

Reihungen in Processing¹

Didaktische Hinweise

Zielgruppe

Die Materialien richten sich an Schülerinnen und Schüler in der Qualifikationsphase. Sie eignen sich sowohl für Kurse auf grundlegendem als auch auf erhöhtem Anforderungsniveau. Bei den Aufgaben kann ggf. eine geeignete Auswahl getroffen werden.

Voraussetzungen

Es wird davon ausgegangen, dass die Lernenden sich die Kompetenzen, die im Lernfeld *Algorithmisches Problemlösen* für die Einführungsphase vorgesehen sind, ebenfalls mit der Programmierumgebung Processing erarbeitet haben. Dazu gehören u. a. der Umgang mit Zeichenketten und eigenen Methoden. Die vorliegenden Materialien bauen damit auf dem Materialpaket „Einstieg in die textbasierte Programmierung mit Processing“ auf. Einige Aufgaben eignen sich auch zur Umsetzung unabhängig von Processing, z. B. als JFrame mit dem JavaEditor².

Lernziele

Anhand des vorliegenden Leitfadens können sich die Lernenden den Umgang mit ein- und zweidimensionalen Reihungen erarbeiten. Neben dem Verständnis der Struktur sollen auch Probleme unter Verwendung von ein- und zweidimensionalen Reihungen eigenständig gelöst werden. Die Aspekte Suchen in und Sortieren von Reihungen werden dabei nicht explizit thematisiert. Es sind jedoch Aufgaben enthalten, die entsprechende Probleme stellen, so dass hier ein entsprechender Exkurs erfolgen und entsprechende Algorithmen selbst entwickelt bzw. angewendet werden können. Zu dem Thema *Suchen und Sortieren* steht ein gesondertes Materialpaket zur Verfügung.

Im Fokus stehen somit die folgenden Kompetenzen für die Qualifikationsphase aus dem Lernfeld *Algorithmisches Problemlösen* des niedersächsischen Kerncurriculums für die Sekundarstufe II³:

Die Schülerinnen und Schüler

- erläutern das Prinzip, mehrere Daten des gleichen Typs in Reihungen zu verwalten [...].
- entwerfen und implementieren Algorithmen unter Verwendung von ein- und zweidimensionalen Reihungen.

Hinweise zu einzelnen Aufgaben

In **Aufgabe 5** und **7** bietet sich auch die Verwendung von zweidimensionalen Reihungen an. Die Aufgaben werden daher in **Aufgabe 16** im Abschnitt zu den zweidimensionalen Reihungen wieder aufgegriffen.

Aufgabe 5c erfordert das Anpassen der vorgegebenen Methode und ist vom Anspruch her eher für leistungsstarke Schülerinnen und Schüler gedacht.

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

² Gerhard Röhner. JavaEditor. <https://javaeditor.org/doku.php?id=start> [Datum des Zugriffs: 13.01.2023]

³ Niedersächsisches Kultusministerium (Hrsg.) (2017) Kerncurriculum für das Gymnasium - gymnasiale Oberstufe, die Gesamtschule – gymnasiale Oberstufe, das Kolleg. Informatik. Hannover: unidruck

[Aufgabe 10](#) und [Aufgabe 20](#) knüpfen an das Thema Kryptologie aus der Einführungsphase an, indem sie die Implementierung eines monoalphabetischen Substitutionsverfahrens bzw. einer Transposition vorsehen. Damit kann die Aufgabe auch als Einstieg in die Vertiefung des Themas Kryptologie für Kurse auf erhöhtem Anforderungsniveau dienen, da hier im KC der Entwurf und die Implementierung eines symmetrischen Verfahrens explizit gefordert werden.

[Aufgabe 11](#) und [Aufgabe 19](#) greifen ebenfalls den Kontext Kryptologie bzw. Fragestellungen aus dem Bereich Kryptoanalyse auf. [Aufgabe 11 e\)](#) erfordert eine Sortierung. Die Schwierigkeit ist hier allerdings, dass die vorliegende Reihung zwar das Kriterium (die Anzahl der Zeichen) enthält, nach dem sortiert werden soll. Das Ergebnis soll aber eine Reihung von Zeichen sein, die nach ihrer Häufigkeit sortiert sind. Diese Aufgabe richtet sich daher eher an leistungsstarke Schüler*innen. [Aufgabe 19](#) greift diesen Aufgabenteil nach der Einführung zweidimensionaler Reihungen noch einmal auf. Werden der ASCII-Wert des Zeichens und die Häufigkeit in einer zweidimensionalen Reihung abgelegt, kann die Aufgabe mit bekannten Sortierverfahren gelöst oder ein passendes entwickelt werden.

Die [Aufgaben 9](#) sowie [17 b](#) und [c](#) thematisieren das Suchen in einer Reihung.

[Aufgabe 20](#) und [Aufgabe 21](#) stellen verschiedene Anwendungskontexte dar, in denen das Vertauschen von Werten innerhalb einer zweidimensionalen Reihung eine Rolle spielt. Aufgabenteil 21d) bezieht auch Nachbarfelder in einer zweidimensionalen Reihung mit ein.

Anhand von [Aufgabe 21](#) kann auch das Kopieren einer Reihung thematisiert werden. Da es sich hier um einen Referenzdatentypen handelt, müssen alle Werte einer Reihung sukzessive in eine neue Reihung kopiert werden, um eine echte Kopie zu erstellen. Wird eine Reihung lediglich einer zweiten Variablen zugewiesen, zeigen beide Variablen auf die gleiche Reihung, so dass sich Änderungen in der Reihung auf beide Variablen auswirken.

Die [Aufgaben 8, 9, 10, 17, 19](#) und [20](#) erfordern keine grafische Ausgabe, so dass sie auch unabhängig von Processing bearbeitet werden können. Da größtenteils die Ein- und Ausgabe von Texten erforderlich ist, bietet sich z. B. auch die Umsetzung als JFrame mit einer GUI aus Textfeldern, Labeln und Buttons an.

Da in Niedersachsen ab 2027 in den Abiturprüfungen auch die Verwendung einer *for-each*-Schleife zum systematischen Durchlaufen einer Reihung explizit zugelassen ist⁴, wird dieses Schleifenkonzept sowohl für ein- als auch für zweidimensionale Reihungen kurz thematisiert. Dabei ist zu beachten, dass eine *for-each*-Schleife in Processing / Java die Elemente einer Reihung temporär in einer Variablen speichert und dadurch zwar das Auslesen der Elemente ermöglicht, nicht aber das Verändern der Elemente in der Reihung. Da auf eine Zählvariable in der Schleife verzichtet wird, steht auch die Position des Elements nicht zur Verfügung. Alle Aufgaben können mit einer klassischen for-Schleife mit Zählvariable gelöst werden. Die *for-each*-Schleife ist daher nur als alternative Lösungsmöglichkeit in geeigneten Kontexten zu sehen und es muss abgewogen werden, wie stark dieses Konstrukt im Unterricht thematisiert wird.

Von interessierten Schüler*innen, die mit der Erstellung von JFrames im JavaEditor vertraut sind, kann auch das beiliegende [Projekt Mind The Bug](#) umgesetzt werden. Das Projekt ist angelehnt an das

⁴ Niedersächsisches Kultusministerium (Hrsg.) (2025) *Ergänzende Hinweise zum Kerncurriculum Informatik für die gymnasiale Oberstufe am Gymnasium, an der Gesamtschule sowie für das Kolleg.*
<https://cuvo.nibis.de/index.php?p=download&upload=736> [Datum des Zugriffs: 13.08.2025]

Spiel *Minesweeper*⁵. Auch dieses Projekt fördert den Umgang mit den Indizes einer zweidimensionalen Reihung, da hier auch jeweils die Nachbarn eines Feldes betrachtet werden müssen. Da die Oberfläche aus sehr vielen Buttons besteht, ist es sinnvoll diese automatisiert zu erzeugen und in einer zweidimensionalen Reihung zu verwalten. Dazu kann den Schülerinnen und Schülern die beiliegende Vorlage zur Verfügung gestellt werden. Da sich die Aufgabe in mehrere Teilaufgaben zerlegen lässt, ist sie auch für eine arbeitsteilige Implementierung geeignet.

Ausblick

Mit diesem Leitfaden ist der Grundstein für den Umgang mit Reihungen gelegt. In Bezug auf das Suchen und Sortieren bietet es sich im Sinne der Förderung von Kreativität und eigenständigem Problemlösen an, die Schüler*innen zunächst selbst entsprechende Algorithmen entwickeln zu lassen und diese angemessen zu würdigen. Darauf aufbauend können z. B. unter dem Aspekt der Effizienz weitere bekannte Algorithmen betrachtet und verglichen werden. Mögliche Zugänge finden Sie in dem Materialpaket zum Thema *Suchen und Sortieren*. Eine Erläuterung verschiedener Sortierverfahren bietet z. B. Klaus Becker auf [inf-schule.de](https://www.inf-schule.de/grenzen/komplexitaet/sortieren): <https://www.inf-schule.de/grenzen/komplexitaet/sortieren> [Datum des Zugriffs: 23.02.21] Das Thema Effizienz kann in Kursen auf erhöhtem Niveau mit dem Materialpaket *Laufzeitkomplexität von Algorithmen* vertieft werden.

In Processing können Bilder mit wenigen Befehlen geladen werden. Die Farben der einzelnen Pixel stehen dann in einer eindimensionalen Reihung zur Verfügung. Hier ergibt sich ein weites Feld an interessanten Projekten. Entsprechende Unterrichtsideen sind im Materialpaket *Rekonstruktion ausgewählter Funktionen eines Bildbearbeitungsprogramms* zu finden.

Hinweise

Die Implementierungen im Lösungsordner stellen lediglich eine mögliche Lösung dar und erheben nicht den Anspruch besonders effizient oder dergleichen zu sein.

Zu beachten ist, dass sich die Materialien zwar am niedersächsischen Kerncurriculum für die gymnasiale Oberstufe orientieren, jedoch keinen Anspruch auf Vollständigkeit hinsichtlich der für die Abiturprüfung erwarteten Kompetenzen erheben. Die Autorin hat zum Teil individuelle Schwerpunkte gesetzt. Darstellungen und Schreibweisen können ggf. von den in der Abiturprüfung verwendeten Darstellungen und Schreibweisen abweichen. Verbindlich für das Abitur in Niedersachsen sind allein das niedersächsische Kerncurriculum für die gymnasiale Oberstufe sowie die ergänzenden Hinweise in der jeweils aktuellen Fassung. Es obliegt daher der jeweiligen Fachlehrkraft, den Unterricht so zu gestalten, dass die Schülerinnen und Schüler umfassend auf das Abitur vorbereitet werden. Die vorliegenden Materialien stellen somit nur eine Anregung dar, die an die individuellen Bedürfnisse der Lerngruppe angepasst werden müssen.

Für die korrekte Ausführbarkeit der beiliegenden Quelltexte wird keine Garantie übernommen. Auch für Folgeschäden, die sich aus der Anwendung der Quelltexte oder durch eventuelle fehlerhafte Angaben ergeben, wird keine Haftung oder juristische Verantwortung übernommen.

Dieses Werk ist lizenziert unter einer [Creative Commons Namensnennung - Nicht kommerziell - Keine Bearbeitungen 4.0 International Lizenz](#).

⁵ Emmett Nicholas: <https://minesweeperonline.com/> [Datum des Zugriffs: 13.01.2023]