

# InfTech Hausaufgaben 3

## Aufgabe 2

- Beschreibung von void `wert (int[] arr)`

zu: ergebnis ist Array mit 10 Einträgen.  
(Integer-Array).

26-9: for - schleife

- läuft 10 mal durch
- ~~evokiert <sup>10</sup> <sup>ein</sup> Array arr für jeden~~  
~~Eintrag in ergebnis - Array und~~
- evokiert Methode "eins" mit Parameter  
`eins(arr, 2)` und speichert das  
Ergebnis im `int-Array ergebnis`
- Ausgeben ~~als~~ mit `println()`.

- Beschreibung von `eins (int)`

- Methode bekommt ein Array und ein `int (Modus)`  
als Parameter

• `Modus == 1`

215-18: Summiert alle Einträge im Array an  
auf und gibt das Ergebnis aus

⇒ KEIN AUFWAND den kein Aufruf  
von Methode zwei.

•  $\text{modus} == 2$

Fallunterscheidung:

→ Fall 1: Array-Länge ist gerade.

• Ausgeben der Summe von  
zwei(array[0]) (erstes Element im array)  
und zwei(array[array.length - 1])  
(letztes Element ~~im~~ im array)

⇒ Aufwand von  $5n + 5n = 10n$

$$O(n) = 10n \quad (*)$$

→ Fall 2: Array-Länge ist ungerade.

~~• Ausgeben der Summe von~~

→ die Methode zwei wird für jeden  
Eintrag im Array ausgewertet und  
anschließend aufsummiert.

→ ~~11 weil Ergebnis ist 1 am Anfang.~~

⇒ Aufwand von  $5n \cdot (\text{array.length})$  ~~44~~

$$O(n) = 5n (\text{array.length}) \quad ~~44~~ \quad (**)$$

•  $\text{modus} == 3$

Ausgeben des Produkts aller Elemente im  
Array ⇒ kein Aufwand.

• sonst

Ausgeben -1 ⇒ kein Aufwand.

(\*\*\*\*) Best-Case:  $\text{modus} \neq 2$  oder  $\text{array.length} == 0$   
 $\Rightarrow$  weil kein Aufruf von zwei bzw.  
 $T(n) = 10n$  mit  $n = 0 \Rightarrow T(n) = 0$ .

(xxx) Worst-Case:  $\text{Modus} == 2$   
AND  $\text{array.length} \% 2 \neq 0$   
AND  $\text{array.length} > 2$   
 $\Rightarrow$  Aufruf der Methode steigt an  
 $5n$  pro  $\text{array.length}$

$\Rightarrow$  void work(int[] arr) wertet arr mit  
eins(arr, 2) aus  $\Rightarrow$  modus 2.

Aufgaben:

1) Zeile 20:  $\text{if}(\text{array.length} \% 2 == 0)$   
Wenn der Ausdruck falsch ist, landet man  
in Zeile 22, mit einer Array-Länge größer  
als 2 ist man im Worst-Case.

2) Worst-Case-Laufzeitabschätzung von work.

• 10 Versuche,  $T(n) = 5n(\text{array.length})$   
 $= 5n^2$

$$\Rightarrow T(n) = 50n^2$$

3) Best-Case-Laufzeitabschätzung von work

• 10 Versuche,  $T(n) = 10n$

$$\Rightarrow T(n) = 100n$$

4) 100 Aufrufe  $\rightarrow$  75 Aufrufe mit ungerader Länge  
25 Aufrufe mit gerader Länge

$$T(n) = (75 \cdot (50n^2) + 25 \cdot (100n)) / 100$$

$$= (3750n^2 + 2500n) / 100 = 37,5n^2 + 25n$$