

Arrays: Übungen 2

Gegeben ist die Klasse „messreihe2“ mit einigen Methoden. Bei der Erzeugung einer Instanz des Typs Messreihe wird ein Array mit zufällig generierten „Messwerten“ vom Typ `double` erzeugt. Bearbeite die folgenden Aufgaben.¹⁾

Aufgaben:

- **(1)** Beschreibe, was die Methode `wasDenn()` macht. Erprobe die Methode und erlautere dann anhand des Quelltextes.
- **(2)** Beschreibe zunächst ohne ausprobieren, was die folgenden Aufrufe machen:
 - `wasMachelch(33)` bzw. `wasMachelch(87)` bzw. `wasMachelch()`
 - Teste deine Antworten
- **(3)** Erstelle je eine Methode,
 - a) die in das Element mit Index 4 den Gewichtswert 44.4444 einträgt.
 - b1) die die Summe der Gewichte der Elemente an den Indizes 16,17,18 zurückgibt.
 - b2) die die Summe der Gewichte der letzten 33 Elemente zurückgibt.
 - c) die in ein Element, dessen Index man vorgeben kann, den Gewichtswert 55.55555 einträgt.
 - d) die in ein Element, dessen Index man vorgeben kann, einen Gewichtswert einträgt, den man vorgeben kann.
- Teste deine Methoden jeweils
- **(4)** Die Methode `gibGewicht` hat eine Absicherung gegen Fehleingaben.
- Welches Problem gibt es fuer die Fehlermeldung bei Fehleingaben? Wie lösen Sie es hier bzw. allgemein?

Zusatzaufgaben:

- **(5)** Erstelle eine Methode,
 - a) die in drei (17) aufeinanderfolgende Elemente, deren Startindex man vorgeben kann, Gewichtswerte eintraegt, die jeweils um 1.2 ansteigen.
 - b) die die Gewichtswerte zweier Elemente miteinander vertauscht.
 - c) die zwei benachbarte Elemente findet, die sich um nicht mehr als 3.3 unterscheiden. (Überlege, was du sinnvoll als Antwort zurueckmeldest).
 - d) die die zwei benachbarten Elemente mit dem größten Gesamtgewicht findet.
- **(6)** Finde die vier benachbarten Elemente der Messreihe, die das größte Gesamtgewicht haben. Lasse die Elemente sowie das Gesamtgewicht ausgeben.
- **(7)** Finde die vier benachbarten Elemente der Messreihe, die sich untereinander am wenigsten unterscheiden. Welche Antwort sollte geliefert werden?

Ohne BlueJ

App.java

```
/** Fachklasse: Messreihe (= eine Reihe von nummerierten Messdaten)
 * @author: thh
 * @version: Aug.16/Mai 14
```

```
*/

public class Messreihe {
    // Objektvariablen deklarieren
    int anzahl = 45;
    double[] gewicht = new double[anzahl];

    /** Konstruktor fuer Objekte der Klasse Messreihe
     *  jede Messreihe enthaelt eine Reihe von positiven Messdaten
     (Gewichten)
     */
    public Messreihe() {
        for (int i=0; i<anzahl; i++) {    // Alle Gewichte
            gewicht[i] = erzeugeZZahl(); // der Reihe nach festlegen
        }
    }

    /** das Element der Reihung mit dem Index i zurueckgeben
     *  Der gewuenschte Index i muss eingegeben werden
     *  Bei Eingabe eines nicht vorhandenen Index wird
     *  -8.888 als Fehlersignal zurueckgemeldet */
    public double gibGewicht(int i) {
        if (i<0 || i>anzahl) {            //<-- 4.
            return -8.888;    // als Fehlersignal!
        }
        else {
            return gewicht[i];
        }
    }

    /** Beschreiben Sie in Worten, was diese Methode
     *  mit den Gewichten der Messreihe macht. */    //<-- 1.
    public void wasDenn() {
        for (int i = 8; i<15; i++) {
            gewicht[i] = 77.7;
        }

        for (int i=20; i<27; i++) {
            gewicht[i]    = 3*i;
            gewicht[i+1] = 4*i;
        }
    }

    /** Beschreibe in Worten, was diese Methode
     *  mit den Gewichten der Messreihe macht. */    //<-- 2.
    public void wasMacheIch(int ab) {
        if (ab<40 && ab>30) {
            for (int i=ab; i<=ab+5; i++) {
```

```
        gewicht[i-20] = gewicht[i];
    }
}

// ----- Hilfsfunktionen
/** dient zum Anzeigen der Reihung am Bildschirm;
 * kann durch INSPECT ersetzt werden */
public void anzeigen() {
    for (int i=0; i< anzahl; i++) {
        System.out.println(formatiere(i)+" : "+gewicht[i]);
    }
}

//----- interne Hilfsfunktionen
/** interne Methode, um eine Zufallszahl im Bereich 200.0 - 799.999
 * mit 3 Nachkommastellen zu erzeugen;
 * Math.random() liefert eine Zahl von 0 (inkl.) bis 1 (exkl.) */
private double erzeugeZZahl() {
    double zufZahl = 200 + 600*Math.random();
    return Math.round((zufZahl*1000))/1000.0;
}

/** interne Hilfsfunktion zum stellenrichtigen
 * Ausdruck von ein- bis zweistelligen Zahlen. */
private String formatiere(int i) {
    String erg = "Index";
    if (i<10) {
        erg = "Index  " + i;    // Zwei Leerzeichen drin !!
    }
    else {
        erg = "Index " + i;    // hier nur eines !!
    }
    return erg;
}
}

/* App Klasse. Steuerklasse für unser Programm */
public class App {

    public static void main(String[] args) {
        Messreihe reihe1 = new Messreihe();
        reihe1.anzeigen();
    }
}

/* Aufgaben:
 * 1. Beschreibe, was wasDenn() macht.
 * Erprobe diese Methode und erlautere dann anhand des Quelltextes.
```

```
*  
* 2. Beschreibe, was diese Aufrufe machen:  
*   wasMacheIch(33) bzw. wasMacheIch(87) bzw. wasMacheIch()  
*   Teste deine Antworten  
*  
* 3. Erstelle je eine Methode,  
*   a) die in das Element mit Index 4 den Gewichtswert 44.4444  
eintraegt.  
*   b) die die Summe der Gewichte der Elemente an den  
*       Indizes 16,17,18 zurueckgibt.  
*   b2) die die Summe der Gewichte der letzten 33 Elemente  
zurueckgibt.  
*   c) die in ein Element, dessen Index man vorgeben kann, den  
*       Gewichtswert 55.55555 eintraegt.  
*   d) die in ein Element, dessen Index man vorgeben kann,  
*       einen Gewichtswert eintraegt, den man vorgeben kann.  
*   Teste deine Methoden.  
*  
* 4. Die Methode "gibGewicht" hat eine Absicherung gegen Fehleingaben.  
*   Welches Problem gibt es fuer die Fehlermeldung bei Fehleingaben?  
*   Wie loesen Sie es hier bzw. allgemein?  
*  
*/  
  
/*  
*   Zusatzaufgaben:  
*  
*   // ---Z -----  
*  
* 5. Erstellen Sie eine Methode,  
*   a) die in drei (17) aufeinanderfolgende Elemente, deren Startindex  
man  
*       vorgeben kann, Gewichtswerte eintraegt, die jeweils um 1.2  
ansteigen.  
*   b) die die Gewichtswerte zweier Elemente miteinander vertauscht.  
(vgl. Folien)  
*  
*  
*   //---- Z* -----  
*   c) die zwei benachbarte Elemente findet, die sich um nicht mehr als  
3.3  
*       unterscheiden.  
*       Überlegen Sie, was Sie sinnvoll als Antwort zurueckmelden.  
*   d) ** die die zwei benachbarten Elemente findet mit dem größten  
Gesamtgewicht.  
*  
*   //---- Z** -----  
*   Finden Sie die vier benachbarten Elemente der Messreihe, die das  
größte
```

```
* Gesamtgewicht haben. Lassen Sie die Elemente sowie das  
Gesamtgewicht angeben.  
* ** Finden Sie die vier benachbarten Elemente der Messreihe, die  
sich untereinander  
* am wenigsten unterscheiden. Welche Antwort liefern Sie?  
*  
*/
```

1)

Diese Übungen stehen unter einer CC-BY-SA Lizenz, sie wurden erstellt in enger Anlehnung an das Material der ZPG BW/Heußner

From:
<https://wiki.qg-moessingen.de/> - QG Wiki

Permanent link:
<https://wiki.qg-moessingen.de/faecher:informatik:oberstufe:java:algorithmen:arrays:uebungen2:start?rev=1615827728>

Last update: **15.03.2021 18:02**

