Programmierung 1
Prof. Dr. Jörg Kreiker
Wintersemester 2017/18

Hochschule Fulda
University of Applied Sciences

30. November 2018

Hausaufgabe 6

Abgabe: 10. Dezember 2018

Abgabe

Abzugeben ist *eine Datei* `Felder.java` mit insgesamt fünf Methoden. Vorgegeben ist diese Datei mit Signaturen, Testfällen und Hilfsmethoden zur Ausgabe.

Aufgabe 1 (2 Punkte)

Schreibt eine Methode `public static void boomBang(int von, int bis)`. Die Methode soll ein Feld von Strings returnieren, das wie folgt aufgebaut wird. Für Zahlen von einschließlich `von` bis ausschließlich `bis` werden alle durch 3 teilbaren Zahlen durch `"Boom"` und alle durch 5 teilbaren Zahlen durch `"Bang"` ersetzt. Zahlen, die durch 3 und durch 5 teilbar sind werden durch `BoomBang` ersetzt. Alle anderen Zahlen `i` werden mit `String.valueOf(i)` in einen String verwandelt. Ihr dürft annehmen, dass `bis > von`. Beispiele:

- `boomBang(1,6)` gibt das String Feld {`"1", "2", "Boom", "4", "Bang"`} zurück.
- `boomBang(12,16)` gibt das String Feld {`"Boom", "13", "14", "BoomBang"`} zurück.

0.1 Aufgabe 2 (3×2 Punkte

In der gesamten Aufgabe soll nur ein einziges Array manipuliert werden. Es sollen keine neuen Arrays angelegt werden. Es werden immer *Verweise* auf Arrays übergeben. Zusammen implementieren die drei Methoden das sogenannten *Bubblesort* zum Sortieren von Arrays.

- `public static void tausche(int[] arr, int i, int j)` vertauscht die Einträge bei den Indizes `i` und `j`. Man darf annehmen, dass beide Indizes innerhalb des Arrays liegen.
- `public static void alleTauschen(int[] arr)` Diese Methode betrachtet jeden Index i zwischen 0 und `arr.length - 2`. Wenn das dortige Element bei Index `i` größer ist als sein Nachfolger am Index $i + 1$, so vertausche die beiden Einträge.
- `public static void vertauscheN(int[] arr)` Diese Methode wendet die `alleTauschen` Methode `arr.length` mal auf `arr` an. Im Anschluss ist `arr` aufsteigend sortiert.

Aufgabe 3 (2 Punkte)

`public static int klumpen(int[] arr)` zählt die Anzahl von *Klumpen* in einem Feld. Ein Klumpen ist eine Menge von mindestens 2 direkt aufeinander folgenden identischen Elementen. Zum Beispiel enthält {`0,0,0,1,2,3,3,4`} zwei Klumpen, und {`0,0,0,0,0,0,0,0,0`} enthält einen Klumpen.