

Programmieren und Software-Engineering I

Übung 19

Name: _____ Klasse: _____ Datum: _____

Lernziele:

- Arrays

Aufgabe 1: Sortieren (Ex_19_01_Sort)

Erstelle ein Java-Programm welches ein unsortiertes Array (z.B. Arraygröße = 20, mit Zufallszahlen zwischen 1 bis 100) sortiert und ausgibt. Verwende dazu mindestens 2 verschiedene Algorithmen (z.B. Selectionsort, Bubblesort, Insertionsort,...)

Beispiel:

Unsortiert:	44, 32, 16, 86, ...
Selectionsort:	5, 8, 16, 23, ...
Unsortiert:	33, 86, 5, ...
Bubblesort:	3, 5, 9, ...
Unsortiert:	37, 22, 16, ...
Insertionsort:	7, 8, 13, ...

Aufgabe 2: Mengen Operationen (Ex_19_02_Set)

Schreibe ein Java-Programm, welches die mathematischen Mengenoperationen Durchschnitt, Vereinigung und Differenz von 2 Mengen (Feldern) zur Verfügung stellt. Das Ergebnis wird dabei in einem neuen Array gespeichert!

Hinweise: In einer Menge kommt jedes Element nur einmal vor.
Die Mengen können natürlich unterschiedlich viele Elemente enthalten.

Beispiel:

```
int[] set1 = { 1, 10, 3, 128, 7, 9};
int[] set2 = { 10, 9, 87, 55, 1, 76, 4, 5};
```

Durchschnitt:	{ 1, 10, 9 }
Vereinigung:	{ 1, 10, 3, 128, 7, 9, 87, 55, 76, 4, 5 }
Menge1 ohne Menge2:	{ 3, 128, 7 }
Menge2 ohne Menge1:	{ 87, 55, 76, 4, 5 }

Aufgabe 3: mehrfache Zahlen im Array löschen (Ex_19_03_DeleteShift)

Folgende Aufgaben sollen in einem Array durchgeführt werden:

- Ausgabe des ursprünglichen Arrays
- Ersetzen aller mehrfach vorkommenden Zahlen durch 0
- Nachschieben der übrig gebliebenen Zahlen nach links

Es darf dabei **kein zweites Array verwendet** werden!!!

Beispiel:

```
int[] ar = { 14, 56, 23, 18, 14, 36, 14, 77, 66, 56, 23, 11, 11};
```

// Normale Ausgabe:
// 14, 56, 23, 18, 14, 36, 14, 77, 66, 56, 23, 11, 11

// Ersetzen durch 0
// 0, 0, 0, 18, 0, 36, 0, 77, 66, 0, 0, 0, 0

// Nachschieben
// 18, 36, 77, 66, 0, 0, 0, 0, 0, 0, 0, 0

Programmieren und Software-Engineering I

Übung 19

Aufgabe 4: Pattermatching (Ex_19_04_Patternmatching)

Patternmatching gehört zu den Standardproblemen in der Informatik. Dabei soll festgestellt werden, ob ein Muster in einem anderen auftaucht oder nicht.

Gegeben sind zwei unterschiedlich lange Arrays. Es soll nun festgestellt werden, ob das kürzere im längeren vorkommt und wenn ja auf welcher Position es beginnt.

Beispiel: `int[] long = {5, 3, 8, 4, 6, 2, 2, 7, 8, 3, 8, 5, 5, 9, 4, 3, 5};`
 `int[] short = {3, 8, 5};`

Ausgabe: Kürzeres Array kommt ab Position 9 vor.

WICHTIG

Alle Programme müssen einen Programmkopf (=Beschreibung) enthalten.

z.B.:

```
/******  
*      Name:      Max Mustermann  
*      Hü:        4  
*      Bsp:       2  
*      Datum:     20.10.2015  
*      Dateiname:  HUE_04_02_Dreieck.java  
*      Beschreibung: Es wird für gegebene Seitenlängen a, b und c geprüft, ob es ein gleichseitiges, ein  
*                   gleichschenkeliges, ein rechtwinkeliges, ein sonstiges gültiges oder ein ungültiges  
*                   Dreieck ist.  
*****/  
public class HUE_04_02_Dreieck {
```