

Aufgabe 2: Sortieren

(Sortier-Vorlage)

Stichwörter: Selectionsort, Bubblesort

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

Die fertigen Methoden sollen in der Lage sein, beliebige Arrays zu sortieren. Im gelben Textfeld des Eingabefensters soll dabei wieder ausführlich und nachvollziehbar angezeigt werden, wie die jeweilige Methode vorgeht. Beispielsweise so:

```
Führe die Methode selectionSort() aus:
Sortiere dieses Feld: 5, 3, 17, 7, 42, 23
Der Marker liegt bei: 5
Das Maximum liegt bei: 4
Diese beiden Elemente werden nun vertauscht.
Ergebnis dieser Runde: 5, 3, 17, 7, 23, 42
Der Marker liegt bei: 4
Das Maximum liegt bei: 4
Diese beiden Elemente werden nun vertauscht.
Ergebnis dieser Runde: 5, 3, 17, 7, 23, 42
Der Marker liegt bei: 3
Das Maximum liegt bei: 2
Diese beiden Elemente werden nun vertauscht.
Ergebnis dieser Runde: 5, 3, 7, 17, 23, 42
Der Marker liegt bei: 2
Das Maximum liegt bei: 2
Diese beiden Elemente werden nun vertauscht.
Ergebnis dieser Runde: 5, 3, 7, 17, 23, 42
Der Marker liegt bei: 1
```

```
Das Maximum liegt bei: 0
Diese beiden Elemente werden nun vertauscht.
Ergebnis dieser Runde: 3, 5, 7, 17, 23, 42
Der Marker liegt bei: 0
Das Maximum liegt bei: 0
Diese beiden Elemente werden nun vertauscht.
Ergebnis dieser Runde: 3, 5, 7, 17, 23, 42
```

```
Führe die Methode bubbleSort() aus:
Sortiere dieses Feld: 5, 3, 17, 7, 42, 23
Das Element an der Stelle 0 ist größer als sein Nachfolger.
Diese beiden werden nun vertauscht.
Ergebnis dieser Runde: 3, 5, 17, 7, 42, 23
Das Element an der Stelle 2 ist größer als sein Nachfolger.
Diese beiden werden nun vertauscht.
Ergebnis dieser Runde: 3, 5, 7, 17, 42, 23
Das Element an der Stelle 4 ist größer als sein Nachfolger.
Diese beiden werden nun vertauscht.
Ergebnis dieser Runde: 3, 5, 7, 17, 23, 42
```

(a) Vervollständige die Methode `selectionSort()`.

Lösungsvorschlag

```
/**
 * SelectionSort: Sortieren durch Selektion.
 *
 * @param array Ein Feld mit Zahlen.
 *
 * @return Ein sortiertes Feld mit Zahlen.
 */
public int[] selectionSort(int[] array) {
    fenster.schreibeZeile("\nFühre die Methode selectionSort() aus:");
    fenster.schreibe("Sortiere dieses Feld: ");
    fenster.schreibeArray(array);
    fenster.schreibeZeile("");
    int marker = array.length - 1;
    while (marker >= 0) {
        int max = 0;
        for (int i = 0; i <= marker; i++) {
            if (array[i] > array[max]) {
                max = i;
            }
        }
        fenster.schreibeZeile("Der Marker liegt bei: " + marker);
        fenster.schreibeZeile("Das Maximum liegt bei: " + max);
        fenster.schreibeZeile("Diese beiden Elemente werden nun vertauscht.");
        swap(array, marker, max);
        fenster.schreibe("Ergebnis dieser Runde: ");
        fenster.schreibeArray(array);
        fenster.schreibeZeile("");
        marker--;
    }
}
```

```

    }
    return array;
}

```

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

(b) Vervollständige die Methode `bubbleSort()`.

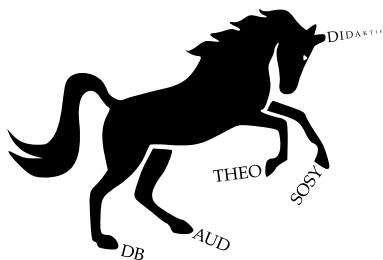
Lösungsvorschlag

```

/**
 * BubbleSort: Sortieren durch Vertauschen.
 *
 * @param array Ein Feld mit Zahlen.
 *
 * @return Ein sortiertes Feld mit Zahlen.
 */
public int[] bubbleSort(int[] array) {
    fenster.schreibeZeile("\nFühre die Methode bubbleSort() aus:");
    fenster.schreibe("Sortiere dieses Feld: ");
    fenster.schreibeArray(array);
    fenster.schreibeZeile("");
    boolean swapped;
    do {
        swapped = false;
        for (int i = 0; i < array.length - 1; i++) {
            if (array[i] > array[i + 1]) {
                fenster.schreibe("Das Element an der Stelle " + i);
                fenster.schreibeZeile(" ist größer als sein Nachfolger.");
                fenster.schreibeZeile("Diese beiden werden nun vertauscht.");
                swap(array, i, i + 1);
                fenster.schreibe("Ergebnis dieser Runde: ");
                fenster.schreibeArray(array);
                fenster.schreibeZeile(" ");
                swapped = true;
            }
        }
    } while (swapped);
    return array;
}

```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/aufgaben/aud/ab_2/Sortieralgorithmen.java](https://github.com/bschlangaul/aufgaben/aud/ab_2/Sortieralgorithmen.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.bsclangaul@gmx.net. Der \LaTeX -Quelltext dieser Aufgabe kann unter folgender URL aufgerufen werden: https://github.com/bsclangaul-sammlung/examens-aufgaben-tex/blob/main/Module/30_AUD/40_Sortieralgorithmen/Aufgabe_Sortier-Vorlage.tex