

Einführung in die Programmierung  
WS 2018/19

Übungsblatt 8: Arrays, Strings

**Aufgabe 8-1**     *Arrays*

Diese Aufgabe beschäftigt sich mit Arrays und einigen ihrer Grundoperationen. Daher dürfen Sie nur Basisoperationen verwenden, d.h. Sie müssen auch auf die Objekt-Methoden von Arrays (z.B. `Arrays.sort()`) verzichten. `length` ist allerdings ein Array-Attribut und darf verwendet werden.

- (a) Implementieren Sie eine Methode `int arrayGet(int[] array, int i)`, die für ein `int`-Array den Wert an der `i`-ten Position im Array zurück gibt. Achten Sie darauf, dass auch Positionen sinnvoll behandelt werden, die nicht im Array vorkommen.
- (b) Implementieren Sie eine Methode `int sum(int[] array)`, die alle Einträge des Arrays aufaddiert und diese Summe zurückgibt.
- (c) Implementieren Sie eine Methode `int mean(int[] array)`, die den Mittelwert aller Einträge eines Arrays zurückgibt.
- (d) Implementieren Sie eine Methode `void square(int[] array)`, die jeden Eintrag eines Arrays quadriert.
- (e) Implementieren Sie eine Methode `int max(int[] array)`, die den größten Eintrag eines Arrays zurückgibt.

**Aufgabe 8-2**     *Arrays 2*

Hier gibt es noch mehr Aufgaben zu Arrays. Auch hier dürfen Sie nur Basisoperationen verwenden, d.h. Sie müssen auch auf die Objekt-Methoden von Arrays verzichten.

- (a) Implementieren Sie eine Methode `void swap(int[] array, int i, int j)`. Diese soll die Einträge `i` und `j` des Arrays `array` miteinander vertauschen.
- (b) Implementieren Sie den Sortieralgorithmus <https://de.wikipedia.org/wiki/Selectionsort> mit dem folgenden Methodenrumpf `void sort(int[] array)`. Selectionsort sortiert die Einträge eines Arrays der Größe nach aufsteigend. Legen Sie dazu ein neues leeres Array der gleichen Größe an. Durchlaufen Sie das Array von links nach rechts und tauschen Sie das Element an der `i`-ten Position mit dem `i`-t kleinsten Element.

- (c) Implementieren Sie eine Methode `int median(int[] array)`, die den Median eines Arrays bestimmt. Bei Arrays mit gerader Anzahl an Elementen entspricht hier der Median dem größeren der beiden Kandidaten. Beachten Sie, dass der Median nicht das Gleiche ist wie der Mittelwert.
- (d) Implementieren Sie eine Methode `void int[] resize(int[] array, int length)`. Diese Methode soll die Länge eines Arrays wie folgt ändern:
- ist die neue Länge kürzer als die Bisherige, wird das Array an den letzten Positionen beschnitten.
  - ist die neue Länge länger als die Bisherige, werden die zusätzlichen Positionen am Ende des Arrays mit dem Wert 0 angefügt.

Korrektur: `resize` muss natürlich den Rückgabewert `int[]` haben. Überlegen Sie für sich selbst, warum!

- (e) Implementieren Sie eine Methode `String show(int[] array)`, die die Einträge des Array elementweise ausgibt.  
Für `int[] bsp = {6,4,9}`; liefert `show(bsp)` den String `"[6 4 9]"` `" [ 6 4 9 ] "`.

### Aufgabe 8-3 *Caesar-Chiffre*

Zum Verschlüsseln von Daten gibt es viele Methoden. Eine sehr alte Variante ist die Rotation des Alphabets um eine bestimmte Anzahl von Stellen. Die bekannteste davon ist vermutlich die Caesar-Chiffre, bei der eine Rotation von 13 Positionen ausgeführt wird. Dies ermöglicht bei 26 Buchstaben, die gleiche Methode zum Ver- und Entschlüsseln zu verwenden. Hier werden wir beliebige Rotationen berücksichtigen. Im Folgenden sehen Sie eine Rotation um 5 Stellen.

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e

Implementieren Sie ein Programm, das das Argument `args` des Benutzers verwendet und alle 26 möglichen Rotationen ausführt und ausgibt. Wenn der Nutzer noch eine Zahl vorgibt, wird nur diese Rotation angezeigt. Das Grundgerüst können Sie herunterladen. Sie benötigen für diese Aufgabe sehr wahrscheinlich die `String`-Methoden `charAt()` und `length()`. Es genügt, wenn Ihr Tool nur mit Kleinbuchstaben arbeiten kann. Belassen Sie alle anderen Zeichen bei ihrer Klardarstellung. Entziffern Sie mit Ihrem Tool folgende Textzeile:

robjvsmrox qveomugexcmr. ne rkcd nso kepqklo qovyocd.