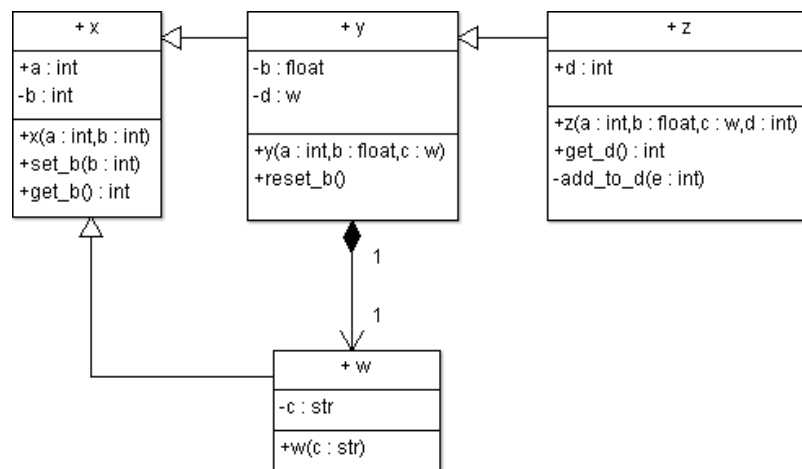
```
a, b = 4, 5
def count(a):
    while a > 0:
        a -= 1
        print(a + b)
```

Welche Ausgabe produziert `count(2)` ?

5.2 Objektorientierte Klassenhierarchien (6 Punkte)

a) Stellen Sie die folgende Klassenhierarchie in Python dar. Für jede Klasse ist in der obersten Zeile der Klassenname vermerkt. Es folgen dann die Instanzvariablen und die Methoden mit ihren Parametern. Die Methode mit demselben Namen der Klasse stellt den Konstruktor dar mit den entsprechenden Funktionsparametern zur Erzeugung des Objektes.

Implementieren Sie die Klassen unter Beachtung der Vererbungsbeziehungen; beachten Sie die Kapselung(- für private, + für public), indem Sie die entsprechenden Namenskonventionen in Python verwenden. In die Methodenrumpfe kann einfach das Schlüsselwort `pass` geschrieben werden.

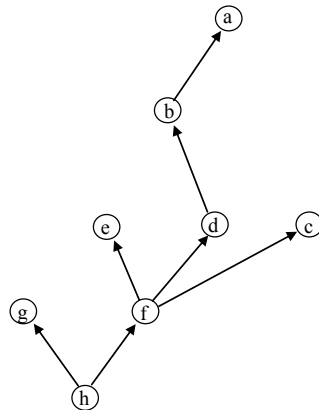


b) Erzeugen Sie schließlich unter Beachtung des Datentyps der Übergabeparameter aus jeder Klasse ein Objekt.

Matrikelnummer: _____

Aufgabe 6: Graphen (10 Punkte)

6.1 Adjazenzmatrix



(a) Stellen Sie für den obigen Baum die Adjazenzmatrix auf. (1,5 Pkt.)

	a	b	c	d	e	f	g	h
a								
b								
c								
d								
e								
f								
g								
h								

(b) Formulieren Sie dies zusätzlich in einer dazu geeigneten Python-Datenstruktur. (2 Pkt.)

6.2 Adjazenzliste

(a) Stellen Sie für den obigen Baum die Adjazenzliste auf. (1,5 Pkt.)

(b) Formulieren Sie dies zusätzlich in einer dazu geeigneten Python-Datenstruktur. (2 Pkt)

Matrikelnummer: _____

6.3 Traversierung (3 Punkte)

Traversieren Sie den Baum und nennen Sie die Reihenfolge, in der die Knoten betrachtet werden, bei (a) Inorder, (b) Postorder und (c) Preorder- Traversierung.

Matrikelnummer: _____

Aufgabe 7 Huffman-Kodierung (6 Punkte)

Gegeben sei die folgende Nachricht:

FISCHERS_FRITZE_FISCHT_FRISCHE_FISCHE

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.

- (a) Wie groß ist der Informationsgehalt der folgenden Einzelzeichen? Geben Sie auch den Rechenweg an! (1 Pkt)

F: $-\log_2(P(X))$ _____

C: _____

R: _____

_: _____

- (b) Erzeugen Sie aus der obigen Nachricht einen Huffman-Code, zeichnen Sie den Codebaum (3 Pkt.) auf und stellen Sie die Codetabelle auf (2 Pkt.).

Aufgabe 8: Debugging und Testen (11 Punkte)

Ein Buchhaltungsprogramm in Python verwaltet zentral in einem globalen Dictionary alle Angestellten einer Firma mit Informationen zum Namen, Gehalt und der derzeitigen Position (alle 3 Werte liegen dabei als Strings vor). Als Schlüssel dient der Name, Gehalt und Position werden als Tupel abgelegt. Das Dictionary Firmendatenbank seien folgende Daten zugewiesen:

```
Firmendatenbank = {"Hans"      : ("2500", "Empfang"),
                  "Meier"     : ("4000", "EDV"),
                  "Schuster"   : ("3000", "Buchhaltung"),
                  "Schmidt"    : ("8000", "Management") }
```

Dieses Programm soll nun von einem Programmierer der Firma um die Funktion erweitert werden, die ein genauso aufgebautes Dictionary aller Angestellten mit deren Gehalt und Position zurückliefert, wobei nur Angestellte in das Dictionary aufgenommen werden, die nach einer Gehaltskürzung um 3% mindestens x Euro verdienen. Der Programmierer schlägt folgende Lösung vor:

```
def finde Alle Mit Mehr Gehalt Als(x):
    arbeitskopie = Firmendatenbank
    namensliste = arbeitsKopie.keys()
    for i in range(1, len(namensliste)):
        name = namensliste[i]
        if Firmendatenbank[name][1]*1.03 > x
            delete arbeitskopie(name)
    return arbeitskopie:
```

8.1 (4 Punkte)

Bei dieser Implementierung hat der Programmierer einige Fehler übersehen, die den Programmablauf mit einer Exception unterbrechen würden oder zu einem fehlerhaften Ergebnis führen. Finden und korrigieren Sie diese Fehler.

Matrikelnummer: _____

8.2 (2 Punkte)

Ordnen Sie die Fehler den folgenden beiden Kategorien zu: Syntaxfehler und logische Fehler.
Beziehen Sie sich auf die gefundenen Fehler in **8.1**.

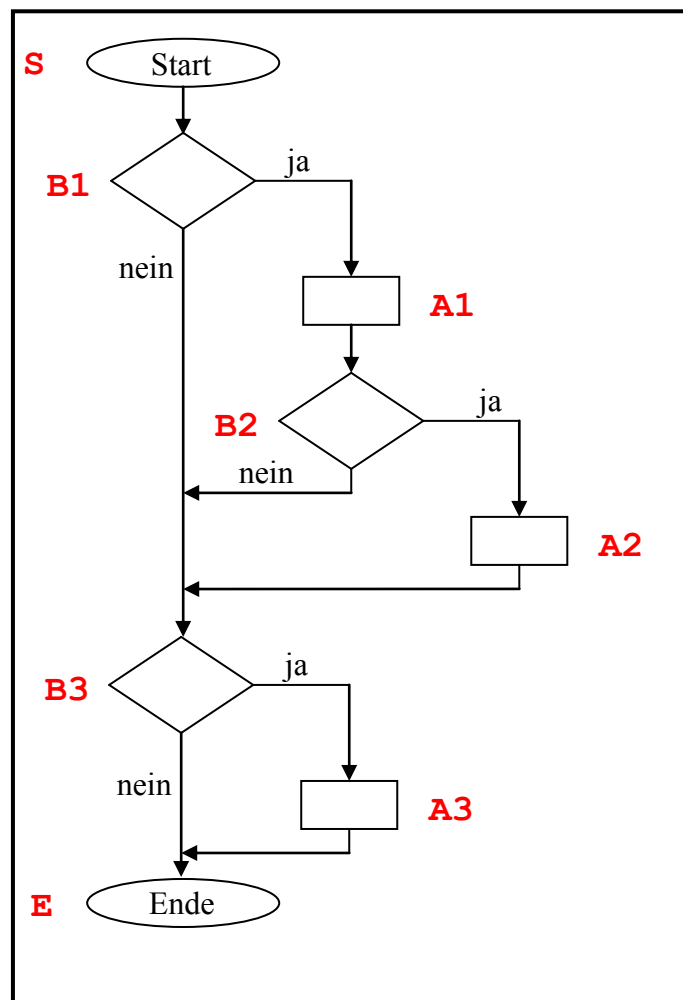
Matrikelnummer: _____

8.3 (1 Punkt)

Gegeben sei folgende Funktion mit den natürlichen Zahlen a , b , c :

```
def Funktion(a, b, c):  
    if b > c:  
        d = b  
        if 2 < c:  
            d = a  
    if a < b:  
        d = b  
    return d
```

Übertragen Sie den Code in die entsprechenden Felder des Diagramms.



Matrikelnummer: _____

8.4 (4 Punkte)

Gegeben sei folgende Testmenge, bei der jeder Testfall die Form (a, b, c) besitzt:

$T = \{(1,4,3), (1,2,3), (1,1,0), (1,1,1)\}$

- Wie viele Testfälle sind für **Ablaufpfadüberdeckung** mindestens notwendig? (1Punkt)
- Erweitern Sie die gegebene Testmenge um nötigen Testfälle, so dass eine minimale Testmenge zur Ablaufpfadüberdeckung entsteht. (3 Punkte)

Aufgabe 9: Entwurfsmethoden und Algorithmen (4 Punkte)

9.1 (4 Punkte)

Erläutern Sie (kurz!) die Entwurfsmethoden

- Greedy

- Divide and Conquer

- Backtracking

Aufgabe 10: Scheduling und Prozesssynchronisierung(4 Punkte)

- (a) Sie haben vier Prozesse (S_1, S_2, S_3, S_4) in Ihrem System und zwei kritische Abschnitte pro Prozess, die unabhängig voneinander durchlaufen werden sollen. Geben Sie ein Schema an, diese mit P- und V-Operationen zu schützen.
- (b) Wie können die P- und V-Operationen hardwaremäßig implementiert werden? Geben Sie die Art der Aktion an und wann beide verschiedenen Hardwareoperationen verwendet werden.