Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

Programmierung 1
WiSe 25/26
Moritz Balg
01.12.2025

Test 3

- Felder -

Aufgabe 1: Speicherbedarf bestimmen (3 Punkte)

Bestimmen Sie den benötigten Speicherplatz für die nachfolgenden Konstrukte. Vernachlässigen Sie den benötigten Overhead für die Referenzen.

- `int[] i = new int[100]`
- `char[][] c = new char[100][100]`
- `double[][][] d = new double[100][100][100]`

Mögliche Lösung:

- `i` $\Rightarrow 100 \times 4$ Byte $= 400$ Byte
- `c` $\Rightarrow 100 \times 100 \times 2$ Byte $= 20.000$ Byte $= 20$ KB
- `d` $\Rightarrow 100 \times 100 \times 100 \times 8$ Byte $= 8.000.000$Byte $= 8$MB

Ein Punkt pro korrekter Lösung

Aufgabe 2: Array Rotation (10 Punkte)

a) Schreiben Sie eine Funktion `arrayRotation`: $\mathbb{R}^n \rightarrow \mathbb{R}^n, n \in \mathbb{N}$, welche jedes Element in einem Array um eine Position nach rechts verschiebt. Das letzte Element, welches bei der Verschiebung hinausfällt, soll am Anfang eingefügt werden.

Beispiel

Sei $\vec{a} = \{a_1, ..., a_n\} \in R^n$, dann führt die Funktion `arrayRotation` folgende Umformung durch: $\{a_1, a_2, ..., a_n\} \Rightarrow \{a_n, a_1, ..., a_{n-1}\}$

b) Überladen Sie die Methode aus Aufgabe a) mit folgender Signatur:

`arrayRotation`: $\mathbb{R}^n \times \mathbb{N}_0 \rightarrow \mathbb{R}^n, n \in \mathbb{N}$.

Neben dem Vektor wird der Methode eine natürliche Zahl `k` übergeben, welche angibt, wie häufig die Rotation ausgeführt werden soll.

- Nutzen Sie Ihre Teilergebnisse aus Aufgabenteil a).
- Überlegen Sie sich, was passiert, wenn der Wert `k` die Länge des Arrays überschreitet. Behandeln Sie diesen Sonderfall sinnvoll.

Mögliche Lösung:

```
// Datentypen korrekt aus der Signatur entnommen (1P)
public static double[] arrayRotation(double[] arr) {
  double tmp = arr[arr.length-1]; // Zwischenspeichern des Wertes (1P)
  // Array korrekt durchlaufen ohne Werte zu verlieren (2P)
  for(int i = arr.length-1; i > 0; i--) {
    arr[i] = arr[i-1]; // Keine ArrayIndexOutOfBoundsException (2P)
  }
  arr[0] = tmp;
  return arr; // Return Statement korrekt (1P)
}

// Methode korrekt überladen (1P)
// Datentypen korrekt aus der Signatur entnommen (1P)
public static double[] arrayRotation(double[] arr, int n) {
  // Sinnvolle Behandlung des Überlaufs mit %-Operator (1P)
  for(int i = 0; i < n % arr.length; i++) {
    arrayRotation(arr);
  }
  return arr;
}
```