

Aufgabe 2: Sortieren

Für diese Aufgabe wird die Vorlage Sortieralgorithmen benötigt, die auf dem Beiblatt genauer erklärt wird.¹

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()`.

```
65     fenster.schreibe("Ergebnis dieser Runde: ");
66     fenster.schreibeArray(array);
67     fenster.schreibeZeile("");
68 }
69
70 return array;
71 }
72
73 /**
74  * SelectionSort: Sortieren durch Selektion.
75  *
76  * @param array Ein Feld mit Zahlen.
77  *
78  * @return Ein sortiertes Feld mit Zahlen.
79  */
80 public int[] selectionSort(int[] array) {
81     fenster.schreibeZeile("\nFühre die Methode selectionSort() aus:");
82     fenster.schreibe("Sortiere dieses Feld: ");
83     fenster.schreibeArray(array);
84     fenster.schreibeZeile("");
85     int marker = array.length - 1;
86     while (marker >= 0) {
87         int max = 0;
88         for (int i = 0; i <= marker; i++) {
89             if (array[i] > array[max]) {
90                 max = i;
91             }
92         }
93     }
94 }
```

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

¹Diese Aufgabe stammt aus dem Übungsblatt 1 zu Algorithmen und Datenstrukturen von Prof. Dr. Martin Hennecke und Rainer Gall an der Universität Würzburg und wurde dankenswerterweise zur Verwendung in diesem Aufgabenblatt zur Verfügung gestellt.

```

93     fenster.schreibeZeile("Der Marker liegt bei: " + marker);
94     fenster.schreibeZeile("Das Maximum liegt bei: " + max);
95
96     ↪ fenster.schreibeZeile("Diese beiden Elemente werden nun vertauscht.");
97     swap(array, marker, max);
98     fenster.schreibe("Ergebnis dieser Runde: ");
99     fenster.schreibeArray(array);
100    fenster.schreibeZeile("");
101    marker--;
102  }
103  return array;
104 }
105
106 /**
107  * BubbleSort: Sortieren durch Vertauschen.
108  *
109  * @param array Ein Feld mit Zahlen.
110  *
111  * @return Ein sortiertes Feld mit Zahlen.
112  */
113 public int[] bubbleSort(int[] array) {
114     fenster.schreibeZeile("\nFühre die Methode bubbleSort() aus:");
115     fenster.schreibe("Sortiere dieses Feld: ");
116     fenster.schreibeArray(array);
117     fenster.schreibeZeile("");
118     boolean swapped;
119     do {

```