

Labor 1 - Deadline (3. Woche - 20 Oktober)

- jede Lösung eines Problems wird durch Klassen getrennt (-1 pkt)
- Variablen und Methoden müssen konsistente und repräsentative Namen haben (-1 pkt)
- die Aufgabe wird abgegeben, indem der Link zum Git-Repository geteilt wird (den Fortschritt bei der Lösung der Aufgabe wird im Betracht genommen und es wird erwartet, dass mehrere Commits vorhanden sind und nicht nur ein einziger, der die gesamte Lösung enthält. Also viele Commits erstellen!) (-10 pkt)
- für jede Anforderung jeder Aufgaben muss man mindestens 2 Tests schreiben (1 Test für den erwarteten Fall, 1 Test für den unerwarteten/falschen Fall) (-10 pkt)
- Die Lösung darf keine Kompilierungsfehler enthalten. Laufzeit Warnungen werden akzeptiert sein (-10 pkt)
- für jede Woche, in der die Aufgabe verspätet abgegeben wird, wird ein Punkt abgezogen
- Alle in den Aufgabenanforderungen genannten Zahlen sind ganze Zahlen

Problem 1:

Die Java Universität hat die folgenden Benotung Regel:

- Jeder Student bekommt eine **Note** zwischen **0** und **100**.
- Eine **Note** weniger als **40** ist eine nicht ausreichende Note.

Der Professor rundet die Note mit die folgenden Regel ab:

- Ob die Differenz zwischen die **Note** und die nächste **multipl von 5** weniger als 3 ist, dann wird die Note zu die nächste **multipl von 5** abgerundet.
- Ob die **Note** weniger als 38 ist, wird die **Note nicht** abgerundet.

z.B. 84 => 85

29 => 29

1. Schreiben Sie eine Methode, die ein Array von **Noten** bekommen soll. Als die Rückgabewert soll die Methode ein Array mit **nicht ausreichende Note** liefern.
2. Schreiben Sie eine Methode, die ein Array von **Noten** bekommen soll. Als die Rückgabewert soll die Methode den **Durchschnittswert** liefern.
3. Schreiben Sie eine Methode, die ein Array von **Noten** bekommen soll. Als die Rückgabewert soll die Methode **eine Array mit die abgerundete Note** liefern.
4. Schreiben Sie eine Methode, die ein Array von **Noten** bekommen soll. Als die Rückgabewert soll die Methode **die maximale abgerundete Note** liefern.

Problem 2:

Es gibt eine Array mit **n** positive Zahlen.

1. Finden Sie die maximale Zahl.
 2. Finden Sie die minimale Zahl.
 3. Finden Sie die maximale Summe von **n-1** Zahlen.
z.B. [4, 8, 3, 10, 17] => 4 + 8 + 10 + 17 = **39**
 4. Finden Sie die minimale Summe von **n-1** Zahlen.
z.B. [4, 8, 3, 10, 17] => 4 + 8 + 3 + 10 = **25**
-

Problem 3:

Es gibt 2 große Zahlen. Die Zahlen sind als Array dargestellt.

1. Berechnen Sie die Summe. Die Zahlen haben die gleiche Anzahl an Ziffern.
z.B. [1 3 0 0 0 0 0 0] + [8 7 0 0 0 0 0 0] = [1 0 0 0 0 0 0 0]
 2. Berechnen Sie die Differenz. Die Zahlen haben die gleiche Anzahl an Ziffern.
z.B. [8 3 0 0 0 0 0 0] - [5 4 0 0 0 0 0 0] = [2 9 0 0 0 0 0 0]
 3. Berechnen Sie die Multiplikation. Erste Zahl ist ein große Zahl, der Zweite Zahl ist nur ein Ziffer.
z.B. [2 3 6 0 0 0 0 0] * 2 = [4 7 2 0 0 0 0 0]
 4. Berechnen Sie die ganzzahlige Division. Erste Zahl ist ein große Zahl, der Zweite Zahl ist nur ein Ziffer.
z.B. [2 3 6 0 0 0 0 0] / 2 = [1 1 8 0 0 0 0 0]
-

Problem 4:

Markus will ein USB Laufwerk und eine Tastatur kaufen. Der Elektronik-Shop hat verschiedene USB Laufwerke und Tastaturen mit verschiedene Preise, als Array von Zahlen dargestellt.

1. Schreiben Sie eine Methode, die die **billigste** Tastatur zurückgibt.
z.B. [40 35 70 15 45] => 15
2. Schreiben Sie eine Methode, die die **teuerste** Gegenstand zurückgibt.
z.B. **Tastatur** = [15 20 10 35], **USB** = [20, 15, 40 15] => 40
3. Schreiben Sie eine Methode, die die **teuerste** USB Laufwerk, die Markus kaufen kann, zurückgibt.
z.B. **Preise** = [15 45 20], **Budget** = 30 => 20
4. Finden Sie, anhand des Markus **Budget** und der **Preislisten** für die Tastaturen und USB-Laufwerke, den Geldbetrag Markus ausgehen wird. Wenn er nicht genug für beide hat, geben Sie stattdessen **-1** zurück. Er kauft nur die zwei benötigten Gegenstände.
z.B. b=60, tastaturen = [40 50 60] und usb Laufwerke = [8 12] => 50 + 8 = 58