

# Klausur: T121 - Programmiersprachen und Methodik T22a

**QUARTAL: I/2023** 

Name des Prüflings:	Matrikelnummer:	Zenturie:				
Dauer: 90 Minuten	Seiten der Klausur <b>ohne</b> Deckblatt: 14	<b>Datum:</b> 15.03.2023				

Hilfsmittel: NORDAKADEMIE-Taschenrechner.

## Bemerkungen:

• Bitte prüfen Sie zunächst die Klausur auf Vollständigkeit.

• Bitte lösen Sie nicht die Heftung.

Es sind 100 Punkte erreichbar.

Zum Bestehen der Klausur sind 50 Punkte ausreichend.

Aufgabe	Erreichbare Punkte	Erreichte Punkte
1	15	
2	8	
3	6	
4	8	
5	4	
6	12	
7	6	
8	8	
9	12	
10	5	
11	16	
Summe	100	

Note:	Prozentsatz:	Ergänzungsprüfung:
Datum:	Unterschri	ft:
Datum:	Unterschri	ft:



## 1. Single Choice

Welche der folgenden Aussagen ist wahr. In jeder Teilfrage ist nur genau eine Aussage wahr. Setzen Sie also genau ein Kreuz. Sollten Sie mehr als eine Aussage ankreuzen, erhalten Sie null Punkte für diese Teilfrage.

(1.1)	(1 Punkt) Ein Compiler übersetzt
	<ul><li>von einer Hochsprache in Assembler.</li></ul>
	<ul><li>von Assembler in Maschinensprache.</li></ul>
	ovon Maschinensprache in Mikrocode.
(1.2)	(1 Punkt) Ein Assembler übersetzt von
	<ul><li>von einer Hochsprache in Assembler.</li></ul>
	<ul><li>von Assembler in Maschinensprache.</li></ul>
	<ul><li>von Maschinensprache in Mikrocode.</li></ul>
(1.3)	(1 Punkt) Welcher Datentyp ist kein Container-Typ in Python:
	o komplexe Zahl
	○ Tuple
	○ Liste
(1.4)	(1 Punkt) Welche der folgenden Operationen führt zu einem Fehler?
	('Hello', 'World')[0]='Bye'
	('Hello', 'World'][0]='Bye'
(4 <b>5</b> )	<pre>('Hello': 'World')[0]='Bye'</pre>
(1.5)	(1 Punkt) Ein Python-Block endet, wenn
	<ul> <li>○ Codezeilen tiefer eingerückt werden.</li> <li>○ die Finrückung auf die Finrückung des einsehließenden Blackes zurückt.</li> </ul>
	<ul> <li>die Einrückung auf die Einrückung des einschließenden Blockes zurück geht.</li> </ul>
	eine geschweifte Klammer geschlossen wird.
(1.6)	(1 Punkt) Das Iterieren über ein Dictionary mittels for-Anweisung
(110)	gibt die Elemente immer in der Reihenfolge des Einfügens zurück.
	gibt in jedem Durchlauf den zu einem Schlüssel gehörenden Wert zurück
	gibt in jedem Durchlauf einen Schlüssel zurück.
(1.7)	(1 Punkt) Eine Funktion
	○ hat immer einen Rückgabewert.
	<ul> <li>hat mindestens einen Parameter.</li> </ul>
	<ul><li>kann sich selbst aufrufen.</li></ul>
(1.8)	(1 Punkt) Klassen
	<ul> <li>haben einen Zustand, ein Verhalten und eine Identität.</li> </ul>
	<ul><li>werden mit einem Konstruktor erzeugt.</li></ul>
	<ul><li>sind Kategorien oder Schablonen von Dingen.</li></ul>
(1.9)	(1 Punkt) Objekte
	<ul> <li>haben einen Zustand, ein Verhalten und eine Identität.</li> </ul>
	haben die gleiche Identität, wenn sie den gleichen Zustand haben.
	<ul><li>haben kein Verhalten, sondern nur einen Zustand.</li></ul>



(1.10)	(1 Punkt) Konstruktoren
	<ul><li>haben keine Parameter.</li></ul>
	<ul> <li>initialisieren Objekte und bringen sie in einen konsistenten Anfangszustand.</li> </ul>
	<ul><li>implementieren das Verhalten eines Objekts.</li></ul>
(1.11)	(1 Punkt) Attribute einer Python-Klasse
	<ul> <li>sind nur von dieser Klasse selbst veränderbar.</li> </ul>
	<ul><li>speichern den Zustand eines Objekts.</li></ul>
	<ul><li>sind nur in dieser Klasse sichtbar.</li></ul>
(1.12)	(1 Punkt) Die Sichtbarkeit von Attributen einer Python-Klasse
	<ul> <li>wird nicht konsequent durch die Laufzeitumgebung erzwungen.</li> </ul>
	<ul> <li>wird mit Unterstrichen am Ende des Variablennamen gekennzeichnet (z B. variable_).</li> </ul>
	wird mit den Schlüsselwörtern public, private und protected definiert.
(1.13)	(1 Punkt) Methoden haben im Gegensatz zu Funktionen
	<ul><li>keine Sichtbarkeitsdefinition.</li></ul>
	<ul><li>keinen Rückgabewert.</li></ul>
	immer den ersten Parameter self.
(1.14)	(1 Punkt) Abstrakte Methoden haben
	<ul><li>keine Sichtbarkeitsdefinition.</li></ul>
	keinen Parameter self.
	<ul><li>keine Implementierung.</li></ul>
(1.15)	(1 Punkt) Abstrakte Klassen können
	<ul> <li>bilden vollständige Geschäftsobjekte aus der realen Welt ab.</li> </ul>
	<ul><li>haben keine Subklassen.</li></ul>
	<ul><li>nicht zu Objekten instanziiert werden.</li></ul>

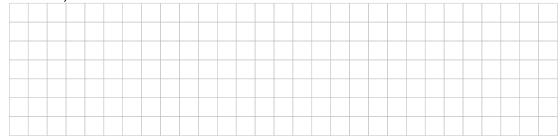


#### 2. Datentypen

Gegeben sei folgende Python-Datenstruktur:

Ermitteln Sie Kommandos, um folgende Daten aus der Datenstruktur zu extrahieren:

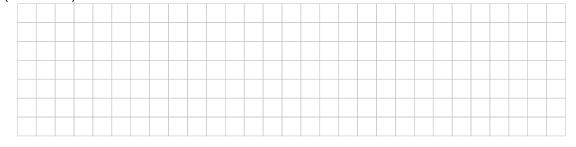
#### (2.1) (2 Punkte) ob die zweite Person Hobbies hat



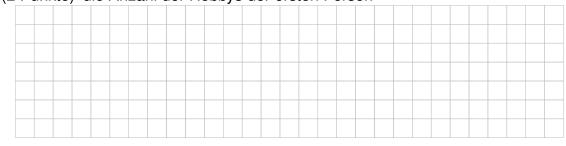
(2.2) (2 Punkte) die Initialen der zweiten Person



(2.3) (2 Punkte) den Vornamen der zweiten Person



(2.4) (2 Punkte) die Anzahl der Hobbys der ersten Person





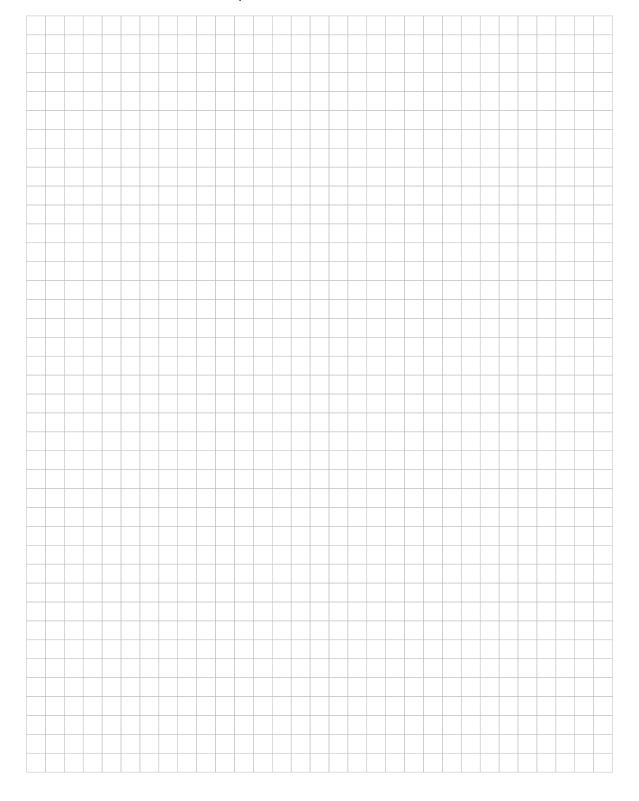
## 3. (6 Punkte) Fibonacci-Zahlen

Schreiben Sie ein Programm, das die n-te Fibonacci-Zahl ermittelt.

Eine Fibonacci-Zahl n ist definiert als die Summe Ihrer beiden Vorgänger n-1 und n-2. Die ersten beiden Fibonacci Zahlen sind mit 0 und 1 definiert. Wie lautet die Fibonacci-Zahl 300?

Es ergibt sich die folgende mathematische Beschreibung:

$$fib(i) = \left\{ \begin{array}{ll} 0 & \text{f\"{u}r} \ i = 0 \\ 1 & \text{f\"{u}r} \ i = 1 \\ fib(i-2) + fib(i-1) & \text{sonst} \end{array} \right.$$





## 4. Auswertung von Python-Code

Im Folgenden werden Ihnen Python-Code-Fragmente gegeben. Schreiben Sie in das dafür vorgesehene Feld, was jedes dieser Code-Fragmente ausgibt.

```
(4.1) (2 Punkte)
    def doSomething(a, b):
    if a - b >= 0:
        return 1 + doSomething(a - b, b)
    return 0
```

## print(doSomething(23, 5))



(4.2) (2 Punkte)

```
a = 3

if a % 2 != 0:

a = a - 5

if a < 0:

a *= -1

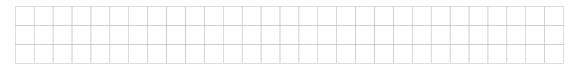
if a % 2 != 1:

a /= 2

else:

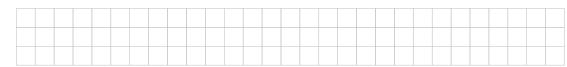
a *= 2
```

## print(a)



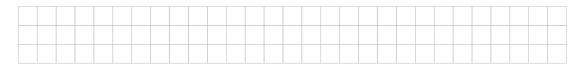
(4.3) (2 Punkte)

#### print(sum)



(4.4) (2 Punkte)

#### print(sum)





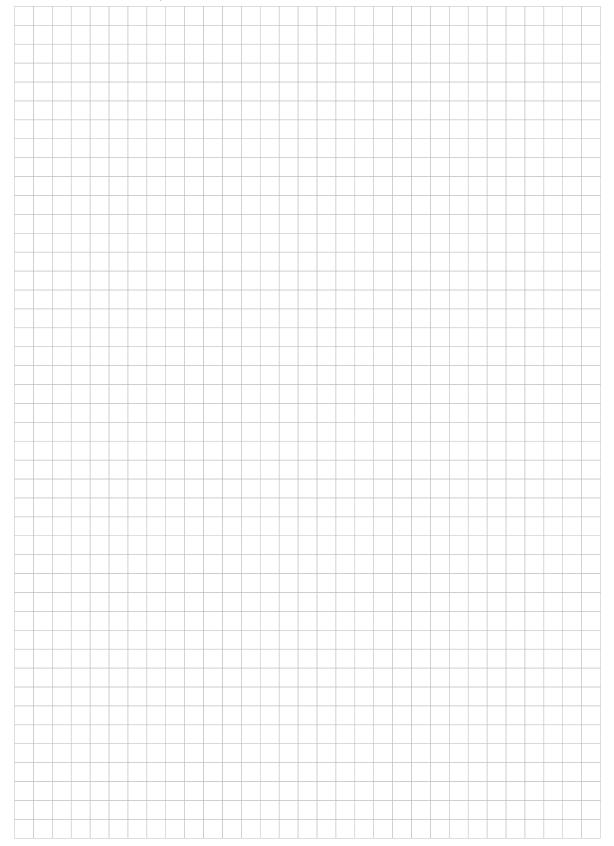
## 5. (4 Punkte) **Schleifen**

Gegeben sei eine Liste in der folgenden Form:

$$I = [4, 6, 1, -2, 9, 4, 5, 8]$$

Schreiben Sie ein Python-Programm, dass durch die Liste iteriert und das maximale Element ermittelt.

Gehen Sie davon aus, dass die Liste immer mindestens ein Element enthält.





## 6. (12 Punkte) Fehlersuche

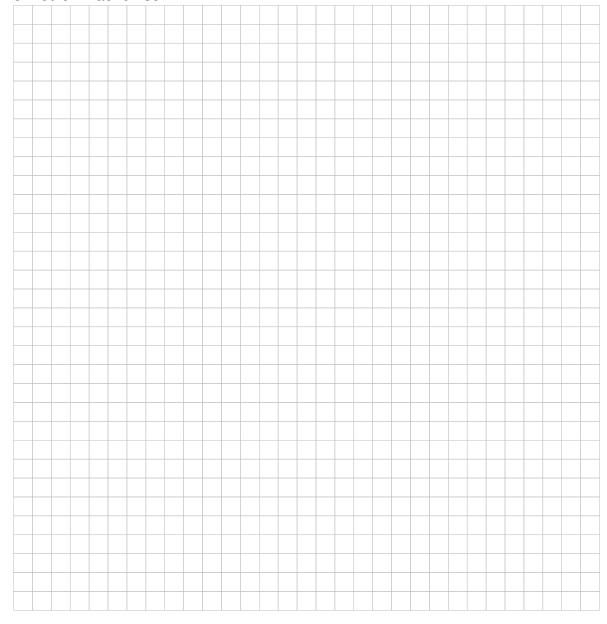
Gegeben sei die folgende Funktion in Python, die alle Elemente einer Liste addiert und die Summe zurückgibt.

```
sumv = 0

def sum-all(int[] liste):
    for el in range(liste)
        sumv += liste[el]:
    return sumv.value
```

Dieser Quelltext hat 8 syntaktische und semantische Fehler. Markieren Sie bitte jeden gefundenen im Quelltext Fehler einzeln. Sollten Kommandos an der falschen Stelle stehen, markieren Sie mit einem Pfeil, wo das Kommando korrekterweise stehen müsste.

Schreiben Sie vollständig korrigierten Quellcode so hin, dass er das macht, was er vermutlich machen soll.





## 7. (6 Punkte) Funktion mean(a)

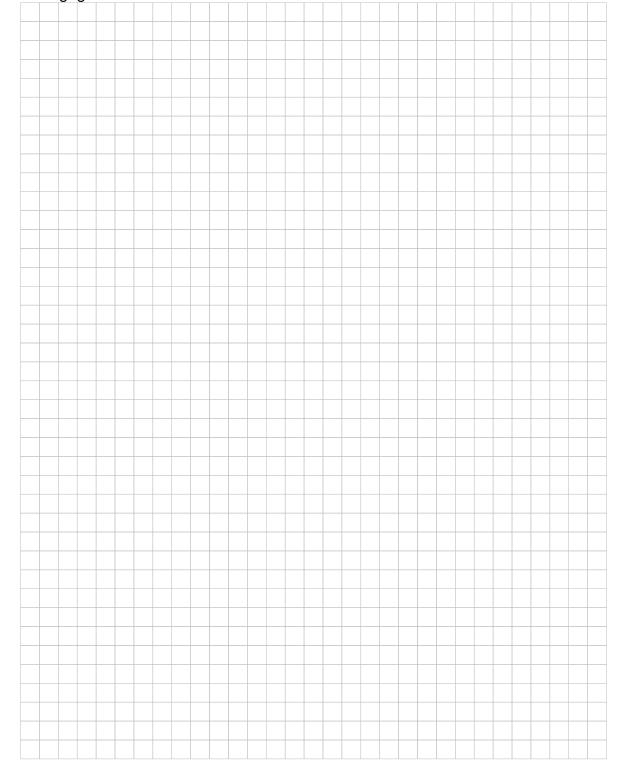
Schreiben Sie eine Funktion mean(a) in Python, die eine Liste a als Parameter erhält und das arithmetische Mittel aller Werte aus dieser Liste ermittelt.

Gehen Sie davon aus, dass die Liste mind. ein Element enthält.

Das arithmetische Mittel einer Folge von Werten  $x_1, ..., x_n$  berechnet sich wie folgt:

$$\frac{1}{n} \sum_{i=1}^{n} x_i$$

Das Ergebnis soll als Rückgabewert der Funktion mean(a) an die aufrufende Funktion zurückgegeben werden.

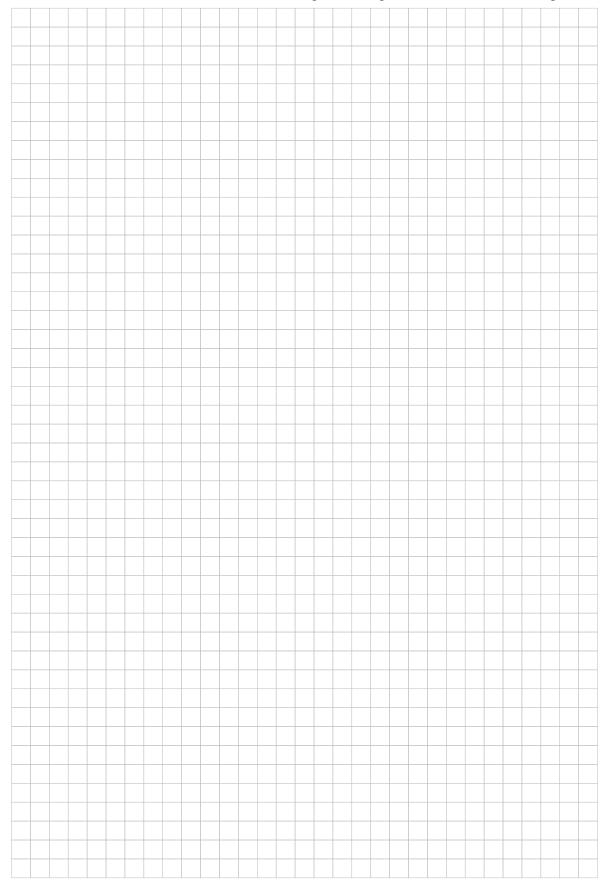




## 8. (8 Punkte) Teiler

Schreiben sie eine Funktion get\_dividers(num), die eine Liste mit allen Teilern der Zahl num zurückgibt.

Bspw. gibt diese Funktion bei Aufruf von get\_dividers(12) die Liste [1, 2, 3, 4, 6, 12] zurück, da dies die Teiler sind, die eine ganzzahlige Division von 12 ermöglichen.





## 9. (12 Punkte) Zwölffache Quersumme

Schreiben Sie eine Funktion cross\_sum\_12(num), die überprüft, ob die mit 12 multiplizierte Quersumme von num dem Wert von num entspricht. Verwenden Sie **keine** String-Funktionen zur Lösung dieser Aufgabe!

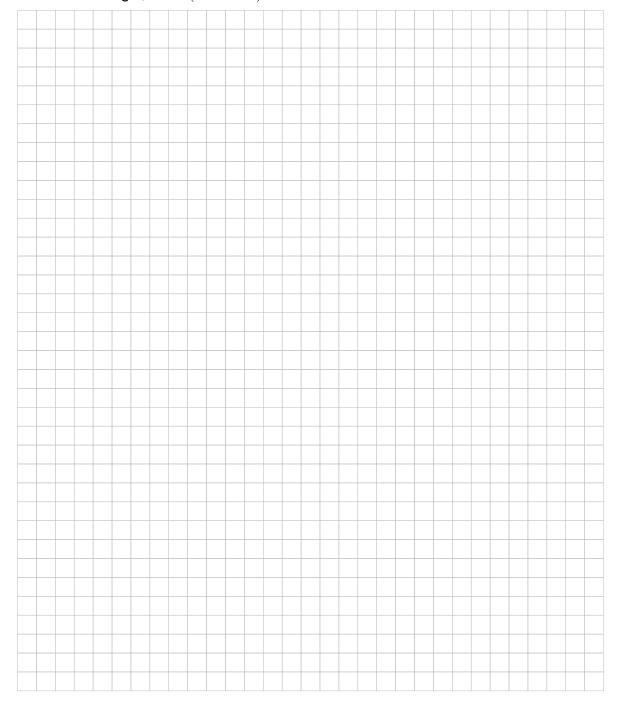
Beachten Sie, dass num auch für Zahlen mit zahlreichen Stellen (auch mehr als drei Stellen) funktionieren soll.

Geben Sie im Anschluss unter Nutzung der gerade entwickelten Funktion alle dreistelligen Zahlen aus, bei denen die 12-fache Quersumme dem Zahlenwert entspricht.

## Beispiel zur Verdeutlichung:

Für die Zahl 108 gibt die implementierte Funktion  ${\tt cross\_sum\_12}$  den Wert  ${\it True}$  zurück, denn

- die Quersumme von 108 ist 1 + 0 + 8 = 9
- außerdem gilt, dass  $(1+0+8) \cdot 12 = 9 \cdot 12 = 108$ .





#### 10. (5 Punkte) Soziales Netzwerk 2

Sie haben bereits einige Klassen für das soziale Netzwerk *TRUTH Antisocial* implementiert. Realisieren Sie nun eine Version, in der die Post-Klasse eine abstrakte Klasse ist.

- Achten Sie darauf, dass jede Klasse nur die Datenhoheit über seine Attribute hat.
- Achten Sie auch darauf, dass es möglichst keine Code-Duplikate gibt.

Die Klasse NewsFeed bleibt unverändert:

```
class NewsFeed:
    def __init__(self):
        self.__posts = []

def add_post(self, post):
        self.__posts.append(post)

def print_feed(self):
    for p in self.__posts:
        p.display()

feed = NewsFeed()
feed.add_post(PhotoPost("Ursula_Urlauberin", "photo.jpg", "Bildunterschrift"))
feed.add_post(MessagePost("Ulf_Urlaub", "Ist_das_Uphuser_Meer_nicht_toll?"))
```





#### 11. Audio-Medien

Gegeben ist der folgende Python-Code zur Implementierung der Vererbungshierarchie Audio-Medien:

```
from abc import ABC, abstractmethod
  class Disc(ABC):
      def __init__(self, nr_tracks):
           self.__nr_tracks = nr_tracks
      def get_tracks(self):
          return self.__nr_tracks
      @abstractmethod
      def play(self):
          pass
  class CompactDisc(Disc):
      def __init__(self):
          super().__init__(2)
      def play(self):
          self.__start_cd_player()
  class MP3(Disc):
      def __init__(self):
          super(). __init__(8)
      def play(self):
          self.__start_dvd_player()
  Beantworten Sie die folgenden Fragen:
(11.1) (2 Punkte) Was passiert bei folgendem Aufruf? Begründen Sie kurz.
     MP3 = Disc(4)
```



(11.2)	•	l. W	<i>l</i> arı	um	nι	ıtze	en								•														Bei- n wir
(11.3)	med: Imp ram das	ia = lem eter s di we	= [ ent ük ese nn	Con ier oer e a zu	mpa en gib bg kü	si Si ot u	Di e ( unc pie	sc ein d di	() : e F e a	ur alle de	om nkti N	pacion led	etD pl lier len	is .ay n <b>m</b> .	c() _me nit nr(	), edi <b>mi</b> Co	DV a( <b>nd</b> de	D() es	), der <b>ter</b> oll (	Co m is	an <b>vie</b> ne '	die <b>r A</b> Ver	e N <b>uc</b> än	/led <b>lio</b> de	die <b>-Tr</b> run	nlis <b>ac</b> ng	ste <b>ks</b> fun	als mit	Pa- tteilt, onie- sen)



(11.4)	(6 Punkte) Was bedeuten die zwei Unterstriche, mit denen einige Methoden und das Attributnr_tracks anfangen? Welche Sichtbarkeiten kennt Python noch, was bedeuten diese und wie werden sie gekennzeichnet?