

Programmieren und Software-Engineering I

Übung 18

Name: _____ Klasse: _____ Datum: _____

Lernziele:

- Arrays

Aufgabe 1: Gewinnspiel Zahlen (Ex_18_01_ProfitPerson)

Erstelle ein Java-Programm, welches ein Gewinnspielspiel simuliert. Dabei werden 10 Gewinnen zwischen 10 und 100 zufällig bestimmt. Ein zweites Array ist mit 10 Namen befüllt. Der 0.Name hat den 0.Gewinn erzielt, der erste Name den ersten Gewinn...

Code: `int[] gewinn = new int[10];`
 `String[] namen = {"Peter", "Max", "Lisa", "Peter", "Konrad", "Sara", "Sara", "Peter", "Stefan", "Max"};`

Folgende Aufgaben sind durchzuführen:

- Entsprechende Ausgabe
- Ermittlung der Person mit dem höchsten Gewinn
- Berechnung des Durchschnittsgewinns
- Wer gewann über dem Durchschnitt
- Ermittlung der größten Differenz der Gewinne
- Ermittle jene Personen, die gleich heißen und gib die jeweiligen Gewinne aus

Aufgabe 2: Häufigkeit(Ex_18_02_Frequency)

Gegeben sei ein mit ganzen Zahlen gefülltes Array. Danach werden die **Häufigkeiten** der einzelnen Zahlen sowie die am häufigsten vorkommende Zahl ausgegeben.

Beispiel: `int [] feld = {7,23,56,1,23,15,23,...};`

Häufigkeiten: 7 : 2x
 23 : 6x
 56: 1x
 ...
Häufigste Zahl: 23 kommt 6x vor

Aufgabe 3: Sieb des Eratosthenes (Ex_18_03_PrimesEratosthenes)

Das Sieb des Eratosthenes ist ein Verfahren zur Berechnung aller Primzahlen im Bereich von 2 bis n. Zuerst wird eine Liste aller Zahlen von 2 bis n aufgestellt. Dann werden alle Vielfachen von 2 in dieser Liste gestrichen. Danach wird die nächste noch in dieser Liste enthaltene Zahl genommen und alle Vielfachen dieser Zahl werden wiederum gestrichen...

Zuletzt werden alle in der Liste verbliebenen Zahlen (Primzahlen von 2 bis n) ausgegeben (immer 15 Zahlen pro Zeile).

Ausgabe: 2 3 5 7 11 13 17 19 23

Ein Beispiel des Algorithmus:

	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50

	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50

Vielfache von 2

Vielfache von 3

(a) Das Sieb mit den aufgeschriebenen Zahlen

(b) Gestrichene Zahlen im Sieb

Programmieren und Software-Engineering I

Übung 18

Aufgabe 4: Mergesort (Ex_18_04_Mergesort)

Gegeben seien 2 mit ganzen Zahlen sortierte Arrays. Diese beiden Arrays sollen nun einem dritten Array zugewiesen werden und dabei wieder in sortierter Reihenfolge aufscheinen.

Beispiel:

```
int []feld1={1,5,7,13,25,36,67,78,92,103,128,169,231}
int []feld2={3,5,10,60,97,118,400,410}

erg = {1,3,5,5,7,10,13,25,36,60,67,78,92,97,103,118,128,169,231,400,410}
```

Aufgabe 5: Zeitschätzung (Ex_18_05_TimeEstimate)

Bei dieser Aufgabe geht es um die Entwicklung eines Programms, das es einem Benutzer erlaubt, sein Zeitabschätzungsgefühl zu trainieren. Schreibe ein Java-Programm, das folgendes tut:

1. Es wird eine Zufallszahl zwischen 10 und 20 generiert. Sie spiegelt die Anzahl an Sekunden wieder, die der Benutzer abschätzen soll (erwartete Zeit).
2. Die Zahl wird ausgegeben.
3. Der Benutzer wird zweimal zu einer beliebigen Eingabe aufgefordert. Unmittelbar nach jeder Eingabe wird die Zeit gemessen. Nutze dazu die Funktion `System.currentTimeMillis()`, die als long-Wert die Anzahl an Millisekunden liefert, die zwischen dem 1.1.1970 und dem Zeitpunkt ihres Aufrufs verstrichen ist.
4. Berechne die zwischen den beiden Eingaben verstrichene Zeit.
5. Berechne und gib die erwartete Zeit, die tatsächlich verstrichene Zeit, die erlaubte Abweichung (hier $A = 1$ Sekunde) und die tatsächliche Abweichung aus.
6. Ist die tatsächliche Abweichung größer als die erwartete Abweichung, beginnt alles von neuem. Im anderen Fall wird das Programm beendet.

Orientiere dich dabei, was den Programmablauf als auch was die Ein- und Ausgaben angeht, an folgendem Beispiel (Eingaben stehen in <>):

Beispiel:

```
Bitte warten Sie möglichst genau 14 Sekunden zwischen Ihren nächsten beiden
Eingaben!
Beliebige Eingabe, um die Zeitnahme zu starten: <>
Beliebige Eingabe, um die Zeitnahme zu stoppen: <>
Erwartete Zeit = 14 Sekunden
Ihre Zeit = 12 Sekunden plus 770 Millisekunden
Erlaubte Abweichung = 1 Sekunden
Ihre Abweichung = 1 Sekunden plus 230 Millisekunden
Das war nicht genau genug: Bitte wiederholen!
```

```
Bitte warten Sie möglichst genau 15 Sekunden zwischen Ihren nächsten beiden
Eingaben!
Beliebige Eingabe, um die Zeitnahme zu starten: <>
Beliebige Eingabe, um die Zeitnahme zu stoppen: <>
Erwartete Zeit = 15 Sekunden
Ihre Zeit = 15 Sekunden plus 205 Millisekunden
Erlaubte Abweichung = 1 Sekunden
Ihre Abweichung = 0 Sekunden plus 205 Millisekunden
Ziel erreicht: Glückwunsch!
```

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 {
```