

## Didaktische Hinweise

### Zeichenkettenverarbeitung mit Processing<sup>1</sup>

#### Zielgruppe

Der Leitfaden zur Zeichenkettenverarbeitung in Processing richtet sich an Schülerinnen und Schüler in der Einführungsphase, die bereits mit den algorithmischen Kontrollstrukturen und den Grundlagen von Processing vertraut sind. Dieser Leitfaden bietet sich zum Beispiel im Anschluss auf den Leitfaden zum Einstieg in Processing an.

#### Voraussetzungen

Es wird vorausgesetzt, dass die Schülerinnen und Schüler bereits Eingaben des Anwenders mithilfe der Bibliothek *javax.swing.JOptionPane.\**; und der Methode *showInputDialog()* erfragt haben. Weiterhin sollten die Schülerinnen und Schüler die Methode *keyPressed()* und die Systemvariable *key* verwenden können, um auf das Ereignis, dass eine Taste auf der Tastatur gedrückt wurde, reagieren zu können. Weiterhin sollten die Schülerinnen und Schüler Ausgaben im Programmfenster, z.B. mithilfe der Methode *text()* erzeugen können. Wenn die Schülerinnen und Schüler zuvor mit dem Leitfaden für den Einstieg in Processing gearbeitet haben, sollten sie über diese Kompetenzen verfügen.

#### Lernziele

Dieser Leitfaden führt in die Zeichenkettenverarbeitung mithilfe elementarer Zeichenkettenoperationen ein. Die Schülerinnen und Schüler lernen ...

1. ... Benutzereingaben zu verwenden.
2. ... Texte in der Konsole auszugeben.
3. ... auf einzelne Zeichen zuzugreifen.
4. ... eine Zeichenkette Zeichen für Zeichen zu durchlaufen und zu bearbeiten.
5. ... die ASCII-Codierung der Zeichen für die Bearbeitung zu verwenden.

#### Didaktische Hinweise

Viele der Übungsaufgaben stammen aus dem Bereich der Kryptographie, so dass sich dieser Leitfaden gut mit dem Erlernen der Kompetenzen aus dem Modul Kryptologie kombinieren lässt.

Die Vorgaben für das Abitur<sup>2</sup> sehen weiterhin die Verwendung von Methoden vor, die ein Zeichen an einer übergebenen Position ersetzen und Zeichenketten lexikographisch vergleichen. Da zum Verändern einer Zeichenkette hier die Strategie, diese zeichenweise zu durchlaufen und parallel die veränderte Zeichenkette aufzubauen, zum Einsatz kommt, wird auf eine entsprechende Operation zum Ersetzen von Zeichen verzichtet. Diese kann zu einem späteren Zeitpunkt bei Bedarf eingeführt werden. Da Processing bzw. Java eine solche Methode nicht zur Verfügung stellen, kann diese im

---

<sup>1</sup> Die Programmierumgebung Processing wurde 2001 von Ben Fry und Casey Reas initiiert. Nähere Informationen finden Sie unter <https://processing.org/>

<sup>2</sup> Niedersächsisches Kultusministerium (Hrsg.) (2017) Ergänzende Hinweise zum Kerncurriculum Informatik für die gymnasiale Oberstufe am Gymnasium, an der Gesamtschule sowie für das Kolleg. [http://db2.nibis.de/1db/cuvo/datei/INF\\_Ergaenzende-Hinweise\\_GO\\_2018.pdf](http://db2.nibis.de/1db/cuvo/datei/INF_Ergaenzende-Hinweise_GO_2018.pdf) [Datum des Zugriffs: 18.07.2019]

Zuge der Einführung von eigenen Methoden selbst erstellt werden (s. Aufgabe 6 im Leitfaden zum Thema „Eigene Methoden“). Bei den vorliegenden Aufgaben wäre sie nur bedingt hilfreich. Das hier verwendete Konzept ist allgemeiner und kann auf eine größere Klasse von Problemen angewendet werden.

Eine lexikographische Ordnung von Zeichenketten erscheint erst sinnvoll, wenn mit mehreren Zeichenketten beispielsweise in einer Reihung gearbeitet wird. Die entsprechenden Java-Methoden *compareTo()* bzw. *compareToIgnoreCase()* können daher im Zusammenhang mit der Einführung von Reihungen eingeführt werden, ggf. erst in der Oberstufe.

Wenn einige Schülerinnen und Schüler eine grafische Sprache bevorzugen, kann das Thema Zeichenkettenverarbeitung parallel in *Snap!*<sup>3</sup> bearbeitet werden. Denn *Snap!* stellt die in diesem Leitfaden verwendeten Methoden als Bausteine ebenfalls zur Verfügung stellt.

## Die Aufgaben

Die Aufgaben üben und festigen die vorhergehenden Erläuterungen. Der Schwierigkeitsgrad variiert. Die mit einem Sternchen versehenen Aufgaben haben ein besonders hohes Anforderungsniveau und eignen sich nur für besonders leistungsstarke Schülerinnen und Schüler.

Der Leitfaden mündet in einem Projekt, in dem die Schülerinnen und Schüler alle erlernten Kompetenzen in Rahmen einer etwas umfangreicheren Problemlösung erproben und anwenden können. Dabei können die prozessbezogenen Kompetenzen aus dem Bereich Kreatives Schaffen und Problemlösen besonders gefördert werden<sup>4</sup>.

Musterlösungen können den Ordnern zu der jeweiligen Aufgabe entnommen werden. Für die korrekte Ausführbarkeit wird keine Garantie übernommen. Auch für Folgeschäden, die sich aus der Anwendung der Quelltexte für die Musterlösungen oder Beispiele des Leitfadens oder durch eventuelle fehlerhafte Angaben ergeben, wird keine Haftung oder juristische Verantwortung übernommen.

## Ausblick

Im Anschluss an diesen Leitfaden bietet sich z. B. eine Gegenüberstellung der Operationen und Algorithmen zur Zeichenkettenverarbeitung in einer graphischen Programmiersprache und in Processing an. An diesem Beispiel lässt sich besonders gut verdeutlichen, dass die algorithmischen Konzepte die gleichen sind. Daraus ergibt sich der Vorteil einer programmiersprachenunabhängigen Notation von Algorithmen z. B. in Form von Struktogrammen. Alternativ kann auch mit der Einführung eigener Methoden begonnen werden.

## Lizenz

Dieses Werk ist lizenziert unter einer [Creative Commons Namensnennung - Nicht kommerziell - Keine Bearbeitungen 4.0 International Lizenz](#). Sie erlaubt Download und Weiterverteilung des vollständigen Werkes unter Nennung meines Namens, jedoch keinerlei Bearbeitung oder kommerzielle Nutzung.

<sup>3</sup> Snap! wird von der University of California, Berkeley zur Verfügung gestellt: <https://snap.berkeley.edu>

<sup>4</sup> vgl. Niedersächsisches Kultusministerium (Hrsg.) (2017) Kerncurriculum für das Gymnasium - gymnasiale Oberstufe, die Gesamtschule – gymnasiale Oberstufe, das Kolleg. Informatik. Hannover: unidruck