



Prof. Dr. Bernhard Seeger
Dipl.-Inf. Marc Seidemann
Johannes Dröner, M.Sc.

Übungen zur Vorlesung
Praktische Informatik I

Abgabe: Montag, 25.11.2013,
bis **spätestens** 10:00 Uhr
über die ILIAS-Plattform

Blatt 5

Hinweis: Am 27.11. beginnt die Vorlesung pünktlich um 10:00 Uhr (s.t.) und endet um 11:30 Uhr.

Aufgabe 5.1:

(2+2+1=5P)

In Kapitel 4.4 der Vorlesung haben Sie Arrays (für beliebige Datentypen) kennengelernt. In dieser Aufgabe sollen Sie zunächst Methoden implementieren, in denen Arrays angelegt, mit Werten gefüllt oder ausgegeben werden.

- a) Implementieren Sie eine Methode `static char[] createCharArrayWithNames()`, die ein Array von `char` zurückliefert. Die Zeichen in diesem Array sollen die Namen aller Mitglieder ihrer Übungsgruppe bilden. Als Trennzeichen für die einzelnen Namen können Sie ein beliebiges Zeichen (z.B. `;`, `'`) benutzen.

Beispiel:

'M'	'a'	'r'	'c'	'_'	'U'	'w'	'e'	';	'K'	'ä'	'n'	'g'	'u'	'r'	'u'
-----	-----	-----	-----	-----	-----	-----	-----	----	-----	-----	-----	-----	-----	-----	-----

- b) Implementieren Sie eine Methode `static float[] createFloatArrayWithNumbers(int count)`, die ein Array erzeugt, in dem `count` viele `float`-Werte gespeichert werden können. In jedem Speicherplatz des Arrays sollen Sie einen Wert speichern, der nicht 0.0f ist oder der Position entspricht!

Beispiel für numbers = 7:

3.0f	5.1f	7.2f	9.3f	11.4f	13.5f
------	------	------	------	-------	-------

- c) Implementieren Sie zwei Methoden, um die Arrays aus Aufgabenteil a) und b) auf der Konsole auszugeben. Die Ausgabe soll der Syntax eines entsprechenden Array-Literal entsprechen!

Testen Sie die Methoden, in dem Sie die Methoden aus Aufgabenteil c) nutzen, um die, von den Methoden in a) und b) erzeugten, Arrays auszugeben

Aufgabe 5.2:

(3+3+3+3=12P)

In dieser Aufgabe sollen Sie verschiedene Operationen für Arrays implementieren. Damit dies nicht zu langweilig wird, dürfen Sie jeden Schleifentyp (**for**-, **while**-, **foreach**-Schleife) sowie **Rekursion** nur einmal benutzen!

Tipp: Für die Rekursion müssen Sie zusätzliche Methoden definieren!

- a) Implementieren Sie eine Methode `static float getMinimumElement(float[] numbers)`, die das Minimum aller Elemente in einem Array `numbers` bestimmt.

- b) Implementieren Sie eine Methode `static float getMaximumElement(float[] numbers)`, die das maximale Element in einem Array `numbers` bestimmt.
- c) Implementieren Sie eine Methode `static float calculateAverage(float[] numbers)`, die den Durchschnittswert aller Elemente für das Array `numbers` bestimmt.
- d) Implementieren Sie eine Methode `static boolean isSorted(float[] numbers, boolean ascending)`, die ermittelt, ob die Zahlen in einem Array sortiert sind.

Für `ascending = true` soll ermittelt werden, ob die Werte aufsteigend sortiert sind.
 Für `ascending = false` soll ermittelt werden, ob die Werte absteigend sortiert sind.

Testen sie alle implementierten Methoden durch Aufrufe in der main-Methode! Verwenden Sie als Parameter auch die Funktion `createFloatArrayWithNumbers(int numbers)` aus Aufgabe 5.1!

Aufgabe 5.3:

(2+4=6P)

- a) Implementieren Sie eine Funktion `static boolean arraysAreEqual(char[] firstArray, char[] secondArray)`, die genau dann `true` zurückgibt, wenn zwei Arrays aus Zeichen identisch sind.
- b) Implementieren Sie eine Funktion `static boolean containsPattern(char[] chars, char[] pattern)`, die genau dann `true` zurückgibt, wenn die im Array `pattern` gespeicherte Zeichenfolge im Array `chars` enthalten ist.

Beispiel:

'M'	'a'	'r'	'c'	'_'	'U'	'w'	'e'	';	'K'	'ä'	'n'	'g'	'u'	'r'	'u'
enthält									'K'	'ä'	'n'	'g'	'u'	'r'	'u'

Testen Sie die implementierten Methoden in der main-Methode! Verwenden Sie die Funktion `createCharArrayWithNames()` aus Aufgabe 5.1 b) als Parameter für `chars`. Überprüfen Sie mind. ein enthaltenes sowie ein nicht enthaltenes Wort.

Aufgabe 5.4:

(4P)

Implementieren Sie eine Funktion `static char[] reverse(char[] chars)`, die den Inhalt eines Arrays umdreht.

Testen Sie die implementierte Methode in der main-Methode! Verwenden Sie dazu `createCharArrayWithNames()` sowie min. ein weiteres Wort als Parameter für `chars`.

Aufgabe 5.5:

(5+3=8P)

- a) Implementieren Sie eine Methode `static int[][] pascalsTriangle(int rows)`, die ein 2D-Array erzeugt und die rows Reihen des pascalschen Dreiecks berechnet. Die Arrays sollen nicht größer sein als notwendig!

Beispiel für rows = 4:

1			
1	1		
1	2	1	
1	3	3	1

- b) In der Vorlesung wurde, auf Folie 232, eine Methode vorgestellt, die eine Pyramide/ein Dreieck von Sternen auf der Kommandozeile ausgibt. Schreiben Sie nun eine Methode `static void printTriangle(int[][] triangle)`, die für ein 2D-Array in Dreiecksform eine ähnliche Ausgabe auf der Kommandozeile erzeugt.

Testen sie die Ausgabe, indem Sie ein pascalsches Dreieck (aus Aufgabenteil a) ausgeben!