

**KLAUSUR**  
**PROGRAMMIERUNG 1**  
**30. MÄRZ 2022**

Bedingungen der Klausur:

1. Fragen stellen Sie bitte im Zoom-Chat.
2. Bitte im Source-Code nicht Ihren Namen vermerken (also nicht `@author`-Tag o.ä.). Die Klausuren werden anonym kontrolliert.
3. Am Ende der Prüfung: Öffnen Sie den Dateexplorer/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**!).
4. Es sind insgesamt 67 Punkte zu erzielen (Teil 1: 17 Pkt., Teil 2: 19 Pkt., Teil 3: 27 Pkt., fehlerfreies Programm: 4 Pkt.).
5. Schreiben Sie Ihre Klassen im package `klausur`!

Notenspiegel:

Note	1,0	1,3	1,7	2,0	2,3	2,7	3,0	3,3	3,7	4,0	5,0
Punkte	>=	60.0	59.5	56.0	53.0	49.5	46.0	43.0	39.5	36.0	<
	63.0	63.0	56.5	53.5	50.0	46.5	43.5	40.0	36.5	33.5	33.0

**Teil 1 (Klasse Akku – Teil 1)**

**17 Punkte**

<p>Erstellen Sie eine Klasse <code>Akku</code>.</p> <p>Objektvariablen sind</p> <ul style="list-style-type: none"> <li>• <code>typ</code> vom Typ <code>String</code> (<i>Typ des Akkus</i>),</li> <li>• <code>kapazitaet</code> vom Typ <code>int</code> (<i>Akku-Kapazität</i>),</li> <li>• <code>anzahlLadungen</code> vom Typ <code>int</code> (<i>Anzahl der bisherigen Ladungen des Akkus</i>).</li> </ul> <p>Die Objektvariablen sind nur innerhalb der Klasse sichtbar!</p>	<p>1 Pkt.</p>
---	---------------

<p>Erstellen Sie für die Klasse <b>Akku</b> einen parametrisierten Konstruktor, dem die Parameter</p> <ul style="list-style-type: none"> <li>- <b>String typ</b> und</li> <li>- <b>int kapazitaet</b></li> </ul> <p>übergeben werden. Mit den Parameterwerten werden die entsprechenden Objektvariablen initialisiert. Die Objektvariable <b>anzahlLadungen</b> wird mit dem Wert <b>0</b> initialisiert.</p>	<p>1 Pkt.</p>
<p>Erstellen Sie eine Methode <b>laden ()</b>, die den Wert der Objektvariablen <b>anzahlLadungen</b> um <b>1</b> erhöht.</p>	<p>1 Pkt.</p>
<p>Erstellen Sie eine Methode <b>getZustand ()</b>, die ein <b>int</b> zurückgibt. Der Zustand berechnet sich wie folgt:</p> <ul style="list-style-type: none"> <li>- Ist der Akku neu, ist der Zustand bei <b>100%</b>.</li> <li>- Durch alle 100 Ladungen verringert sich der Zustand um <b>1%</b>.</li> <li>- Der Zustand kann nicht kleiner als <b>0%</b> werden.</li> </ul> <p>Sie müssen also die <b>anzahlLadungen</b> betrachten und durch alle <b>100</b> Ladungen reduziert sich der Zustands-Wert <b>100</b> um <b>1</b>.</p> <p>Beispiele:</p> <ul style="list-style-type: none"> <li>- <b>anzahlLadungen: 99</b> → Zustand 100</li> <li>- <b>anzahlLadungen: 199</b> → Zustand 99</li> <li>- <b>anzahlLadungen: 1099</b> → Zustand 90</li> <li>- <b>anzahlLadungen: 10099</b> → Zustand 0</li> </ul> <p><b>Tipps:</b></p> <ul style="list-style-type: none"> <li>- <b>Nutzen Sie die Integer-Division durch 100, um zu ermitteln, welchen Wert Sie von 100 abziehen müssen.</b></li> <li>- <b>Prüfen Sie, ob das Ergebnis unter 0 ist, dann geben Sie 0 zurück.</b></li> </ul>	<p>3 Pkt.</p>
<p><u>Überschreiben</u> Sie die Methode <b>toString ()</b> so, dass ein <u>formatierter String</u> in der folgenden Form zurückgegeben wird (Beispielwerte):</p> <p><b>A1 : 10000mAh 984 Ladungen. Zustand 91%</b></p> <p>Beachten Sie: „A1“ ist der Typ des Akkus.</p> <ul style="list-style-type: none"> <li>- Reservieren Sie für den Typ <b>4</b> Zeichen (dann kommt ein Doppelpunkt),</li> <li>- reservieren Sie für die Kapazität <b>5</b> Zeichen (dann kommt „mAh“),</li> <li>- reservieren Sie für die Anzahl der Ladungen <b>5</b> Zeichen (dann kommt „Ladungen. Zustand “),</li> <li>- reservieren Sie für den Zustand <b>3</b> Zeichen (dann kommt „%“ → siehe <b>Tipp</b>).</li> </ul>	<p>3 Pkt.</p>

<p><b>Tip:</b> das „%“-Zeichen in einem formatierten String wird durch „%%“ erzeugt.</p> <p>Erstellen Sie eine <code>print()</code>-Methode, die den von <code>toString()</code> erzeugten String auf die Konsole ausgibt.</p>	
<p>Überschreiben Sie die Methode <code>equals(Object o)</code> so, dass zwei Akkus gleich sind, wenn sie denselben Typ haben.</p>	2 Pkt.
<p>Erstellen Sie eine <b>Testklasse</b> mit <code>main()</code>-Methode. Erzeugen Sie in der <code>main()</code>-Methode vier Akku-Objekte mit den folgenden Werten für die Objektvariablen:</p> <pre>"A1", 10000 "A2", 20000 "A3", 30000 "A4", 40000</pre> <p>Geben Sie alle drei Buch-Objekte auf die Konsole unter Verwendung der <code>print()</code>-Methode aus. Es entsteht folgende Ausgabe:</p> <pre>A1 : 10000mAh      0 Ladungen. Zustand 100% A2 : 20000mAh      0 Ladungen. Zustand 100% A3 : 30000mAh      0 Ladungen. Zustand 100% A4 : 40000mAh      0 Ladungen. Zustand 100%</pre>	2 Pkt.
<p>Erweitern Sie die <code>main()</code>-Methode der <b>Testklasse</b> um eine Schleife, um 4000 Ladungen durchzuführen. Innerhalb dieser Schleife wird eine Zufallszahl aus dem Bereich <code>[0, ..., 3]</code> (3 inkl.) erzeugt. Nutzen Sie dazu die Klasse <b>Random</b> aus dem <code>java.util</code>-Paket.</p> <ul style="list-style-type: none"> <li>- Ist die Zufallszahl 0, wird der erste Akku (<b>A1</b>) geladen.</li> <li>- Ist die Zufallszahl 1, wird der zweite Akku (<b>A2</b>) geladen.</li> <li>- Ist die Zufallszahl 2, wird der dritte Akku (<b>A3</b>) geladen.</li> <li>- Ist die Zufallszahl 3, wird der vierte Akku (<b>A4</b>) geladen</li> </ul> <p>Rufen Sie für die vier Akkus jeweils die <code>print()</code>-Methode auf. Es entsteht folgende Ausgabe (Zufallswerte):</p> <pre>A1 : 10000mAh      984 Ladungen. Zustand  91% A2 : 20000mAh      945 Ladungen. Zustand  91% A3 : 30000mAh     1032 Ladungen. Zustand  90% A4 : 40000mAh     1039 Ladungen. Zustand  90%</pre>	4 Pkt.

**Teil 2 (Klasse AkkuBestand – Teil 1)****19 Punkte**

<p>Erstellen Sie eine Klasse <b>AkkuBestand</b>.</p> <p>Objektvariable ist</p>	1 Pkt.
--	--------

<ul style="list-style-type: none"> <li>• <b>akkus</b> vom Typ <b>Akku []</b>.</li> </ul> <p>Die Objektvariable ist nur innerhalb der Klasse sichtbar!</p> <p>Erstellen Sie einen parameterlosen Konstruktor. In dem Konstruktor wird das <b>akkus</b>-Array mit der Länge 0 erzeugt.</p>	
<p>Erstellen Sie eine Methode <b>akkuBereitsImBestand (Akku a)</b> . Diese Methode gibt ein <b>true</b> zurück, wenn <b>a</b> bereits in <b>akkus</b> enthalten ist und <b>false</b> sonst.</p>	1 Pkt.
<p>Erstellen Sie eine Methode <b>akkuHinzufuegen (Akku a)</b> . Diese Methode fügt den Akku <b>a</b> dem <b>akkus</b>-Array hinzu.</p> <p>Die Methode gibt ein <b>true</b> zurück, wenn <b>a</b> dem <b>akkus</b>-Array hinzugefügt wurde und ein <b>false</b>, wenn nicht.</p> <p>Der Akku <b>a</b> wird genau dann <b>nicht</b> dem <b>akkus</b>-Array hinzugefügt, wenn er bereits im <b>akkus</b>-Array enthalten ist!</p> <p>Tipps:</p> <ul style="list-style-type: none"> <li>- Nutzen Sie die Methode <b>akkuBereitsImBestand (Akku a)</b> , um zu ermitteln, ob <b>a</b> bereits in <b>akkus</b> enthalten ist.</li> <li>- Wenn <b>a</b> hinzugefügt wird, muss das <b>akkus</b>-Array um <b>1</b> größer werden.</li> </ul>	4 Pkt.
<p>Erstellen Sie eine Methode <b>akkuEntfernen (Akku a)</b> . Diese Methode entfernt den Akku <b>a</b> aus dem <b>akkus</b>-Array.</p> <p>Die Methode gibt ein <b>true</b> zurück, wenn <b>a</b> aus dem <b>akkus</b>-Array entfernt wurde und ein <b>false</b>, wenn nicht.</p> <p>Der Akku <b>a</b> wird genau dann <b>nicht</b> aus dem <b>akkus</b>-Array entfernt, wenn er gar nicht im <b>akkus</b>-Array enthalten ist!</p> <p>Tipps:</p> <ul style="list-style-type: none"> <li>- Nutzen Sie die Methode <b>akkuBereitsImBestand (Akku a)</b> , um zu ermitteln, ob <b>a</b> überhaupt in <b>akkus</b> enthalten ist.</li> <li>- Wenn <b>a</b> entfernt wurde, muss das <b>akkus</b>-Array danach um <b>1</b> kleiner sein.</li> </ul>	3 Pkt.
<p>Erstellen Sie eine Methode <b>getLength ()</b> , die die Länge des <b>akkus</b>-Arrays zurückgibt.</p>	1 Pkt.
<p><u>Überschreiben</u> Sie die Methode <b>toString ()</b> so, dass ein <b>String</b> in der folgenden Form zurückgegeben wird (Beispielwerte):</p> <p><b>Bestand (3):</b></p> <p><b>A1 : 10000mAh 984 Ladungen. Zustand 91%</b>  <b>A2 : 20000mAh 945 Ladungen. Zustand 91%</b></p>	3 Pkt.

<p><b>A3 : 3000mAh 1032 Ladungen. Zustand 90%</b></p> <p>Die Zahl in Klammern rechts von „Bestand“ gibt die Länge des <b>akkus</b>-Arrays an (hier <b>(3)</b>).</p> <p>Beachten Sie! Enthält das <b>akkus</b>-Array <i>keine</i> Akkus, dann soll folgender <b>String</b> zurückgegeben werden:</p> <p><b>Derzeit sind keine Akkus im Bestand.</b></p> <p>Erstellen Sie eine <b>print()</b>-Methode, die den von <b>toString()</b> erzeugten <b>String</b> auf die Konsole ausgibt.</p>	
<p>Erzeugen Sie in der <b>main()</b>-Methode der <b>Testklasse</b> ein Objekt von <b>AkkuBestand</b>.</p> <p>Geben Sie das <b>AkkuBestand</b>-Objekt unter Verwendung der <b>print()</b>-Methode auf die Konsole aus. Es entsteht folgende Ausgabe:</p> <p><b>Derzeit sind keine Akkus im Bestand.</b></p>	2 Pkt.
<p>1. Fügen Sie den Akku „A1“ aus Teil 1 dem <b>AkkuBestand</b>-Objekt hinzu. Beachten Sie dabei die Rückgabe der <b>akkuHinzufuegen()</b>-Methode. Wird der Akku hinzugefügt, soll die Ausgabe</p> <p><b>Akku hinzugefuegt!</b></p> <p>auf der Konsole erscheinen. Wird er nicht hinzugefügt, erscheint</p> <p><b>Akku nicht hinzugefuegt!</b></p> <p>2. Rufen Sie die <b>print()</b>-Methode für das <b>AkkuBestand</b>-Objekt auf. Es erscheint:</p> <p><b>Bestand (1):</b></p> <p><b>A1 : 1000mAh 984 Ladungen. Zustand 91%</b></p> <p>3. Wiederholen Sie die Schritte 1 und 2 für</p> <ul style="list-style-type: none"> <li>○ Akku „A2“ aus Teil 1 (wird tatsächlich hinzugefügt),</li> <li>○ Akku „A3“ aus Teil 1 (wird tatsächlich hinzugefügt) und</li> <li>○ nochmal Akku „A3“ aus Teil 1 (wird nicht hinzugefügt)</li> </ul> <p>Insgesamt entsteht folgende Ausgabe:</p> <p><b>Akku hinzugefuegt!</b> <b>Bestand (1):</b></p> <p><b>A1 : 1000mAh 984 Ladungen. Zustand 91%</b></p> <p><b>Akku hinzugefuegt!</b></p>	3 Pkt.

<p><b>Bestand (2):</b></p> <p>A1 : 10000mAh 984 Ladungen. Zustand 91%          A2 : 20000mAh 945 Ladungen. Zustand 91%</p> <p><b>Akku hinzugefüegt!</b>  <b>Bestand (3):</b></p> <p>A1 : 10000mAh 984 Ladungen. Zustand 91%          A2 : 20000mAh 945 Ladungen. Zustand 91%          A3 : 30000mAh 1032 Ladungen. Zustand 90%</p> <p><b>Akku nicht hinzugefüegt!</b>  <b>Bestand (3):</b></p> <p>A1 : 10000mAh 984 Ladungen. Zustand 91%          A2 : 20000mAh 945 Ladungen. Zustand 91%          A3 : 30000mAh 1032 Ladungen. Zustand 90%</p>	
<p>Entfernen Sie mithilfe der <code>akkuEntfernen()</code>-Methode den Akku „A2“ zwei Mal (einmal wird er tatsächlich entfernt und einmal nicht) und berücksichtigen Sie die Rückgabe der Methode, so dass folgende Ausgaben entstehen (jeweils auch <code>print()</code>-Methode von <code>AkkuBestand</code> aufrufen) :</p> <p><b>Akku entfernt!</b>  <b>Bestand (2):</b></p> <p>A1 : 10000mAh 984 Ladungen. Zustand 91%          A3 : 30000mAh 1032 Ladungen. Zustand 90%</p> <p><b>Akku nicht entfernt!</b>  <b>Bestand (2):</b></p> <p>A1 : 10000mAh 984 Ladungen. Zustand 91%          A3 : 30000mAh 1032 Ladungen. Zustand 90%</p>	1 Pkt.

**Teil 3 (Klasse AkkuBestand – Teil 2)****27 Punkte**

<p>Erstellen Sie einen weiteren Konstruktor in der <code>AkkuBestand</code>-Klasse. Diesem Konstruktor soll ein <code>AkkuBestand</code>-Objekt als Parameter übergeben werden. Das <code>akkus</code>-Array des neu zu erstellenden Objektes soll alle <code>Akku</code>-Objekte des als Parameter übergebenen <code>AkkuBestand</code>-Objektes aufnehmen.</p>	3 Pkt.
<p>Testen Sie in der <code>main()</code>-Methode der <code>Testklasse</code> den neuen Konstruktor, indem Sie ein neues <code>AkkuBestand</code>-Objekt erzeugen und dem Konstruktor den „alten“ <code>AkkuBestand</code> aus Teil 2 übergeben. Rufen Sie für das neue <code>AkkuBestand</code>-Objekt die <code>print()</code>-Methode auf. Es sollte erscheinen (Zufallswerte):</p> <p><b>Bestand (2):</b></p>	2 Pkt.

<p><b>A1 : 10000mAh 984 Ladungen. Zustand 91%</b>  <b>A3 : 30000mAh 1032 Ladungen. Zustand 90%</b></p> <p>Fügen Sie nun dem neuen <b>AkkuBestand</b>-Objekt auch noch die <b>Akku</b>-Objekte „A2“ und „A4“ hinzu und rufen Sie die <b>print ()</b>-Methode erneut auf. Es sollte erscheinen (Zufallswerte):</p> <p><b>Bestand (4):</b></p> <p><b>A1 : 10000mAh 984 Ladungen. Zustand 91%</b>  <b>A3 : 30000mAh 1032 Ladungen. Zustand 90%</b>  <b>A2 : 20000mAh 945 Ladungen. Zustand 91%</b>  <b>A4 : 40000mAh 1039 Ladungen. Zustand 90%</b></p>	
<p>Erstellen Sie in der Klasse <b>AkkuBestand</b> eine Methode <b>fehlendeAkkus (AkkuBestand ab)</b>. Diese Methode gibt ein <b>Akku</b>-Array zurück. Das zurückgegebene <b>Akku</b>-Array enthält alle <b>Akku</b>-Objekte, die im <b>akkus</b>-Array von <b>ab</b>, nicht aber im <b>akkus</b>-Array des aufrufenden <b>AkkuBestand</b>-Objektes enthalten sind.</p>	5 Pkt.
<p>Testen Sie die <b>fehlendeAkkus ()</b>-Methode in der <b>main ()</b>-Methode der <b>Testklasse</b>, indem Sie sie für das <b>AkkuBestand</b>-Objekt aus Teil 2 aufrufen und das neue <b>AkkuBestand</b>-Objekt als Parameter übergeben, welches alle <b>Akku</b>-Typen enthält. Geben Sie die Akkus des zurückgegebenen <b>Akku</b>-Arrays aus (direkt in der <b>main ()</b>-Methode). Es entsteht folgende Ausgabe (Zufallswerte):</p> <p><b>Fehlende :</b></p> <p><b>A2 : 20000mAh 945 Ladungen. Zustand 91%</b>  <b>A4 : 40000mAh 1039 Ladungen. Zustand 90%</b></p>	1 Pkt.
<p>Erstellen Sie in der Klasse <b>AkkuBestand</b> eine Methode <b>schlechtesterZustand ()</b>. Diese Methode gibt das (erste) <b>Akku</b>-Objekt zurück, das im <b>akkus</b>-Array den kleinsten Zustands-Wert hat. Sie können davon ausgehen, dass mindestens ein <b>Akku</b> im <b>akkus</b>-Array ist.</p> <p>Testen Sie die Methode in der <b>main()</b>-Methode, z.B.</p> <p><b>Akku mit schlechtestem Zustand :</b>  <b>A3 : 30000mAh 1032 Ladungen. Zustand 90%</b></p>	2 Pkt.
<p>Erstellen Sie in der Klasse <b>AkkuBestand</b> eine Methode <b>getAkkuAtIndex (int index)</b>. Diese Methode gibt den <b>Akku</b> zurück, der im <b>akkus</b>-Array unter dem Index <b>index</b> gespeichert ist.</p> <p>Ist <b>index</b> kein gültiger Index aus dem <b>akkus</b>-Array, dann geben Sie <b>null</b> zurück.</p>	3 Pkt.
<p>Testen Sie die <b>getAkkuAtIndex (int index)</b> wie folgt:</p> <p>- Falls Sie ungültigen Index übergeben, dann soll die Ausgabe so sein:</p>	1 Pkt.

<p><b>Kein gueltiger Index!</b></p> <ul style="list-style-type: none"> <li>- Falls Sie einen korrekten Index übergeben, wird das zurückgegebene Akku ausgegeben, z.B.</li> </ul> <p><b>A2 : 20000mAh 995 Ladungen. Zustand 91%</b></p>	
<p>Erstellen Sie in der Klasse <b>AkkuBestand</b> eine Methode <b>akkusMitSchlechtemZustand()</b>. Diese Methode gibt ein <b>Akku</b>-Array zurück. Das zurückgegebene <b>Akku</b>-Array enthält alle <b>Akku</b>-Objekte des <b>akkus</b>-Arrays, deren Zustand unter 10% ist.</p>	5 Pkt.
<p>Testen Sie in der <b>main()</b>-Methode der Testklasse die <b>akkusMitSchlechtemZustand()</b>-Methode wie folgt:</p> <ul style="list-style-type: none"> <li>- in einer Schleife <b>laden</b> Sie alle Akkus aus dem <b>AkkuBestand</b>-Objekt, welches alle <b>Akku</b>-Typen enthält (haben Sie hier in Teil 3 erstellt)</li> <li>- Sie laden <b>alle</b> Akkus solange, bis <b>alle</b> Akkus in dem zurückgegebenen Array der <b>akkusMitSchlechtemZustand()</b> enthalten sind (reicht zu prüfen, dass das zurückgegebene Array die Länge 4 hat). Geben Sie dann dieses Array aus. Es erscheint z.B. (Zufallswerte):</li> </ul> <p><b>Bestand (4):</b></p> <p><b>A1 : 10000mAh 9143 Ladungen. Zustand 9%</b>  <b>A3 : 30000mAh 9158 Ladungen. Zustand 9%</b>  <b>A2 : 20000mAh 9127 Ladungen. Zustand 9%</b>  <b>A4 : 40000mAh 9100 Ladungen. Zustand 9%</b></p>	5 Pkt.

Zur Kontrolle: Die möglichen Ausgaben (Beispielwerte) könnten sein:

----- Teil 1 - Akku: Objekte erzeugen -----

**A1 : 10000mAh 0 Ladungen. Zustand 100%**  
**A2 : 20000mAh 0 Ladungen. Zustand 100%**  
**A3 : 30000mAh 0 Ladungen. Zustand 100%**  
**A4 : 40000mAh 0 Ladungen. Zustand 100%**

----- Teil 2 - Akku: 4000 Ladungen -----

**A1 : 10000mAh 1011 Ladungen. Zustand 90%**  
**A2 : 20000mAh 995 Ladungen. Zustand 91%**  
**A3 : 30000mAh 1026 Ladungen. Zustand 90%**  
**A4 : 40000mAh 968 Ladungen. Zustand 91%**

----- Teil 3 - AkkuBestand -----

Derzeit sind keine Akkus im Bestand.

Akku hinzugefuegt!  
Bestand (1):

**A1 : 10000mAh 1011 Ladungen. Zustand 90%**

Akku hinzugefuegt!  
Bestand (2):



A1 : 10000mAh 1011 Ladungen. Zustand 90%  
A2 : 20000mAh 995 Ladungen. Zustand 91%

**Akku hinzugefügt!**  
Bestand (3):

A1 : 10000mAh 1011 Ladungen. Zustand 90%  
A2 : 20000mAh 995 Ladungen. Zustand 91%  
A3 : 30000mAh 1026 Ladungen. Zustand 90%

**Akku nicht hinzugefügt!**  
Bestand (3):

A1 : 10000mAh 1011 Ladungen. Zustand 90%  
A2 : 20000mAh 995 Ladungen. Zustand 91%  
A3 : 30000mAh 1026 Ladungen. Zustand 90%

**Akku entfernt!**  
Bestand (2):

A1 : 10000mAh 1011 Ladungen. Zustand 90%  
A3 : 30000mAh 1026 Ladungen. Zustand 90%

**Akku nicht entfernt!**  
Bestand (2):

A1 : 10000mAh 1011 Ladungen. Zustand 90%  
A3 : 30000mAh 1026 Ladungen. Zustand 90%

----- Teil 4 - AkkuBestand -----

Bestand (2):

A1 : 10000mAh 1011 Ladungen. Zustand 90%  
A3 : 30000mAh 1026 Ladungen. Zustand 90%

Bestand (4):

A1 : 10000mAh 1011 Ladungen. Zustand 90%  
A3 : 30000mAh 1026 Ladungen. Zustand 90%  
A2 : 20000mAh 995 Ladungen. Zustand 91%  
A4 : 40000mAh 968 Ladungen. Zustand 91%

Fehlende :

A2 : 20000mAh 995 Ladungen. Zustand 91%  
A4 : 40000mAh 968 Ladungen. Zustand 91%

Kein gueltiger Index!

A2 : 20000mAh 995 Ladungen. Zustand 91%

Akku mit schlechtestem Zustand :

A1 : 10000mAh 1011 Ladungen. Zustand 90%

Bestand (4):

A1 : 10000mAh 9143 Ladungen. Zustand 9%  
A3 : 30000mAh 9158 Ladungen. Zustand 9%  
A2 : 20000mAh 9127 Ladungen. Zustand 9%  
A4 : 40000mAh 9100 Ladungen. Zustand 9%

**Viel Erfolg!**