

# Java Arrays - Klausurvorbereitung

Dozent: Prof. Dr. Michael Eichberg  
Kontakt: [michael.eichberg@dhw.de](mailto:michael.eichberg@dhw.de), Raum  
149B  
Version: 1.0.1

# Aufgabe: Median bestimmen (Niveau: sehr einfach)

Sie haben ein *sortiertes* Array von Double-Werten. Schreiben Sie eine Methode, die den Median des Arrays bestimmt.

## ※ Hinweis

Sollte die Anzahl der Werte gerade sein, so ist der Median der Durchschnitt der beiden mittleren Werte, ansonsten ist es der mittlere Wert.

# Aufgabe: Zwei Arrays Komponentenweise aufsummieren (Niveau: einfache)

Gegeben sind zwei Arrays von ganzen Zahlen (Typ `long`). Schreiben Sie eine Methode (`sumArrays`), die die beiden Arrays komponentenweise aufsummiert und das Ergebnis in einem neuen Array zurückgibt. Das Ergebnis-Array hat die Länge des kürzeren der beiden Arrays.

# Aufgabe: Naive Textsuche (Niveau: mittel)

Gegeben sei ein Array von Zeichen (`text`) und ein Array von Zeichen (`zuSuchen`), die im Text(-Array) gesucht werden sollen. Zurückgegeben werden soll die Startposition des zu suchenden Textes im Text (Arrays) oder -1, wenn der Text nicht vorkommt.

Schreiben Sie eine entsprechende Methode `suche(...)`.

## Beispiel

```
char text[] = {'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p'};  
char zuSuchen[] = {'i', 'j', 'k', 'l'};  
  
IO.println(suche(text, zuSuchen)); // => 8
```

# Aufgabe: Füllen von Flächen (Niveau: mittel bis hoch)

## ◆ Bemerkung

Diese Aufgabe kann erst durchgeführt werden nachdem wir auch über Klassen und Objekte gesprochen haben.

Gegeben sei ein zwei-dimensionales Array von Farbwerten (Instanzen der Klasse *Color*).

z. B. `Color[][] colors = new Color[10][10];`

Die Klasse *Color* implementiert die Methode `boolean equals(Object o)`, die zwei Farben auf Gleichheit überprüft. Legen Sie eine entsprechende Klasse an oder nutzen Sie eine vorhandene Implementierung (z. B. `java.awt.Color`).

Schreiben Sie eine Methode, die – gegeben eine bestimmte Position  $(x,y)$  und eine Farbe  $c$  – das Feld selber und auch alle benachbarten Felder, die die gleiche Farbe haben wie das ursprüngliche Feld, mit der Farbe  $c$  füllt. Ein Feld ist benachbart, wenn es sich in der gleichen Zeile oder Spalte befindet und eine Kante teilt.

Stellen Sie sich die folgenden Fragen:

- Funktioniert Ihre Implementierung auch dann, wenn die Arrays in der zweiten Dimension unterschiedlich lang sind?
- Wie verhält sich Ihre Implementierung, wenn die Farbe an der Position  $(x,y)$  bereits die Farbe  $c$  hat?
- Dokumentieren Sie Ihre Implementierung. Achten Sie bei der Dokumentation darauf die Sonderfälle zu berücksichtigen.

## ※ Hinweis

Es bietet sich ggf. an eine rekursive Methode (fill) zu implementieren.