

KLAUSUR
PROGRAMMIERUNG 1
30. MÄRZ 2021

Bedingungen der Klausur:

1. Fragen stellen Sie am besten 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 Klausur2)!
4. Es sind insgesamt 80 Punkte zu erzielen (Teil 1: 20 Pkt., Testklasse: 8 Pkt., Teil 2: 19 Pkt., Testklasse: 7 Pkt., Teil 3: 17 Pkt., Testklasse 5 Pkt., fehlerfreies Programm: 4 Pkt.).
5. Schreiben Sie Ihre Klassen im package **klausur**!
6. Die Verwendungen der Klassen **Arrays** und **Math** sind nicht gestattet (und nicht nötig).

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	>=	75.5	71.5	67.5	63.5	59.5	55.5	51.5	47.5	43.5	<
	76.0	— 72.0	— 68.0	— 64.0	— 60.0	— 56,0	— 52.0	— 48.0	— 44.0	— 40.0	40

Teil 1 (Klasse MyString)

20 Punkte

Information:

- Wir bauen in Ansätzen die Klasse **String** nach, d.h. wir bauen uns einen eigenen Datentypen für Zeichenketten: **MyString**.
- Ein Objekt von **MyString** ist intern ein `char[]` (wir nennen es **wort**).
- Von der Klasse **String** selbst verwenden wir nur zwei Methoden: `charAt(index)` und `length()`, und zwar in dem Konstruktor von **MyString**, dem ein **String** als Parameter übergeben wird → das steht aber nochmal explizit dort

<ul style="list-style-type: none">Ansonsten verwenden wir von String keine weiteren Methoden (ist nicht erlaubt, wäre aber auch nirgendwo hilfreich)Wir erstellen insgesamt nur 2 Klassen: MyString und Testklasse. Das ist aber kein Nachteil! Trotzdem ist die Klausur in 3 Teile unterteilt, wobei es von Teil zu Teil immer etwas schwieriger wird, aber das ist ja immer so. <p>Viel Erfolg!</p>						
Erstellen Sie eine Klasse MyString . Objektvariable ist <ul style="list-style-type: none">wort vom Typ char[] (unsere Zeichenkette – Array von Zeichen). Die Objektvariable ist nur innerhalb der Klasse sichtbar!	2 Pkt.					
Schreiben Sie für MyString einen parameterlosen Konstruktor. In diesem Konstruktor wird das wort -Array erzeugt mit der Länge 0.	1 Pkt.					
Schreiben Sie für MyString einen parametrisierten Konstruktor, dem ein einzelnes Zeichen (char) als Parameter übergeben wird. In diesem Konstruktor wird das wort -Array erzeugt mit der Länge 1 und das einzelne Zeichen wird in das wort -Array gespeichert.	2 Pkt.					
Schreiben Sie für MyString einen parametrisierten Konstruktor, dem ein String als Parameter übergeben wird. In diesem Konstruktor wird das wort -Array erzeugt mit der Länge des Strings und jedes einzelne Zeichen des Strings wird in das wort -Array gespeichert. <ul style="list-style-type: none">Hierzu nutzen Sie die Methoden charAt(index) und length() der Klasse StringAnsonsten benötigen Sie für den Rest der Klausur keine Methoden mehr aus String	3 Pkt.					
<u>Überschreiben</u> Sie die Objektmethode toString() so, dass alle Zeichen aus dem wort -Array als ein String zurückgegeben werden, also z.B. <table><tr><td>'h'</td><td>'a'</td><td>'l'</td><td>'l'</td><td>'o'</td></tr></table> → "hallo"	'h'	'a'	'l'	'l'	'o'	2 Pkt.
'h'	'a'	'l'	'l'	'o'		
Schreiben Sie eine Objektmethode print() . Diese Methode gibt den durch toString() erzeugten String auf der Konsole aus.	1 Pkt.					
Schreiben Sie die Objektmethode getLength() . Diese Methode gibt die Länge des wort -Arrays zurück.	1 Pkt.					
Schreiben Sie die Objektmethode getCharAt(int index) . Diese Methode gibt ein Leerzeichen zurück, wenn der übergebene index kein Index des wort -Arrays	2 Pkt.					

ist. Ansonsten wird das Zeichen zurückgegeben, das im wort -Array unter dem Index index gespeichert ist.	
Schreiben Sie die Objektmethode contains(char c) . Diese Methode gibt ein true zurück, wenn c im wort -Array vorkommt. Ansonsten false .	2 Pkt.
Überschreiben Sie die Objektmethode equals(Object o) . Diese Methode gibt ein true zurück, wenn das aufrufende Objekt die gleiche wort -Länge hat wie o und auch alle Zeichen in den beiden wort -Arrays elementweise gleich sind (also Index für Index). Sonst false .	4 Pkt.

Testklasse

8 Punkte

<p>Schreiben Sie eine Klasse Testklasse mit main()-Methode. Erzeugen Sie in der main()-Methode</p> <ul style="list-style-type: none"> - ein Objekt s1 von MyString mithilfe des parameterlosen Konstruktors, - ein Objekt s2 von MyString, dem der einzelne Buchstabe a übergeben wird, - ein Objekt s3 von MyString, dem die Zeichenkette hallo übergeben wird, - rufen Sie für alle drei Objekte die print()-Methode auf. Es entsteht folgende Ausgabe: <pre>--- // das hier gehört nicht zur Ausgabe, soll nur zeigen, dass zunächst eine Leerzeile kommt</pre> <pre>a hallo</pre>	3 Pkt.
<p>Wenden Sie die getCharAt()-Methode an und lassen sich für das dritte MyString-Objekt (s3) den Buchstaben auf der Konsole ausgeben, der am Index 1 gespeichert ist.</p> <p>Wenden Sie die contains()-Methode 2 Mal an und lassen sich für das dritte MyString-Objekt auf der Konsole ausgeben,</p> <ul style="list-style-type: none"> - ob der Buchstabe a in dessen wort-Array gespeichert ist und - ob der Buchstabe b in dessen wort-Array gespeichert ist. <p>Es entsteht folgende Ausgabe:</p> <pre>a a in hallo ? true b in hallo ? false</pre>	3 Pkt.
Erzeugen Sie zwei weitere Objekte s4 und s5 von MyString , einmal mit der Zeichenkette hallo und einmal mit der Zeichenkette halLo und vergleichen Sie	2 Pkt.

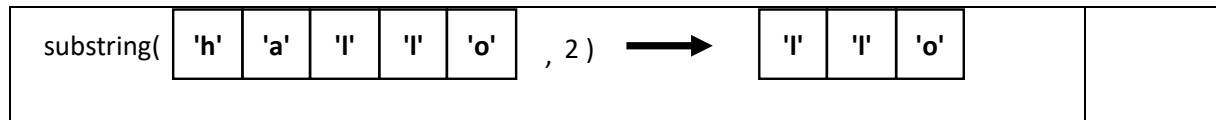
beide Objekte mithilfe der `equals()`-Methode mit dem dritten **MyString**-Objekt **s3**. Es entstehen folgende Ausgaben:

```
s3 gleich s4 ? true
s3 gleich s4 ? false
```

Teil 2 (weiter Klasse MyString)

19 Punkte

Schreiben Sie eine Objektmethode <code>getAsciiCodeOfCharAt(int index)</code> . Diese Methode gibt den Ascii-Code des Zeichens zurück, das im wort -Array unter dem Index index gespeichert ist. Ist index kein Index aus dem wort -Array, wird -1 zurückgegeben.	2 Pkt.
Schreiben Sie eine Objektmethode <code>append(char c)</code> . Diese Methode hängt das Zeichen c an das wort -Array, d.h. danach ist das wort -Array um 1 länger als vor Aufruf der Methode.	4 Pkt.
Schreiben Sie eine Objektmethode <code>append(MyString s)</code> . Diese Methode hängt alle Zeichen aus s (aus dessen wort -Array) an das wort -Array des aufrufenden Objektes. Tipp: Verwenden Sie dazu die <code>append(char c)</code> -Methode.	2 Pkt.
Schreiben Sie eine Objektmethode <code>concat(MyString s1, MyString s2)</code> . Diese Methode gibt ein (neues) MyString -Objekt zurück, in dem das wort -Array alle Zeichen aus s1 und s2 enthält. (Diese <code>concat()</code> -Methode entspricht s1+s2 für Strings) <div style="display: flex; align-items: center;"> <div style="margin-right: 10px;">concat(</div> <div style="display: flex; align-items: center;"> <div style="border: 1px solid black; padding: 2px; margin: 0 5px;">'h'</div> <div style="border: 1px solid black; padding: 2px; margin: 0 5px;">'a'</div> <div style="border: 1px solid black; padding: 2px; margin: 0 5px;">' '</div> <div style="border: 1px solid black; padding: 2px; margin: 0 5px;">' '</div> <div style="border: 1px solid black; padding: 2px; margin: 0 5px;">'o'</div> </div> <div style="margin: 0 10px;">,</div> <div style="display: flex; align-items: center;"> <div style="border: 1px solid black; padding: 2px; margin: 0 5px;">'b'</div> <div style="border: 1px solid black; padding: 2px; margin: 0 5px;">'a'</div> <div style="border: 1px solid black; padding: 2px; margin: 0 5px;">' '</div> <div style="border: 1px solid black; padding: 2px; margin: 0 5px;">' '</div> <div style="border: 1px solid black; padding: 2px; margin: 0 5px;">'o'</div> </div> </div> <div style="margin-left: 10px;">)</div> <div style="display: flex; align-items: center;"> <div style="margin-right: 10px;">→</div> <div style="border: 1px solid black; padding: 2px; display: flex; gap: 5px;"> <div>'h'</div><div>'a'</div><div>' '</div><div>' '</div><div>'o'</div><div>'b'</div><div>'a'</div><div>' '</div><div>' '</div><div>'o'</div> </div> </div>	3 Pkt.
Schreiben Sie die Objektmethode <code>firstIndexOf(char c)</code> . Diese Methode gibt den Index zurück, an dem das Zeichen c zum <i>ersten</i> Mal im wort -Array vorkommt. Kommt das Zeichen c nicht im wort -Array vor, wird -1 zurückgegeben.	2 Pkt.
Schreiben Sie die Objektmethode <code>lastIndexOf(char c)</code> . Diese Methode gibt den Index zurück, an dem das Zeichen c zum <i>letzten</i> Mal im wort -Array vorkommt. Kommt das Zeichen c nicht im wort -Array vor, wird -1 zurückgegeben.	2 Pkt.
Schreiben Sie eine Objektmethode <code>substring(int beginIndex)</code> . Diese Methode gibt ein (neues) MyString -Objekt zurück. Das neue MyString -Objekt enthält im wort -Array alle Zeichen aus dem wort -Array des aufrufenden Objektes ab dem Index beginIndex (inklusive). Ist beginIndex kein korrekter Index aus dem wort -Array des aufrufenden Objektes, wird ein MyString -Objekt mit leerem wort -Array zurückgegeben.	4 Pkt.

**Testklasse****7 Punkte**

Testen Sie die getAsciiCodeOfCharAt() -Methode, indem Sie sich den Ascii-Code des Zeichens von s3 auf dem Index 1 ausgeben lassen: Ascii-Code Zeichen aus s3 mit Index 1 = 97	1 Pkt.
Testen Sie die append(char) -Methode, indem Sie an s3 das Zeichen c hängen: halloc	1 Pkt.
Testen Sie die append(MyString) -Methode, indem Sie sich ein weiteres MyString -Objekt s6 mit ballo erzeugen und dieses Objekt an s3 hängen: hallocballo	1 Pkt.
Testen Sie die concat() -Methode, indem Sie sich ein weiteres MyString -Objekt s7 mit leerem wort -Array erzeugen und für dieses Objekt die concat() -Methode aufrufen, wobei Sie die Objekte s4 und s6 übergeben: halloballo	1 Pkt.
Testen Sie die firstIndexOf() - und die lastIndexOf() -Methode, indem Sie jeweils in s3 nach dem Buchstaben l suchen (kleines l): erstes l in s3 am Index 2 letztes l in s3 am Index 9	2 Pkt.
Testen Sie die substring() -Methode, indem Sie den Teilstring ab Index 5 für s3 ausgeben: cballo	1 Pkt.

Teil 3 (weiter Klasse MyString)**17 Punkte**

Schreiben Sie eine Objektmethode substring(int beginIndex, int endIndex) . Diese Methode gibt ein (neues) MyString -Objekt zurück. Das neue MyString -Objekt enthält im wort -Array alle Zeichen aus dem wort -Array des aufrufenden Objektes ab dem Index beginIndex (inklusive) bis zum Index endIndex (exklusive). Ist beginIndex kein korrekter Index aus dem wort -Array	3 Pkt.
--	--------

<p>des aufrufenden Objektes oder ist beginIndex größer gleich endIndex, wird ein MyString-Objekt mit leerem wort-Array zurückgegeben .</p> <p>substring(<table border="1"><tr><td>'h'</td><td>'a'</td><td>'l'</td><td>'l'</td><td>'o'</td></tr></table> , 2, 4) \longrightarrow <table border="1"><tr><td>'l'</td><td>'l'</td></tr></table></p>	'h'	'a'	'l'	'l'	'o'	'l'	'l'														
'h'	'a'	'l'	'l'	'o'																	
'l'	'l'																				
<p>Schreiben Sie eine Objektmethode toUpperCase() , die ein (neues) MyString-Objekt zurückgibt. Alle Kleinbuchstaben aus dem aufrufenden Objekt sind im neuen Objekt Großbuchstaben.</p> <p><table border="1"><tr><td>'h'</td><td>'a'</td><td>'3'</td><td>'4'</td><td>'!'</td><td>'b'</td><td>'a'</td><td>'?'</td><td>'.'</td><td>'o'</td></tr></table></p> <p>\longrightarrow <table border="1"><tr><td>'H'</td><td>'A'</td><td>'3'</td><td>'4'</td><td>'!'</td><td>'B'</td><td>'A'</td><td>'?'</td><td>'.'</td><td>'O'</td></tr></table></p> <p><i>Tipps:</i></p> <ul style="list-style-type: none">- Bei der Umwandlung müssen Sie sich nur um die Kleinbuchstaben kümmern, alle anderen Zeichen werden direkt übernommen.- Kleinbuchstaben haben die Ascii-Codes 97 (a) – 122 (z)- Großbuchstaben haben die Ascii-Codes 65 (A) – 90 (Z)	'h'	'a'	'3'	'4'	'!'	'b'	'a'	'?'	'.'	'o'	'H'	'A'	'3'	'4'	'!'	'B'	'A'	'?'	'.'	'O'	4 Pkt.
'h'	'a'	'3'	'4'	'!'	'b'	'a'	'?'	'.'	'o'												
'H'	'A'	'3'	'4'	'!'	'B'	'A'	'?'	'.'	'O'												
<p>Schreiben Sie eine Objektmethode reverse() , die ein (neues) MyString-Objekt zurückgibt. In diesem neuen Objekt ist die Zeichenfolge im wort-Array genau umgekehrt wie im aufrufenden Objekt.</p> <p><table border="1"><tr><td>'h'</td><td>'a'</td><td>'l'</td><td>'l'</td><td>'o'</td></tr></table> \longrightarrow <table border="1"><tr><td>'o'</td><td>'l'</td><td>'l'</td><td>'a'</td><td>'h'</td></tr></table></p>	'h'	'a'	'l'	'l'	'o'	'o'	'l'	'l'	'a'	'h'	3 Pkt.										
'h'	'a'	'l'	'l'	'o'																	
'o'	'l'	'l'	'a'	'h'																	
<p>Schreiben Sie die Objektmethode contains(MyString s) . Diese Methode gibt ein true zurück, wenn das wort-Array von s im wort-Array des aufrufenden Objektes vorkommt. Ansonsten false.</p>	7 Pkt.																				

Testklasse

5 Punkte

<p>Testen Sie alle 4 Methoden aus Teil 3.</p> <p>----- Teil 3 -----</p> <pre> s3 : halloballo s3 substring(3,7) : locb s3 toUpperCase() : HALLOCBALLO s3 reverse() : ollabcollah s4 : locba s4 in s3 ? : true </pre>	5 Pkt.
---	--------