

## Didaktische Hinweise

### Fehlererkennung und -korrektur

#### Zielgruppe

Das niedersächsische Kerncurriculum Informatik für die gymnasiale Oberstufe (s. [3]) sieht Codierungen, die eine Fehlererkennung bzw. Fehlerkorrektur ermöglichen, als verpflichtendes Thema nur für Kurse auf erhöhtem Niveau vor. Die Materialien können aber auch für interessierte Schülerinnen und Schüler in Kursen auf grundlegendem Niveau oder in der Einführungsphase verwendet werden. Insbesondere das erste Arbeitsblatt zu fehlererkennenden Codes im Alltag eignet sich als Kontext für Programmieraufgaben in der Einführungsphase, da hierfür nur elementare Zeichenkettenoperationen benötigt werden.

#### Voraussetzungen

Die Aufgaben zur Erarbeitung der verschiedenen Codierungen zur Fehlererkennung und -korrektur können ohne besondere Vorkenntnisse bearbeitet werden. Lediglich einige Aufgaben des dritten Arbeitsblattes zur Reflexion beziehen das Binärsystem und Kompressionsverfahren mit ein. Diese Aufgaben können aber ggf. auch übersprungen werden bzw. kann das Binärsystem an dieser Stelle ggf. thematisiert werden.

Für die Implementierung der Verfahren sind elementare Zeichenkettenoperationen und der Umgang mit Variablen sowie die Umwandlung von Datentypen notwendig.

#### Aufbau der Materialien und Lernziele

Die Materialien sollen den Lernenden eine Auseinandersetzung mit dem Thema Fehlererkennung und -korrektur nach dem Prinzip des entdeckenden Lernens ermöglichen. Die Arbeitsblätter enthalten dabei verschiedene Aufgaben, die nicht nur die Anwendung der Codierungen, sondern auch die dahinterstehenden Konzepte und Zusammenhänge in den Blick nehmen. Die Lernenden werden dazu angeregt eigene Vermutungen aufzustellen und Ideen zu entwickeln.

Beginnend bei Prüfziffern, die im Alltag das Erkennen von Tippfehlern ermöglichen, erarbeiten sich die Lernenden grundlegende Prinzipien fehlererkennender Codes. Zentral ist hier das Ergänzen von zusätzlichen Daten, die so von den ursprünglichen Daten abhängen, dass eine Veränderung in den Daten auffällt. Dieses Prinzip wird in Form von Paritätsbits zur Fehlererkennung auf die Bitebene übertragen. Eine spezielle Anordnung und Kombination der Paritätsbits ermöglichen in Form des Hamming-Codes schließlich auch eine Fehlerkorrektur.

Mit den Materialien kann insbesondere die folgende Kompetenz aus dem Modul „Codierung und Übertragung von Daten“ im Lernfeld „Information und Daten“ gefördert werden (s. [3]):

*Die Schülerinnen und Schüler erläutern Möglichkeiten der Fehlererkennung und der Fehlerkorrektur bei der Datenübertragung, u. a. Paritätsbit, (7, 4)-Hamming-Code.*

Jedes Arbeitsblatt schließt mit der Implementierung der betrachteten Codierungen ab. Da die Codes für die Fehlererkennung und -korrektur bei der maschinellen Verarbeitung gedacht sind, liegt eine praktische Umsetzung in Form einer Implementierung nahe. Dabei muss nicht jeder Code von jedem Schüler bzw. jeder Schülerin implementiert werden. Die Aufgaben können auch zum Abschluss der Einheit arbeitsteilig bearbeitet und ggf. zu einem Gesamtprodukt zusammengefügt werden.

In der Einführungsphase kann die Anwendung und Implementierung der Codes in den Vordergrund gestellt werden, da sie einen geeigneten Kontext für den Umgang mit Zeichenkettenoperationen bieten. Gefördert werden können insbesondere folgende Kompetenzen aus dem Modul „Grundlagen der Algorithmik“ aus dem Lernfeld „Algorithmen und Datenstrukturen“ (s. [3]):

*Die Schülerinnen und Schüler ...*

- *verwenden geeignete Variablentypen zur Speicherung von Werten*
- *entwerfen und implementieren Algorithmen unter Verwendung elementarer Zeichenkettenoperationen.*

Exemplarische Implementierungen stehen im Lösungsordner sowohl in Processing<sup>1</sup> als auch als JFrame<sup>2</sup>, der mit dem Java-Editor erstellt wurde, zur Verfügung.

## Hinweise zu einzelnen Arbeitsblättern

### AB 01 Fehlererkennende Codes im Alltag

Es gibt eine Vielzahl von Algorithmen zur Berechnung von Prüfziffern, die häufig die Berechnung einer (gewichteten) Quersumme und eine Modulo-Rechnung enthalten. Hier werden exemplarisch die Verfahren für die Seriennummern von Eurobanknoten, für ISBN bzw. EAN und der Luhn-Algorithmus betrachtet. Verfügen ältere Schülerinnen und Schüler bereits über eine Kreditkarte kann der Luhn-Algorithmus alternativ zur abgebildeten Loknummer darauf angewendet werden.

Für eine individuelle Implementierung können die Lernenden auch eigene Verfahren entwickeln oder weitere recherchieren. Die Bundesbank stellt z. B. eine Übersicht über die Methoden zur Prüfziffernberechnung bei Kontonummern zur Verfügung (s. [2]).

Eine Herausforderung bei der Implementierung ist die Umwandlung einer Eingabe als Zeichenkette in einzelne Zahlen. Das Arbeitsblatt enthält daher entsprechende Hilfestellungen, die ggf. mit den Lernenden besprochen werden müssen. Für die Berechnung der Prüfziffer der ISBN oder nach dem Luhn-Algorithmus benötigen die Schülerinnen und Schüler ggf. den Hinweis, dass sich die Differenz zum nächstgrößeren Vielfachen von 10 berechnen lässt, indem erst modulo 10 gerechnet und dann die Differenz zu 10 bestimmt wird.

### AB 02 Fehlererkennung auf Bitebene

Fehlererkennung und – korrektur am Rechner ist den Lernenden bislang vermutlich vor allem im Zusammenhang mit dem Testen von Implementierungen geläufig. Es ist daher zunächst zu klären, dass es sich hier um technische Fehler, die beim Speichern und Übertragen von Daten auftreten können, handelt und nicht um Fehler, die auf fehlerhafte Berechnungen zurückzuführen sind. Das Beispiel in Aufgabe 1 kann zur Veranschaulichung dienen.

Basierend auf den Ideen der Schülerinnen und Schüler zu Aufgabe 1 lässt sich vermutlich bereits herausarbeiten, dass Fehlererkennung bzw. -korrektur nur durch das Hinzufügen zusätzlicher Bits möglich ist. Die folgenden Aufgaben und AB 03 können dann dazu verwendet werden, die Ideen der Schülerinnen und Schüler zu vertiefen oder zu ergänzen und schließlich eine möglichst effiziente Ergänzung von Prüfbits zu erarbeiten.

<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> Der Java-Editor wird von Gerhard Röhner zur Verfügung gestellt: <https://javaeditor.org/>

Grundsätzlich kann bei Paritätsbits mit gerader oder ungerader Parität gearbeitet werden. Hier wird vorrangig die gerade Parität verwendet, da diese auch im Zusammenhang mit dem Hamming-Code gebräuchlich ist.

### AB 03 Fehlerkorrektur

Zum besseren Verständnis der Funktionsweise des Hamming-Codes wird zunächst der Hamming-Abstand eines Codes eingeführt.

Für Prüfungskurse ist zu beachten, dass eine Tabelle wie in Abbildung 3 im AB 03 in Niedersachsen auch als Hilfsmittel in der Abiturprüfung zur Verfügung steht (vgl. [4]).

Für eine kompakte Darstellung der Berechnung der Prüfbits wird die logische Operation *xor* eingeführt. Alternativ kann darauf verzichtet und über eine gerade bzw. ungerade Anzahl an Einsen argumentiert werden. Insbesondere bei der Implementierung fällt den Lernenden dieser Zugang ggf. leichter.

Im niedersächsischen Kerncurriculum Informatik wird nur der (7, 4)-Hamming-Code in den erwarteten Kompetenzen für Kurse auf erhöhtem Niveau explizit genannt. In AB 03 wird in einem Exkurs zusätzlich der allgemeine Aufbau des Hamming-Codes auch für längere Bitfolgen betrachtet. Diese Ergänzung kann aus zwei Gründen sinnvoll sein: Erstens wird das System zum Erzeugen des Hamming-Codes bei der allgemeinen Variante sichtbar, so dass die Regeln zur Bildung der Prüfbits nicht für jedes Prüfbit auswendiggelernt werden müssen, sondern aus der Systematik hergeleitet werden können. Das fördert das Verständnis der zugrundeliegenden Mechanismen zur Fehlererkennung. Zweitens erscheint das Hinzufügen der Prüfbits nach dem Hamming-Code-Verfahren erst für längere Bitfolgen effizient, da sich das Verhältnis von Datenbits zu Prüfbits bei längeren Bitfolgen zunehmend verbessert.

Eine Implementierung des Erzeugens, Prüfens und Korrigierens kann mithilfe elementarer Zeichenkettenoperationen erfolgen. Aufbauend auf einer Operation zum Erzeugen eines (7, 4)-Hamming-Codes können systematisch alle 16 Hamming-Codes und ihre paarweisen Hamming-Abstände ausgegeben werden. Dafür bietet sich die Verwaltung der Hamming-Codes sowie ihrer Hamming-Abstände in einer 2D-Reihung an. Diese Aufgabe eignet sich daher für Lernende mit entsprechend fortgeschrittenen Programmierkenntnissen. Anhand der Auflistung aller Hamming-Codes und ihrer paarweisen Hamming-Abstände kann gezeigt werden, dass der (7, 4)-Hamming-Code tatsächlich einen Hamming-Abstand von 3 hat.

QR-Codes sind vermutlich die fehlerkorrigierenden Codes, denen Schülerinnen und Schüler im Alltag am häufigsten begegnen. Da das bei QR-Codes verwendete Reed-Solomon-Verfahren zur Fehlerkorrektur jedoch für die Schule zu komplex ist, wurde hier auf einen Einstieg mit QR-Codes verzichtet. Stattdessen stehen diese als Beispiel aus der Lebenswelt am Ende der Einheit. Mit ihrem bis dahin erworbenen Wissen über fehlerkorrigierende Codes können die Lernenden QR-Codes als fehlerkorrigierenden Code einordnen und zumindest allgemein erklären, warum beschädigte QR-Codes bis zu einem gewissen Grad der Beschädigung noch lesbar sind. Bei Interesse können für eine Vertiefung des Aufbaus von QR-Codes z. B. die Materialien unter [5] von Marion Stadtmüller genutzt werden. Gleichzeitig bieten die QR-Codes einen Ausblick, dass die Codes zur Fehlererkennung und Fehlerkorrektur auf Basis der Forschung von Hamming weiterentwickelt wurden (s. [1]).

## Erweiterungen

Insbesondere in Kursen auf erhöhtem Niveau kann die Implementierung der fehlererkennenden und fehlerkorrigierenden Codes mit anderen Codierungen zur Kompression oder Verschlüsselung zu einem größeren Projekt im Bereich Codierung kombiniert werden. In diesem Zusammenhang kann die Reihenfolge von Codierungen beispielsweise zur Kompression und Fehlererkennung auch praktisch erörtert werden. Es steht z. B. auch ein Materialpaket zur Implementierung einer Huffman-Codierung zur Verfügung.

Eine praktische Anwendung könnte die Fehlererkennung auch bei der Übertragung von Nachrichten mithilfe von Lichtsignalen durch eine Pappröhre finden (vgl. dazu das Materialpaket „Übertragung von Daten über einen Kommunikationskanal“ im Lernfeld „Information und Daten“.)

## Hinweis

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, die auch über die im KC geforderten Kompetenzen hinausgehen können. 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 (s. [2]) sowie die ergänzenden Hinweise (s. [4]) in der jeweils aktuellen Fassung. Es obliegt daher den jeweiligen Fachlehrkräften, 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.

## Literatur

- [1] Bischoff, Manon (09.05.2024). Die fabelhafte Welt der Mathematik: Wie Maschinen den Fehlerteufel bekämpfen. <https://www.spektrum.de/kolumne/wie-der-hamming-code-zur-fehlerkorrektur-funktioniert/2214247> [Datum des Zugriffs: 04.06.2025]
- [2] Deutsche Bundesbank. (Stand: 2018) Prüfzifferberechnungsmethoden zur Prüfung von Kontonummern auf ihre Richtigkeit. <https://www.bundesbank.de/de/aufgaben/unbarer-zahlungsverkehr/serviceangebot/pruefzifferberechnung/pruefzifferberechnung-fuer-kontonummern-603282>
- [3] Niedersächsisches Kultusministerium (Hrsg.) (2017) Kerncurriculum für das Gymnasium - gymnasiale Oberstufe, die Gesamtschule – gymnasiale Oberstufe, das Kolleg. Informatik. Hannover: unidruck
- [4] Niedersächsisches Kultusministerium (Hrsg.) (2021) Ergänzenden Hinweisen zum Kerncurriculum Informatik für die gymnasiale Oberstufe am Gymnasium und an der Gesamtschule sowie für das Kolleg. <https://cuvo.nibis.de/index.php?p=download&upload=260> [Datum des Zugriffs: 09.04.2025]
- [5] Stadtmüller, Marion (2024). Struktur von QR-Codes. [https://inf-schule.de/information/barcode\\_qrcode/qrcode/qrcode\\_struktur](https://inf-schule.de/information/barcode_qrcode/qrcode/qrcode_struktur) [Datum des Zugriffs: 03.06.2025]

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.

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.