

Bitte lösen Sie die Übungsaufgaben bis zum 18/19 November. Wir besprechen die Lösungen in der nächsten Übung. Lösen Sie die Aufgaben in Pair Programming. Optionale Aufgaben könnten ein wenig länger dauern und müssen nicht gelöst werden. Das ist eine total subjektive Einschätzung, es könnte sein, dass Sie sehr schnell die Lösung finden. Daher meine Bitte: IMMER als Richtwert nehmen.

Aufgabe 5.1 (Array Ausgeben) – Schreiben Sie eine Methode, die die Inhalte eines eindimensionalen Integer-Arrays auf der Konsole ausgibt.

Aufgabe 5.2 (Echte Array Kopie) – Schreiben Sie eine Methode, die eine echte Kopie eines Arrays erstellt (ohne JAVA-Bibliotheken zu verwenden). Testen Sie Ihre Methode mit einem leeren Array und mit einem gefüllten Array.

```
// Erstellt eine echte Kopie des übergebenen Arrays a
public static int[] arrayCopy (int[] a)
```

Aufgabe 5.3 (Arrays, Rotation um 1) – Schreiben Sie eine Methode, die ein integer-Array um eine Position nach rechts bzw. nach links rotiert. (Alternativ können Sie auch zwei Methoden mit Links- und Rechtsrotation erstellen).

Beispiel Rechtsrotation {1, 2, 3, 4} -> {4, 1, 2, 3}

Beispiel Linksrotation {1, 2, 3, 4} -> {2, 3, 4, 1}

Aufgabe 5.4. (Arrays, Rotation um n) - Schreiben Sie eine Methode, die ein integer-Array um n Positionen nach rechts bzw. nach links rotiert. Auch hier können Sie alternativ zwei Methoden erstellen, jeweils für die Links- oder Rechtsrotation.

Aufgabe 5.5 (Nochmals Schleifen) – Schreiben Sie eine Methode, die solange eine Zufallszahl zwischen 1 und 6 generiert bis die gewünschte Würfel-Zahl (n) „gewürfelt“ wurde. Geben Sie auf der Konsole aus, wie viele Versuche hierzu benötigt wurden.

```
// n ist die zu würfelnde Zahl
public static void zahlWuerfeln (int n)
```

Aufgabe 5.6 (Schleifen - optional) – Schreiben Sie eine Methode, die solange zwei Zufallszahlen zwischen 1 und 6 generiert (simuliert das Würfeln) bis hintereinander dieselbe Zahl gewürfelt wurde. Geben Sie auf der Konsole aus, wie viele Versuche hierzu benötigt wurden.

Aufgabe 5.7 (Schleifen - optional) – Schreiben Sie eine Methode, die solange zwei Zufallszahlen zwischen 1 und 6 generiert bis hintereinander jeweils die 6 „gewürfelt“ (also generiert) wurde. Geben Sie auf der Konsole aus, wie viele Versuche hierzu benötigt wurden.

Aufgabe 5.8 (Arrays, Moral-O-Mat) – Gegeben seien drei Wortgruppen mit „Teilsätzen“. Schreiben Sie ein Programm, welches aus den drei Wortgruppen jeweils einen Teilsatz auswählt und zu einem ganzen Satz kombiniert. Gerne können Sie die Wortgruppen um eigene Teilsätze ergänzen ;-)

zufällig gewählter Teilsatz aus Wortgruppe 1 +

zufällig gewählter Teilsatz aus Wortgruppe 2 +

zufällig gewählter Teilsatz aus Wortgruppe 3 ergibt einen Satz.

Testen Sie Ihr Programm und haben Sie Spaß ☺.

Wortgruppe 1: Mut ist, Leichtsinn ist, Verantwortung ist, Demokratie ist, Geld ist, Liebe ist, Betrug ist, Freiheit ist, Denken ist, Selbstlosigkeit ist, Rekursion ist, Egoismus ist, Moral ist, Glück ist, Konsum ist, Das Leben ist, Das Schicksal ist, Unabhängigkeit ist, Gerechtigkeit ist, Der Tod ist, Programmiertechnik ist.

Wortgruppe 2: letztlich, sozusagen, am Ende des Tages, quasi, genaugenommen, wirklich, leider, zum Glück, durch die Blume gesagt, rückblickend, voraussichtlich, unter Umständen, erwiesenermaßen, gelinde gesagt, strenggenommen, unterm Strich, in gewisser Hinsicht, primär, mitnichten, vor allem.

Wortgruppe 3: die höchste Tugend, die Basis des Zusammenlebens, eine gesellschaftliche Pflicht, der Anfang allen Übels, (das, wofür es sich zu leben lohnt), falsch, zutiefst menschlich, kein Kavaliersdelikt mehr, völlig überschätzt, völlig unterschätzt, immer Gabe und Aufgabe zugleich, mit Vorsicht zu genießen, fragwürdig, existenzgefährdend, der Schlüssel zum Glück, eine Frage des Anstands, jedem selbst überlassen, eine Fügung, völlig willkürlich, immer schon vorherbestimmt, nicht mehr haltbar, unsere Rettung, eine Illusion, völlig überholt, nur die Spitze des Eisbergs.

Aufgabe 5.9 (Schleifen und Arrays - optional) – Versuchen Sie, ohne den Code auszuführen herauszufinden, was die Methode macht. Lassen Sie den Code laufen. Hat sich Ihre Annahme bestätigt? Machen Sie eine Skizze, die hilft zu verstehen, was passiert. Alternativ zur Skizze: Beschreiben Sie natürlichsprachlich, wie der Algorithmus funktioniert. Kommentieren Sie den Code derart, dass dieser einfach verständlich ist.

```
public static int[] machWas(int[] arr) {
    int n = arr.length;
    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i - 1; j++)
            if (arr[j] > arr[j + 1]) {
                int temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
    }
    return arr;
}
```