

Programmieren und Software-Engineering I **Übung 24**

Name: _____ **Klasse:** _____ **Datum:** _____

Lernziele:

- Testverbesserung

Testverbesserung (Ex_24_01_Correction)

Aufgabe 1a: Fülle ein beliebig großes Array (z.B. size = 10) mit **Zufallszahlen** zwischen 1 und 100 (ug bzw. og verwenden). Dabei wird immer abwechselnd eine gerade und eine ungerade Zahl festgelegt. Gib die Zahlen anschließend aus. Bestimme anschließend die **größte** und **zweitgrößte** Zahl und gib diese aus. Prüfe im letzten Schritt, ob es sich um lauter verschiedene Zahlen handelt.

Ausgabe: Array: 64 85 18

Größte Zahl: 93

Zweitgrößte Zahl: 87

Verschiedene Zahlen: Nein

Aufgabe 1b: Erstelle ein vollständiges Java-Programm am PC, welches zuerst ein beliebig großes **char-Array** **zufällig** mit **Großbuchstaben** füllt („A“ steht an Position 65, „B“ an Position 66,...).

Nach der zufälligen Füllung wird das **Array ausgegeben**.

Im nächsten Schritt wird das **Array sortiert** und erneut **ausgegeben**.

Beispiel: Ausgabe nach dem Füllen: B M X V ... A

Ausgabe nach dem Sortieren: A B B F

Aufgabe 1c: Erstelle ein vollständiges Java-Programm am PC, welches **ein zufällig Passwort** der Länge n erzeugt. (n **MUSS** zwischen **5 und 30** liegen und wird vom Benutzer eingegeben. Bei **ungültiger Eingabe** erfolgt eine Fehlermeldung und der Benutzer muss erneut eingeben). Das Passwort beginnt mit einem **Großbuchstaben**, dann folgt ein **Kleinbuchstabe** und dann eine **Ziffer** (zwischen 0 und 9). In weiterer Folge wieder ein Großbuchstabe, ein Kleinbuchstabe...

Beispiel: für $n = 10$

z.B. Gp4Qw2Br8A

//

Aufgabe 1d: Erstelle ein vollständiges Java-Programm am PC für folgende Aufgabenstellung:

- Mehrmaliges Würfeln mit 2 Würfeln und Ausgabe (sollte auch für 3 oder mehr Würfel leicht anpassbar sein)
- Es wird solange gewürfelt, bis dass jeder mögliche Pasch (1 und 1, 2 und 2, ..., 6 und 6) mindestens einmal vorgekommen ist

Beispiel:

Wurf 1:	5 und 3
Wurf 2:	3 und 3 (Pasch)
Wurf 3:	1 und 4
Wurf 4:	6 und 1
...	
Wurf 9:	6 und 6 (Pasch)
...	
Wurf 38:	1 und 1 (Pasch)

Programmieren und Software-Engineering I

Übung 24

Testverbesserung (Ex_24_02_Correction)

1. Arraybeispiel 1

Erstelle ein Java-Programm das jeden der folgenden Schritte **einzeln(2)** durchführt:

- Anlegen eines 20stelligen Arrays für int-Zahlen (3) (im HP)
- Füllen des Arrays mit Zufallszahlen zwischen -20 und 20 (5)
 - Methode mit Array als Parameter
 - Keine Rückgabe
- Ausgabe des Arrays in Zeilen zu je 4 Werten (3)
 - Methode mit dem Array und der Anzahl der Werte pro Zeile als Parameter
 - Keine Rückgabe
- Bestimmen des Maximums im Array (4)
 - Methode mit dem Array als Parameter
 - Maximum als Rückgabewert
 - Ausgabe im Hauptprogramm
- Shift-Left des Arrays (8): Die Zahlen im Array sollen solange nach links verschoben werden, bis eine positive Zahl oder eine Null an erster Stelle steht.
 - Methode mit Array als Parameter
 - keine Rückgabe
- Ausgabe des Arrays in Zeilen zu 4 Werten
- Geprüfte Eingabe des Benutzers einer Zahl zwischen 0 und 20. Gibt er eine falsche Zahl ein, so wird er so lange aufgefordert eine richtige einzugeben, bis er dieses tut. (3)
 - im Hauptprogramm
- Ermitteln und Ausgeben aller Zahlen im Array, deren absoluter Wert (Zahl ohne Vorzeichen, daher immer positiv) größer ist als die eingegebene Zahl. (6)
 - Methode mit Array und Zahl als Parameter
 - Ausgabe in der Methode
 - Keine Rückgabe
- Aufsteigend sortieren des Arrays (6)
 - Methode mit Array als Parameter
 - Algorithmus beliebig
 - Keine Rückgabe
- Ausgabe des Arrays in Zeilen zu je 5 Werten
- Kumulieren des Arrays (7)
 - Methode mit dem Array als Parameter
 - an jeder Stelle im Lösungsarray soll die Summe der Zahlen im Originalarray bis zu dieser Stelle stehen.
 - Lösungsarray als Rückgabe
- Ausgabe des Lösungsarrays in Zeilen zu 5 Werten

Lösungsbeispiel für Kumulieren:

Ausgangsarray

1	1	4	6	7
---	---	---	---	---

Kumuliertes Array

1	2 (1+1)	6 (1+1+4)	12 (1+1+4+6)	19 (1+1+4+6+7)
---	---------	-----------	--------------	----------------

2. Arrayprüfung

Erstelle eine Methode, die für ein Array feststellt, ob der Abstand zwischen den Werten immer gleich groß ist(10). Das Ergebnis wird als Rückgabewert verwendet (Vorschlag: boolscher Wert)(4). Im Hauptprogramm erfolgt eine Ausgabe des Arrays und eine entsprechende Meldung.(3)

Programmieren und Software-Engineering I

Übung 24

Prüfe das Programm für folgende beiden Arrays:

arr1 = {1,3,5,7,8,10,12,14,16}

arr2 = {3,10,17,24,31,38,45,52}

Ausgabebeispiel:

Array

1 3 5 7 8 10 12 14 16

Der Abstand ist nicht immer gleich.

Array

3 10 17 24 31 38 45 52

Der Abstand ist immer gleich

3. Namenssuche

Fr. Prof. Schlaumeyer hat schon wieder ein Problem. Die Namen ihrer Schüler und Schülerinnen sind so ungewöhnlich, dass sie nicht weiß, ob es sich dabei um Buben oder Mädchen handelt. Sie braucht aber nun dringend eine Liste aller Mädchen ihrer Klasse. Um ihr zu helfen, hat ihr die Sekretärin eine Liste gegeben, die für jedes Klassenkind aussagt, ob es männlich (false) oder weiblich (true) ist. Erstelle ein Java-Programm, dass diese Liste der weiblichen Schüler ausgibt.

Lösungshinweis:

Verwende dazu folgende Arrays:

```
String[] names = {"Eren", "Tabea", "Alex", "Maha", "Kaya", "Mate", "Reka", "Junis", "Kazuya",  
"Riko", "Rico", "Flynn"};
```

```
boolean[] isFemal = {false, true, true, true, false, false, true, false, false, true, false, false};
```

4. Sieb des Eratosthenes

Primzahlen haben schon die alten Wissenschaftler interessiert. Da sie aber relativ aufwändig zu ermitteln sind, hat der griechische Mathematiker Eratosthenes von Kyrene schon ca. 200 Jahre vor Christus eine Möglichkeit entwickelt alle Primzahlen bis zu einer vorgegebenen Obergrenze einfach zu ermitteln.

Dazu werden alle Zahlen ab 2 bis zu dieser Grenze aufgeschrieben. Im nächsten Schritt wird die erste Zahl größer als 1 gesucht (in diesem Fall 2) und alle Vielfachen dieser Zahl durch 0 ersetzt. Der Schritt wird solange wiederholt, bis die nächste Zahl größer ist als die Hälfte der vorgegebenen Grenze. Alle übrigbleibenden Zahlen (von 0 verschieden) sind Primzahlen.

Erstelle ein Java-Programm, das diesen Algorithmus mit Hilfe eines Arrays nachbildet. Dabei soll der Benutzer in einem ersten Schritt eine Obergrenze zwischen 10 und 1000 eingeben können (Achtung: Prüfung notwendig) (4). Nach der Ermittlung der Primzahlen(18) sollen diese in Reihen zu 10 Zahlen ausgegeben werden.

Ablaufbeispiel:

Geben Sie bitte eine Zahl ein: 250

Primzahlen bis 250:

2 3 5 7 11 13 17 19 23 29
31 37 41 43 47 53 59 61 67 71

.....

233 239 241

Programmieren und Software-Engineering I

Übung 24

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 gleich...
*****/
```