

# KLAUSUR

## PROGRAMMIERUNG 1

28. MÄRZ 2025

Bedingungen der Klausur:

1. Es handelt sich um eine open book Klausur. **Untersagt** sind jedoch alle Arten von Kommunikation mit anderen, auch nicht mit KI-Portalen, wie z.B. Chat GPT, Bing-Suche etc. Es führt bereits zum Ausschluss, wenn Programme zur Kommunikation (E-Mail, Slack, WhatsApp, Signal, ...) geöffnet sind bzw. wenn Webseiten zur Kommunikation geöffnet sind (z.B. Chat GPT, ...). Schließen Sie vor Klausur also alle entsprechenden Programme und Webseiten!
2. Am Ende der Prüfung: Öffnen Sie den Dateixplorer/Finder und wechseln Sie in Ihren Workspace. Laden Sie entweder alle **\*.java**-Dateien aus dem package **klausur** hoch oder zippen Sie den package-Ordner und laden Sie die zip-Datei in Moodle hoch (bei Aufgabe **Klausur2PZ**)!
3. Es sind insgesamt 59 Punkte zu erzielen (Teil 1: 10 Pkt., Teil 2: 22 Pkt., Teil 3: 23 Pkt., fehlerfreies Programm: 4 Pkt.).
4. Schreiben Sie Ihre Klassen im package **klausur**!

Notenspiegel:

Note	Min	Max
1,0	56,0	59,0
1,3	53,0	55,5
1,7	50,0	52,5
2,0	47,0	49,5
2,3	44,0	46,5
2,7	41,0	43,5
3,0	38,0	40,5
3,3	35,0	37,5
3,7	32,0	34,5
4,0	29,5	31,5
5,0	0	29,0

**Teil 1 (Klasse KeyValue)****10 Punkte**

<p>Erstellen Sie eine Klasse <b>KeyValue</b>.</p> <p>Objektvariablen sind</p> <ul style="list-style-type: none"> <li>• <b>key</b> vom Typ <b>int</b> und</li> <li>• <b>value</b> vom Typ <b>String</b>.</li> </ul> <p>Die Objektvariablen sind nur innerhalb der Klasse sichtbar!</p>	1 Pkt.
<p>Erstellen Sie für die Klasse <b>KeyValue</b> einen parametrisierten Konstruktor, dem als Parameter ein <b>int key</b> und ein <b>String value</b> übergeben werden. Initialisieren Sie mit dem Parameterwerten die Objektvariablen.</p>	1 Pkt.
<p>Erstellen Sie für die beiden Objektvariablen die Getter <b>getKey()</b> und <b>getValue()</b>.</p>	1 Pkt.
<p>Erstellen Sie für die Objektvariable <b>value</b> einen Setter <b>setValue(String value)</b>.</p>	1 Pkt.
<p><u>Überschreiben</u> Sie die Objektmethode <b>toString()</b>, so dass ein <b>KeyValue</b>-Objekt in der folgenden Form ausgegeben wird (Beispielwert):</p> <p><b>  1   Potsdam  </b></p> <ul style="list-style-type: none"> <li>- es kommt also zunächst ein „ “,</li> <li>- dann der <b>key</b> (reservieren Sie dafür <b>3</b> Stellen),</li> <li>- dann wieder ein „ “,</li> <li>- dann der <b>value</b> (reservieren Sie dafür <b>10</b> Stellen) und</li> <li>- dann erneut ein „ “</li> </ul>	1 Pkt.
<p><u>Überschreiben</u> Sie die Objektmethode <b>equals(Object o)</b>. Diese Methode gibt ein <b>true</b> zurück, wenn die <b>keys</b> von <b>this</b> und <b>o</b> gleich sind (und <b>false</b> sonst).</p>	2 Pkt.
<p>Erstellen Sie eine <b>Programmklasse</b> mit <b>main()</b>-Methode. Kopieren Sie in die <b>main()</b>-Methode zunächst folgende Deklaration:</p> <pre>String[] staedte = { "Berlin", "Potsdam", "Hamburg", "Magdeburg", "Bremen", "Dresden", "Leipzig", "Halle", "Dortmund", "Stuttgart", "Mainz", "Frankfurt" };</pre> <ul style="list-style-type: none"> <li>- Erstellen Sie eine Schleife, so dass Sie durch das <b>staedte</b>-Array laufen.</li> <li>- Verwenden Sie die jeweilige Stadt aus dem <b>staedte</b>-Array als <b>value</b> und den jeweiligen Index aus dem <b>staedte</b>-Array als <b>key</b> und erzeugen Sie damit jeweils ein <b>KeyValue</b>-Objekt.</li> </ul>	3 Pkt.

- Geben Sie jedes **KeyValue**-Objekt mithilfe der **toString()**-Methode auf die Konsole aus. Es entsteht folgende Ausgabe:

```
| 0 | Berlin |
| 1 | Potsdam |
| 2 | Hamburg |
| 3 | Magdeburg |
| 4 | Bremen |
| 5 | Dresden |
| 6 | Leipzig |
| 7 | Halle |
| 8 | Dortmund |
| 9 | Stuttgart |
| 10 | Mainz |
| 11 | Frankfurt |
```

**Teil 2 (Klasse Table)****22 Punkte**

<p>Erstellen Sie eine Klasse <b>Table</b>. Objektvariable ist</p> <ul style="list-style-type: none"> <li>• <b>entries</b> vom Typ <b>KeyValue[]</b>, //Tabelle mit lauter Schlüssel-Werte-Paaren</li> </ul> <p>Die Objektvariable ist nur innerhalb der Klasse sichtbar!</p>	1 Pkt.
<p>Erstellen Sie für die Klasse <b>Table</b> einen parameterlosen Konstruktor. Innerhalb des Konstruktors wird das <b>entries</b>-Array mit der Länge <b>0</b> erzeugt.</p>	1 Pkt.
<p>Erstellen Sie eine Objektmethode <b>containsKey(int key)</b>, die ein <b>true</b> zurückgibt, wenn ein <b>KeyValue</b>-Objekt mit dem Schlüssel <b>key</b> in <b>entries</b> enthalten ist und <b>false</b> sonst.</p>	2 Pkt.
<p>Erstellen Sie eine Objektmethode <b>put(int key, String value)</b>. Diese Methode fügt ein <b>KeyValue</b>-Objekt in die Tabelle <b>entries</b> ein, wenn der Schlüssel <b>key</b> noch nicht in <b>entries</b> existiert. Existiert <b>key</b> jedoch bereits, wird nur der dazugehörige Wert durch den neuen <b>value</b> ersetzt. Es gibt also 2 Möglichkeiten:</p> <ol style="list-style-type: none"> <li>1. <b>key</b> existiert noch nicht in <b>entries</b>. Dann wird ein neues <b>KeyValue</b>-Objekt bestehend aus (<b>key,value</b>) <b>entries</b> hinzugefügt (<b>entries</b> wird um <b>1</b> größer).</li> <li>2. <b>key</b> existiert bereits. Dann wird nur der dazugehörige Wert mit <b>value</b> aktualisiert.</li> </ol> <p>Im ersten Fall (1.) gibt die <b>put(int, String)</b>-Methode ein <b>true</b> zurück, im 2. Fall <b>false</b>.</p>	5 Pkt.
<p>Erstellen Sie eine Objektmethode <b>fill(String[] values)</b>. Diese fügt neue <b>KeyValue</b>-Objekte <b>entries</b> hinzu. Nutzen Sie dazu die <b>put(int, String)</b>-Methode. Es werden alle Einträge aus <b>values</b> der Table <b>entries</b> hinzugefügt.</p> <p><b>Aber Achtung!</b> Achten Sie darauf, dass auch tatsächlich <i>neue</i> Einträge hinzugefügt werden, d.h. die verwendeten Schlüssel sollen zuvor noch nicht existieren.</p> <p><b>Tipp:</b> Sie können ja versuchen, den Index des Arrays <b>values</b> als <b>key</b> zu verwenden. Sollte dieser <b>key</b> jedoch bereits existieren, erhöhen Sie den Wert für <b>key</b> solange, bis er noch nicht in <b>entries</b> existiert und verwenden ihn.</p>	3 Pkt.
<p>Erstellen Sie eine Objektmethode <b>getEntry(int key)</b>, die das <b>KeyValue</b>-Objekt mit dem Schlüssel <b>key</b> aus <b>entries</b> zurückgibt. Ist ein solches Objekt nicht in <b>entries</b> vorhanden, wird <b>null</b> zurückgegeben.</p>	2 Pkt.

<p>Erstellen Sie eine Objektmethode <b>getSize()</b>, die die Anzahl der Einträge in <b>entries</b> zurückgibt.</p>	1 Pkt.
<p>Überschreiben Sie die Methode <b>toString()</b> so, dass ein <b>String</b> in der folgenden Form zurückgegeben wird (Beispielwerte):</p> <p><b>12 Eintraege</b>        -----          0   Berlin            1   Potsdam            2   Hamburg            3   Magdeburg            4   Bremen            5   Dresden            6   Leipzig            7   Halle            8   Dortmund            9   Stuttgart            10   Mainz            11   Frankfurt          -----</p> <p>d.h.</p> <ul style="list-style-type: none"> <li>- erst kommt die Anzahl von Einträgen in <b>entries</b> gefolgt von „<b>Eintraege</b>“</li> <li>- dann eine Linie (Länge muss nicht passen)</li> <li>- dann alle <b>KeyValue</b>-Objekte aus <b>entries</b></li> <li>- dann noch eine Linie (Länge muss nicht passen)</li> </ul>	3 Pkt.
<p>Erzeugen Sie in der <b>main()</b>-Methode der <b>Programmklasse</b> ein <b>Table</b>-Objekt. Rufen Sie für dieses Objekt</p> <ul style="list-style-type: none"> <li>- die <b>fill()</b>-Methode auf und übergeben dieser Methode das <b>staedte</b>-Array.</li> <li>- Geben Sie das Table-Objekt mithilfe der <b>toString()</b>-Methode auf die Konsole aus.</li> </ul> <p>Es entsteht folgende Ausgabe:</p> <p><b>12 Eintraege</b>        -----          0   Berlin            1   Potsdam            2   Hamburg            3   Magdeburg            4   Bremen            5   Dresden            6   Leipzig  </p>	2 Pkt.

<pre>   7   Halle          8   Dortmund       9   Stuttgart     10   Mainz         11   Frankfurt    -----  - Rufen sie die <b>fill()</b>-Methode erneut auf und übergeben der Methode erneut das <b>staedte</b>-Array.  - Geben Sie das Table-Objekt mithilfe der <b>toString()</b>-Methode erneut auf die Konsole aus.  Es entsteht folgende Ausgabe:  <b>24 Eintraege</b> -----   0   Berlin         1   Potsdam        2   Hamburg        3   Magdeburg      4   Bremen         5   Dresden        6   Leipzig        7   Halle          8   Dortmund       9   Stuttgart     10   Mainz         11   Frankfurt     12   Berlin        13   Potsdam       14   Hamburg       15   Magdeburg     16   Bremen        17   Dresden       18   Leipzig       19   Halle         20   Dortmund      21   Stuttgart     22   Mainz         23   Frankfurt    ----- </pre>	
<p>Rufen Sie in der <b>main()</b>-Methode der <b>Programmklasse</b> für das <b>Table</b>-Objekt die Methoden</p>	<p>1 Pkt.</p>

<ul style="list-style-type: none"> <li>- <b>put(0, "Potsdam")</b> (mit Ausgabe des Rückgabewertes auf die Konsole) und</li> <li>- <b>put(24, "Potsdam")</b> (mit Ausgabe des Rückgabewertes auf die Konsole) auf und geben Sie danach mithilfe</li> <li>- der <b>toString()</b>-Methode das <b>Table</b>-Objekt auf die Konsole aus. Es entsteht folgende Ausgabe:</li> </ul> <pre> false true 25 Eintraege -----   0   Potsdam     1   Potsdam     2   Hamburg     3   Magdeburg     4   Bremen     5   Dresden     6   Leipzig     7   Halle     8   Dortmund     9   Stuttgart     10   Mainz     11   Frankfurt     12   Berlin     13   Potsdam     14   Hamburg     15   Magdeburg     16   Bremen     17   Dresden     18   Leipzig     19   Halle     20   Dortmund     21   Stuttgart     22   Mainz     23   Frankfurt     24   Potsdam   -----                     </pre>	
<p>Testen Sie die <b>getEntry()</b>-Methode mit einem existierenden <b>key</b> und geben Sie das zurückgegebene <b>KeyValue</b>-Objekt auf die Konsole aus.</p>	<p>1 Pkt.</p>

**Teil 3 (weiter mit Klasse Table)****23 Punkte**

Erstellen Sie in der Klasse <b>Table</b> eine Objektmethode <b>getKeys()</b> . Diese Methode gibt alle <b>keys</b> aus <b>entries</b> als <b>int[]</b> zurück.	2 Pkt.
Erstellen Sie in der Klasse <b>Table</b> eine Objektmethode <b>printKeys()</b> . Diese Methode ruft die <b>getKeys()</b> -Methode auf und gibt das <b>int[]</b> , das alle Schlüssel enthält, in der folgenden Form auf die Konsole aus (Beispielwerte):  <b>0 - 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12</b>  d.h. es werden alle Schlüssel getrennt durch „ - „ auf die Konsole ausgegeben	2 Pkt.
Erstellen Sie eine Objektmethode <b>contains(String[] arr, String s)</b> . Diese Methode gibt ein <b>true</b> zurück, wenn <b>s</b> in <b>arr</b> enthalten ist und <b>false</b> sonst.	1 Pkt.
Erstellen Sie eine Objektmethode <b>getValues()</b> . Diese Methode gibt alle <b>values</b> aus <b>entries</b> als <b>String[]</b> zurück.  <b>Aber Achtung!</b> Kein <b>value</b> darf im zurückgegebenen <b>String[]</b> doppelt vorkommen.  <b>Tip:</b> Nutzen Sie die <b>contains(String[], String)</b> -Methode.	4 Pkt.
Erstellen Sie in der Klasse <b>Table</b> eine Objektmethode <b>printValues()</b> . Diese Methode ruft die <b>getValues()</b> -Methode auf und gibt das <b>String[]</b> , das alle Werte enthält, in der folgenden Form auf die Konsole aus (Beispielwerte):  <b>Potsdam - Hamburg - Magdeburg - Bremen - Dresden - Leipzig - Halle - Dortmund - Stuttgart - Mainz - Frankfurt - Berlin</b>  d.h. es werden alle Werte getrennt durch „ - „ auf die Konsole ausgegeben	1 Pkt.
Erstellen Sie eine Objektmethode <b>sort(boolean asc)</b> . Diese Methode sortiert die Tabelle <b>entries</b> nach Schlüsselgröße. Ist <b>asc</b> gleich <b>true</b> , wird aufsteigend sortiert. Ist <b>asc</b> gleich <b>false</b> wird absteigend sortiert.	3 Pkt.
Erstellen Sie eine Objektmethode <b>remove(int key)</b> . Diese Methode löscht das <b>KeyValue</b> -Objekt aus <b>entries</b> , das den Schlüssel <b>key</b> hat. Wird ein solcher Eintrag gelöscht, gibt die <b>remove()</b> -Methode <b>true</b> zurück. Ist ein solcher Eintrag nicht vorhanden, gibt die Methode <b>false</b> zurück.	3 Pkt.
Erstellen Sie eine Objektmethode <b>remove(String value)</b> . Diese Methode löscht die <b>KeyValue</b> -Objekte aus <b>entries</b> , die den Wert <b>value</b> haben. Wird ein solcher Eintrag bzw.	4 Pkt.



<p>solche Einträge gelöscht, gibt die <b>remove()</b>-Methode <b>true</b> zurück. Ist ein solcher Eintrag nicht vorhanden, gibt die Methode <b>false</b> zurück.</p> <p>Beachten Sie, dass es mehrere gleiche <b>values</b> in <b>entries</b> geben kann (im Gegensatz zu <b>key</b>).</p>	
<p>Testen Sie alle Objektmethoden aus Teil 3. Es entstehen z.B. folgende Ausgaben:</p> <p><b>0 - 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 - 18 - 19 - 20 - 21 - 22 - 23 - 24</b></p> <p><b>Potsdam - Hamburg - Magdeburg - Bremen - Dresden - Leipzig - Halle - Dortmund - Stuttgart - Mainz - Frankfurt - Berlin</b></p> <p><b>25 Eintraege</b></p> <p>-----</p> <p>  24   Potsdam  </p> <p>  23   Frankfurt  </p> <p>  22   Mainz  </p> <p>  21   Stuttgart  </p> <p>  20   Dortmund  </p> <p>  19   Halle  </p> <p>  18   Leipzig  </p> <p>  17   Dresden  </p> <p>  16   Bremen  </p> <p>  15   Magdeburg  </p> <p>  14   Hamburg  </p> <p>  13   Potsdam  </p> <p>  12   Berlin  </p> <p>  11   Frankfurt  </p> <p>  10   Mainz  </p> <p>  9   Stuttgart  </p> <p>  8   Dortmund  </p> <p>  7   Halle  </p> <p>  6   Leipzig  </p> <p>  5   Dresden  </p> <p>  4   Bremen  </p> <p>  3   Magdeburg  </p> <p>  2   Hamburg  </p> <p>  1   Potsdam  </p> <p>  0   Potsdam  </p> <p>-----</p> <p><b>25 Eintraege</b></p> <p>-----</p> <p>  0   Potsdam  </p> <p>  1   Potsdam  </p>	<p>3 Pkt.</p>

2   Hamburg	
3   Magdeburg	
4   Bremen	
5   Dresden	
6   Leipzig	
7   Halle	
8   Dortmund	
9   Stuttgart	
10   Mainz	
11   Frankfurt	
12   Berlin	
13   Potsdam	
14   Hamburg	
15   Magdeburg	
16   Bremen	
17   Dresden	
18   Leipzig	
19   Halle	
20   Dortmund	
21   Stuttgart	
22   Mainz	
23   Frankfurt	
24   Potsdam	
-----	
remove(9) = true	
remove(25) = false	
<b>24 Eintraege</b>	
-----	
0   Potsdam	
1   Potsdam	
2   Hamburg	
3   Magdeburg	
4   Bremen	
5   Dresden	
6   Leipzig	
7   Halle	
8   Dortmund	
10   Mainz	
11   Frankfurt	
12   Berlin	
13   Potsdam	
14   Hamburg	
15   Magdeburg	
16   Bremen	
17   Dresden	
18   Leipzig	

<pre>  19   Halle          20   Dortmund        21   Stuttgart       22   Mainz           23   Frankfurt       24   Potsdam       -----  remove(Potsdam) = true remove(Brandenburg) = false  20 Eintraege -----    2   Hamburg         3   Magdeburg       4   Bremen          5   Dresden         6   Leipzig         7   Halle           8   Dortmund        10   Mainz           11   Frankfurt       12   Berlin          14   Hamburg         15   Magdeburg       16   Bremen          17   Dresden         18   Leipzig         19   Halle           20   Dortmund        21   Stuttgart       22   Mainz           23   Frankfurt     -----</pre>	
---	--

**Viel Erfolg!**

KeyValue	
-	key : int
-	value : String
+ KeyValue(int, String)	
+	getKey() : int
+	getValue() : String
+	setValue(String)
+	toString() : String
+	equals(Object) : boolean

Table	
-	entries : KeyValue[]
+ Table()	
+	containsKey(int) : boolean
+	put(int, String) : boolean
+ fill(String[])	
+	getEntry(int) : KeyValue
+	getSize() : int
+	toString() : String
+	getKeys() : int[]
+ printKeys()	
+	contains(String[], String) : boolean
+	getValues() : String[]
+ printValues()	
+ sort(boolean)	
+	remove(int) : boolean
+	remove(String) : boolean