

BubbleSort: Blasensortierung

Weiterführende Literatur:

- Algorithmen und Datenstrukturen: Tafelübung 11, WS 2018/19, Seite 43
- Saake und Sattler, Algorithmen und Datenstrukturen, Seite 129-131 (PDF 147-149)
- Wikipedia-Artikel „Bubblesort“

3	2	5	1	4
---	---	---	---	---

1. Durchgang

3	2	5	1	4
---	---	---	---	---

2	3	5	1	4
---	---	---	---	---

2	3	1	5	4
---	---	---	---	---

2. Durchgang

2	3	1	4	5
---	---	---	---	---

2	1	3	4	5
---	---	---	---	---

3. Durchgang

2	1	3	4	5
---	---	---	---	---

fertig

1	2	3	4	5
---	---	---	---	---

Funktionsweise

- solange zu sortierende Liste nicht vollständig sortiert ist:
- iteriere von vorne über die Elemente
- falls zwei aufeinanderfolgende Elemente nicht sortiert sind: vertauschen
- fertig, wenn während einer Runde keine Vertauschung passiert ist

Eigenschaften

- Laufzeitkomplexität:
 - $\mathcal{O}(n)$ (im Best-Case)
 - $\mathcal{O}(n^2)$ (im Average- und Worst-Case)
- stabil
- in-situ

```

10  /**
11   * Sortiere mit Hilfe des Bubblesort-Algorithmus.
12   *
13   * <p><strong>Abkürzungen:</strong></p>
14   *
15   * <ul>
16   * <li>t: getauscht
17   * </ul>
18   */
19  public int[] sortiere() {
20      boolean t;
21      do {
22          t = false;
23          for (int i = 0; i < a.length - 1; i++) {
24              if (a[i] > a[i + 1]) {
25                  vertausche(i, i + 1);
26                  t = true;
27              }
28          }
29      } while (t);
30      return a;
31  }

```

Code-Beispiel auf Github ansehen: [src/main/java/org/beschlangaul/sortier/BubbleMinimal.java](https://github.com/beschlangaul/sortier/BubbleMinimal.java)

```

9  /**
10   * Sortiere ein Zahlen-Feld mit Hilfe des Bubblesort-Algorithmus.
11   *
12   * @return Das sortierte Zahlenfeld.
13   */
14  public int[] sortiere() {
15      boolean getauscht;
16      do {
17          getauscht = false;
18          for (int i = 0; i < zahlen.length - 1; i++) {
19              if (zahlen[i] > zahlen[i + 1]) {
20                  // Elemente vertauschen
21                  vertausche(i, i + 1);
22                  getauscht = true;
23              }
24          }
25          // solange Vertauschung auftritt
26      } while (getauscht);
27      return zahlen;
28  }

```

Code-Beispiel auf Github ansehen: [src/main/java/org/beschlangaul/sortier/BubbleIterativ.java](https://github.com/beschlangaul/sortier/BubbleIterativ.java)

1

```

11  */
12  private void sortiereRekursiv(int[] zahlen, int letztes) {
13      // Bei einem leeren Feld kann die Variable letztes -1 annehmen.
14      // Wenn letztes 0 ist, dann sind die Zahlen sortiert.
15      if (letztes <= 0)
16          return;
17
18      // Ein Durchgang der BlasenSortierung. Nach diesem Durchgang befindet sich
19      // ↳ das größte Element am Ende des Felds.
20      for (int i = 0; i < letztes; i++)

```

¹Saake und Sattler, *Algorithmen und Datenstrukturen*, Seite 130-131.

```

21     if (zahlen[i] > zahlen[i + 1]) {
22         vertausche(i, i + 1);
23     }
24
25     // Sortiere jetzt die restlichen Elemente.
26     sortiereRekursiv(zahlen, letztes - 1);
27 }
28
29 public int[] sortiere() {
30     sortiereRekursiv(zahlen, zahlen.length - 1);
31     return zahlen;

```

Code-Beispiel auf Github ansehen: [src/main/java/org/beschlangaul/sortier/BubbleRekursiv.java](https://github.com/orgs/beschlangaul/sortier/BubbleRekursiv.java)

Literatur

- [1] *Algorithmen und Datenstrukturen: Tafelübung 11, WS 2018/19.* https://www.studon.fau.de/file2567217_download.html. FAU: Lehrstuhl für Informatik 2 (Programmiersysteme).
- [2] Gunter Saake und Kai-Uwe Sattler. *Algorithmen und Datenstrukturen. Eine Einführung in Java.* 2014.
- [3] *Wikipedia-Artikel „Bubblesort“.* <https://de.wikipedia.org/wiki/Bubblesort>.