

Aufgabe 3: Suchen

(Methode „sucheBinaer()“)

Stichwörter: Binäre Suche

Aufgabe 3: Suchen

Für diese Aufgabe wird die Vorlage Suchalgorithmen benötigt, die auf dem Beiblatt genauer erklärt wird.

Vervollständigen Sie die Methode `sucheBinaer()`. Die fertige Methode soll in der Lage sein, beliebige Werte in beliebigen sortierten Arrays zu suchen. Im gelben Textfeld des Eingabefensters soll dabei ausführlich und nachvollziehbar angezeigt werden, wie die Methode vorgeht. Beispielsweise so:

Führe die Methode `sucheSequenziell()` aus:
Suche in diesem Feld: 3, 5, 7, 17, 42, 23
Überprüfen des Werts an der Position 0
Überprüfen des Werts an der Position 1
Überprüfen des Werts an der Position 2
Überprüfen des Werts an der Position 3
Fertig :-)

Führe die Methode `sucheBinaer()` aus:
Suche in diesem Feld: 3, 5, 7, 17, 42, 23
Suchbereich: 0 bis 5
Mitte: 2, also neuer Bereich: 3 bis 5
Mitte: 4, also neuer Bereich: 3 bis 3
Mitte: 3, Treffer : -)

(a) Sequenzielle Suche

Lösungsvorschlag

```
}

/**
 * Sequenzielle Suche: Durchsucht das Array nach dem Wert und gibt dessen
 * Position als Ergebnis zurück.
 *
 * @param array Ein Feld mit Zahlen.
 * @param wert Die Zahl, die gesucht werden soll.
 *
 * @return Die Indexnummer der gesuchten Zahl.
 */
public int sucheSequenziell(int[] array, int wert) {
    fenster.schreibeZeile("\nFühre die Methode sucheSequenziell() aus:");
    fenster.schreibe("Suche in diesem Feld: ");
    fenster.schreibeArray(array);
    fenster.schreibeZeile("");
    // Wiederhole für alle Elemente des Arrays:
    for (int i = 0; i < array.length; i++) {
        fenster.schreibeZeile("Überprüfen des Werts an der Position " + i);
        // Wenn das Element an der Stelle i der gesuchte Wert ist:
        if (array[i] == wert) {
            fenster.schreibeZeile("Fertig :-)");
            // Gib die Position i als Ergebnis zurück:

```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/aufgaben/aud/ab_2/Suchalgorithmen.java](https://github.com/bschlangaul/aufgaben/aud/ab_2/Suchalgorithmen.java)

(b) Binäre Suche

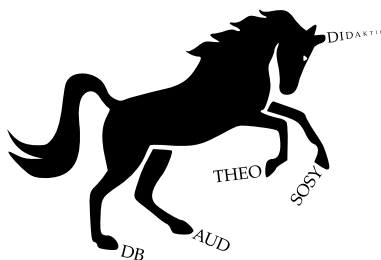
```

    }
}
fenster.schreibeZeile("Nichts gefunden :-(");
// Gib den Sonderfalls -1 (nichts gefunden) als Ergebnis zurück:
return -1;
}

/**
 * Binäre Suche: Durchsucht das Array nach dem Wert und gibt dessen Position als
 * Ergebnis zurück.
 *
 * @param array Ein Feld mit Zahlen.
 * @param wert Die Zahl, die gesucht werden soll.
 *
 * @return Die Indexnummer der gesuchten Zahl.
 */
public int sucheBinaer(int[] array, int wert) {
    fenster.schreibeZeile("\nFühre die Methode sucheBinaer() aus:");
    fenster.schreibe("Suche in diesem Feld: ");
    fenster.schreibeArray(array);
    fenster.schreibeZeile("");
    int u = 0;
    int o = array.length - 1;
    fenster.schreibeZeile("Suchbereich: " + u + " bis " + o);
    while (u <= o) {
        int m = (u + o) / 2;
        if (array[m] == wert) {
            fenster.schreibe("Mitte: " + m);
            fenster.schreibeZeile(", Treffer : -) ");
            return m;
        } else if (array[m] > wert) {

```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/aufgaben/aud/ab_2/Suchalgorithmen.java](https://github.com/bschlangaul/aufgaben/blob/main/java/org/bschlangaul/aufgaben/aud/ab_2/Suchalgorithmen.java)



Die Bschlangaul-Sammlung

Hermine Bschlangaul and Friends

Eine freie Aufgabensammlung mit Lösungen von Studierenden für Studierende zur Vorbereitung auf die 1. Staatsexamensprüfungen des Lehramts Informatik in Bayern.



Diese Materialsammlung unterliegt den Bestimmungen der Creative Commons Namensnennung-Nicht kommerziell-Share Alike 4.0 International-Lizenz.

Hilf mit! Die Hermine schafft das nicht allein! Das ist ein Community-Projekt! Verbesserungsvorschläge, Fehlerkorrekturen, weitere Lösungen sind herzlich willkommen - egal wie - per Pull-Request oder per E-Mail an hermine.bschlangaul@gmx.net. Der TeX-Quelltext dieses Dokuments kann unter folgender URL aufgerufen werden: https://github.com/bschlangaul-sammlung/examens-aufgaben/blob/main/Module/30_AUD/30_Suchalgorithmen/Aufgabe_Methode-sucheBinaer.tex