

## Praktikumstutorium – Blatt 2

Als Vorbereitung auf das Testat 2 sollten Sie unbedingt diese Aufgaben bearbeiten.

Sofern dabei Schwierigkeiten auftreten, sollten Sie unbedingt das **Tutorium des Programmierpraktikums** besuchen

Tutoriumszeiten im Raum OH 12/4.030

Montag	Dienstag	Mittwoch	Donnerstag	Freitag
12.00 – 14.00 Uhr	10.00 – 16.00 Uhr	10.00 – 14.00 Uhr	10.00 – 14.00 Uhr	Testat

### Methoden mit Feldern als Parameter

- ~~Werte zählen~~  
Schreiben Sie eine Methode `countNegatives`, die als Parameter ein eindimensionales Feld des Typs `int` besitzt. Die Methode `countNegatives` soll die negativen Werte in diesem Feld zählen und den ermittelten Wert **zurückgeben**.
- ~~Werte aufsummieren~~  
Schreiben Sie eine Methode `sumUpNegatives`, die als Parameter ein eindimensionales Feld des Typs `int` besitzt. Die Methode `sumUpNegatives` soll die Summe der negativen Werte in diesem Feld bilden und den ermittelten Wert **zurückgeben**.
- ~~Bestimmen des Maximums in einem Feld~~  
Schreiben Sie eine Methode `maximum`, die ein Feld von `int`-Werten als Parameter besitzt und die den größten Wert in diesem Feld bestimmt und **zurückgibt**.  
*Hinweis:* Versuchen Sie, mit einem Durchlauf durch das Feld auszukommen.
- ~~Bestimmen der Häufigkeit des Maximums in einem Feld~~  
Schreiben Sie eine Methode `countMaximum`, die ein Feld von `int`-Werten als Parameter besitzt und die zählt, wie häufig der größte Wert vorkommt. Die ermittelte Anzahl soll **zurückgegeben** werden.
- ~~Sortierung prüfen~~  
Schreiben Sie eine Methode `boolean isSorted`, die als Parameter ein eindimensionales Feld des Typs `int` besitzt. Die Methode `isSorted` soll `true` **zurückgeben**, falls die im Feld enthaltenen Werte aufsteigend sortiert sind; sonst soll `false` **zurückgegeben** werden.
- ~~Palindrome~~  
Ein Palindrom ist eine Folge von `int`-Werten, die vorwärts und rückwärts identisch gelesen werden kann.  
Beispiele sind die Folgen `12 34 78 34 12` oder `5 17 88 88 17 5`.  
Schreiben Sie eine Methode `checkArray`, die für ein über einen Parameter übergebenes Feld von `int` bestimmt, ob die Folge der Zahlen ein Palindrom bildet. Die Methode soll einen Wert des Typs `boolean` **zurückgeben**.
- ~~Erhöhen der Inhalte eines Feldes~~  
Schreiben Sie eine Methode `increaseArray`, die als Parameter ein eindimensionales Feld des Typs `int` und einen `int`-Wert besitzt. Die Methode `increaseArray` soll alle Werte des Feldes um den im zweiten Parameter übergebenen Wert erhöhen und das veränderte Feld **zurückgeben**.  
*Beispiel:* Werden ein Feld mit den Elementen `80, 7, 1, 56, 11, 72, 43, 37` als erstes und der Wert `17` als zweites Argument übergeben, so soll dieses Feld manipuliert und als `97, 24, 18, 73, 28, 89, 60, 54` **zurückgegeben** werden.
- ~~Bedingtes Verdoppeln der Inhalte eines Feldes~~  
Schreiben Sie eine Methode `doubleIfContainsPositive`, die als Parameter ein eindimensionales Feld des Typs `int` besitzt. Die Methode `doubleIfContainsPositive` soll alle Werte des Feldes verdoppeln, falls in dem Feld mindestens ein positiver Wert vorkommt; sonst soll das Feld unverändert bleiben. Das (veränderte) Feld soll **zurückgegeben** werden.  
*Beispiel:* Wird ein Feld mit den Elementen `0, -7, 1, 5, -1, 2, 4` als Argument übergeben, so soll dieses Feld manipuliert und als `0, -14, 2, 10, -2, 4, 8` **zurückgegeben** werden.
- Erzeugen eines Textes  
Schreiben Sie eine Methode `toString`, die als Parameter ein eindimensionales Feld des Typs `int` besitzt. Die Methode `toString` soll einen Text erzeugen, der alle Inhalte des Feldes in der Reihenfolge ihres Auftretens durch Kommas getrennt enthält. Der erzeugte Text soll **zurückgegeben** werden.

- ~~Erzeugen eines Feldes mit drei Elementen~~

Schreiben Sie eine Methode `copyStartingValues`, die als Parameter ein eindimensionales Feld des Typs `int` besitzt. Die Methode `copyStartingValues` soll ein neues Feld erzeugen, das in seinen Elementen genau die ersten drei Werte des als Argument übergebenen Feldes enthält, falls dieses mehr als zwei Elemente besitzt. Das erzeugte Feld soll **zurückgegeben** werden. Besitzt das als Argument übergebene Feld weniger als drei Elemente, soll vollständig kopiert werden.

*Beispiel:* Wird ein Feld mit den Elementen `80, 7, 1, 56, 11, 72, 43, 37` als Argument übergeben, so soll ein neues Feld mit den Werten `80, 7, 1` **zurückgegeben** werden.

- ~~Erzeugen eines Feldes mit ausgesuchten Inhalten~~

Schreiben Sie eine Methode `selectNegatives`, die als Parameter ein eindimensionales Feld des Typs `int` besitzt. Die Methode `selectNegatives` soll ein Feld zurückgeben, in dem ausschließlich die negativen Werte des als Argument übergebenen Feldes enthalten sind. Die Methode `countNegatives` kann dazu benutzt werden, die Größe des zurückgegebenen Feldes zu bestimmen.

*Beispiel:* Wird ein Feld mit den Elementen `80, -7, 1, 56, -11, -72, 0, 37` als Argument übergeben, so soll ein neues Feld mit den Werten `-7, -11, -72` **zurückgegeben** werden.

- ~~Erzeugen eines Feldes mit invertiertem Inhalt~~

Schreiben Sie eine Methode `copyAndInvert`, die als Parameter ein eindimensionales Feld des Typs `int` besitzt. Die Methode `copyAndInvert` soll ein Feld zurückgeben, das alle Werte des als Argument übergebenen Feldes in umgekehrter Reihenfolge enthält.

*Beispiel:* Wird ein Feld mit den Elementen `80, -7, 1, 56, -11, -72, 0, 37` als Argument übergeben, so soll ein neues Feld mit den Werten `37, 0, -72, -11, 56, 1, -7, 80` **zurückgegeben** werden.

- ~~Zusammenführen von Feldern~~

Schreiben Sie eine Methode `addArrays`, die zwei `int`-Felder als Parameter besitzt. Haben diese Felder die gleiche Länge, so sollen die Werte mit dem gleichen Index addiert und die Summe unter diesem Index in einem dritten `int`-Feld abgelegt werden. Dieses Feld soll als Ergebnis des Aufrufs von `addArrays` **zurückgegeben** werden. Unterscheiden sich die als Argumente übergebene Felder in ihrer Länge, soll ein leeres Feld **zurückgegeben** werden.

- ~~Zählen von Folgen~~

Schreiben Sie eine Methode `countSequences`, die ein Feld von `int`-Werten als Parameter besitzt. Die Methode `countSequences` soll die Anzahl der im Feld enthaltenen Zahlenfolgen ermitteln, in denen **nicht** der Wert `0` vorkommt. Eine solche Zahlenfolge endet immer mit dem Auftreten einer `0` oder dem Ende des Feldes. Die ermittelte Anzahl soll von der Methode **zurückgegeben** werden.

*Beispiel:* Wird ein Feld mit den Elementen `80, 7, 1, 0, 11, 72, 0, 0, 37, 61` als Argument übergeben, so soll der Aufruf von `countSequences` als Ergebnis `3` liefern, da die Folgen `80, 7, 1` und `11, 72` und `37, 61` auftreten.

## Implementierungen zum Übungsblatt 1 (Aufgabe 2)

Gehen Sie bei den folgenden Aufgaben davon aus, dass in den als Argumente übergebenen Feldern keine Duplikate von Werten vorkommen.

- ~~Prüfen auf einen gemeinsamen Wert~~

Schreiben Sie eine Methode `oneInBoth`, die zwei `int`-Felder als Parameter besitzt. Gibt es mindestens einen Wert, der in beiden Feldern vorkommt, so soll **true** **zurückgegeben** werden; sonst soll **false** **zurückgegeben** werden.

- ~~Prüfen auf unterschiedliche Werte~~

Schreiben Sie eine Methode `noneInBoth`, die zwei `int`-Felder als Parameter besitzt. Gibt es keinen Wert, der in beiden Feldern vorkommt, so soll **true** **zurückgegeben** werden; sonst soll **false** **zurückgegeben** werden.

- ~~Zählen der gemeinsamen Werte~~

Schreiben Sie eine Methode `countValuesInBoth`, die zwei `int`-Felder als Parameter besitzt. Die Methode soll die Anzahl der Werte, die in beiden Feldern vorkommen, **zurückgeben**.

- ~~Vergleichen der Werte 1~~

Schreiben Sie eine Methode `oneIsGreater`, die zwei `int`-Felder als Parameter besitzt. Die Methode soll **true** **zurückgeben**, falls es in dem als ersten Argument übergebenen Feld mindestens einen Wert gibt, der größer ist als alle Werte des zweiten Feldes; sonst soll **false** **zurückgegeben** werden.

- ~~Vergleichen der Werte 2~~

Schreiben Sie eine Methode `allAreGreater`, die zwei `int`-Felder als Parameter besitzt. Die Methode soll **true** **zurückgeben**, falls alle in dem als ersten Argument übergebenen Feld auftretenden Werte größer sind als alle Werte des zweiten Feldes; sonst soll **false** **zurückgegeben** werden.