

Kontextbezogene Aufgaben

Medizinische Informatik, Strichcode-Scanner, Zeichenprogramm

von Kerstin Strecker

Warum kontextbezogene Aufgaben?

Durch kontextbezogene Aufgaben kann den Schülerinnen und Schülern die Sinnhaftigkeit der Informatik eindrucksvoll vermittelt werden, wenn die Informatik, deren Denkweisen, Strategien und Werkzeuge zur Lösung von Problemen genutzt werden, die in anderen (Lebens-)Bereichen, sozusagen „im Kontext“ auftreten (vgl. z.B. Diethelm u. a., 2011, Modrow, 2010, oder Koubek u.a., 2009). „Informatik für einen bestimmten Zweck“ und nicht „Informatik an sich“ ist die Botschaft, die durch kontextbezogene Aufgaben vermittelt werden soll. Die Integration anderer Fächer oder Lebensbereiche sollte so umfassend wie möglich sein. Nebenbei kann ein Kontext aus der Lebenswelt der Schülerinnen und Schüler ein zusätzlicher Motivator sein.

Die Überlegungen lassen sich an einem Aufgabenbeispiel aus der Oberstufe konkretisieren: Die Lehrperson möchte von den Schülerinnen und Schülern das Durchlaufen einer zweidimensionalen Reihung (array, Feld) auf der Suche nach einer bestimmten Information fordern. Die fachliche Kompetenz ist also, eine zweidimensionale Reihung zu durchlaufen und die einzelnen Feldelemente auf ein Merkmal hin zu untersuchen. Eine mögliche Aufgabe ist, die zweidimensionale Reihung mit Zufallszahlen zu füllen und anschließend zu prüfen, wie oft die Zahl 3 in der Matrix vorkommt. Eine solche Aufgabe wäre rein innerinformatisch, also „Informatik an sich“.

Ein Anwendungsbezug kommt hinzu, wenn die Zahlen 0 bis 255, die in der zweidimensionalen Reihung vorkommen sollen, als Farbwerte einzelner Pixel und die gesamte Matrix als Bild interpretiert werden. Dann könnte die Aufgabe lauten: Erzeugt ein Negativbild des gegebenen Bildes, indem die Farbwerte der einzelnen Pixel umgekehrt werden (neuer Farbwert = 255 – aktueller Farbwert):

140	140	140	140	140	115	115	115	115	115
140	80	255	80	140	115	175	0	175	115
140	255	255	255	140	115	0	0	0	115
255	80	255	80	255	0	175	0	175	0
255	255	80	255	255	0	0	175	0	0

Eine alternative Aufgabe lautet: Erzeugt eine Tontrennung des gegebenen Bildes, indem alle Pixelwerte ≥ 125 auf den neuen Wert 255 und alle Pixelwerte < 125 auf den neuen Wert 0 gesetzt werden:

140	140	140	140	140	255	255	255	255	255
140	80	255	80	140	255	0	255	0	255
140	255	255	255	140	255	255	255	255	255
255	80	255	80	255	255	0	255	0	255
255	255	80	255	255	255	255	0	255	255

Dann wird die fachliche Anforderung in einen Anwendungsbezug gekleidet, aber dieser Kontext selbst wird nicht Grundlage einer Diskussion oder weiteren Vertiefung sein. Es steht immer noch die Informatik und nicht der Zweck im Vordergrund.

Eine gute kontextbezogene Aufgabe, die dieselbe informatische Kompetenz fordert, ist in meinen Augen eher die im Kasten „Aufgabe: ‚Fieberscan‘“ wiedergegebene (siehe nächste Seite).

Der Kontextbezug in der Aufgabe – die Suche nach kranken Personen in Zeiten der SARS-Pandemie – hat zwei entscheidende Vorteile:

Aufgabe: „Fieberscan“

„Als in Asien im Jahr 2003 der Erreger der Lungenkrankheit SARS wütete, waren auf Flughäfen in Taiwan und Singapur Wärmebildkameras zu sehen. Mit dem sogenannten Thermal Screening sollten fiebrige Einreisewillige frühzeitig erkannt, aussortiert und eingehend untersucht werden.“

Quelle:
<http://www.spiegel.de/wissenschaft/mensch/pandemie-praevention-experten-gegen-waermescanner-auf-flughaefen-a-621716.html>
 (zuletzt geprüft: 16.12.2013)



Foto: fotolia / Ingo Bartussek

Das Wärmebild einer Person (siehe Bild links) wird in einer zweidimensionalen Reihung gespeichert.

Die Inhalte der einzelnen Zellen sind Zahlen 0 bis 255, die die Farbe eines Pixels repräsentieren, was wiederum einem Wärmewert entspricht (Abbildung unten). Um Personen auf das Vorhandensein von Fieber zu scannen, sollen Mediziner die Schwelle für „fiebrige“ Pixel (den Wärmewert) und die Anzahl derer eingeben, ab der eine Person einer ärztlichen Kontrolle unterzogen werden sollte. Schreibt eine Operation *scan*, die ein Wärmebild auf Fieber hin untersucht. Die Eingabeparameter sind die Anzahl fiebriger Pixel, der Schwellenwert für Fieber und eine zweidimensionale Reihung (das Wärmebild). Der Rückgabewert ist ein Wahrheitswert.

140	140	140	140	140
140	80	255	80	140
140	255	255	255	140
255	80	255	80	255
255	255	80	255	255

- ▷ Die Schülerinnen und Schüler lösen keine informatischen Probleme um ihrer selbst Willen, sondern sie betreiben Informatik, um Probleme zu lösen, die in anderen Bereichen entstanden sind. Im hier vorliegenden Fall soll die Ausbreitung einer Pandemie gestoppt werden. Der Zweck steht also im Vordergrund, und der Zweck kann auch hinterfragt werden oder weitere Diskussionen nach sich ziehen. Wenn in einer zweidimensionalen Reihung 25-mal die Zahl 3 gefunden wurde, dann gibt es wenig Anlass zur Diskussion. Wenn allerdings auf einem Wärmebild 25 „fiebrige“ Pixel gefunden wurden, dann kann der Reisende ja auch eine heiße Kaffeetasse in der Hand gehalten haben. Oder die Pixel sind gar nicht großflächig zusammenhängend. Oder es liegt ein Messfehler vor. Der Kontext selbst wirft also neue informatische Fragestellungen auf: Wie erkennt man z. B., ob die Pixel zusammenhängend sind?

Steckbrief zur Aufgabe „Fieberscan“

- (1) *Jahrgangsstufe/Schulart/Autor (Bundesland)*
Jahrgangsstufe: Oberstufe.
Schulart: Gymnasium.
Autor (Bundesland): Kerstin Strecker (Niedersachsen).
- (2) *Fachliche Thematik*
 Zweidimensionale Arrays im Anwendungskontext.
- (3) *Vorwissen*
 Umgang mit zweidimensionalen Arrays.
- (4) *Einsatzbereich der Aufgabe*
 Klausur- bzw. Übungsaufgabe.
- (5) *Zeitlicher Umfang*
 20 bis 30 Minuten.
- (6) *Aufgabenstellung, (7) Lösungsskizze*
 Siehe nebenstehenden Text.
- (8) *Einordnung in die Bildungsstandards*
 Schwerpunkte sind der Prozessbereich *Implementieren* sowie der Inhaltsbereich *Algorithmen*.

- ▷ Die Lösungen dieser Fragestellungen können wiederum implementiert werden, oder zumindest können die Grenzen der zugrunde liegenden Modellierung und des Messens reflektiert werden. Weiterhin bedarf es natürlich auch einer Diskussion, ob es vertretbar ist, Einreisende auszusortieren und ihnen eine ärztliche Untersuchung aufzuzwingen.

Unbewusst wird weiterhin noch eine andere Botschaft vermittelt: Ein komplex wirkendes Informatiksystem, der Wärmebildscanner, wird entmystifiziert. Den Schülerinnen und Schülern wird durch die Aufgabenstellung zugehört, mit ihren Fähigkeiten die Funktionalität eines realen Systems zu rekonstruieren. Dies zeigt auch eine gewisse Wertschätzung (zur Dekonstruktion von Informatiksystemen vgl. beispielsweise auch Magenheimer, 2000).

Kontextbezogene Aufgaben ermöglichen Differenzierung

Informatik-Lerngruppen sind in der Regel sehr heterogen, z. B. bedingt durch außerschulische Erfahrungen in diesem Lernbereich. Hier hat die Informatik Ähnlichkeit mit den Fächern Musik und Sport, in denen die Schülerinnen und Schüler durch das Beherrschen eines Instruments oder durch außerschulisches Training in einem Sportverein ganz unterschiedliche und zum Teil erhebliche Vorerfahrungen mitbringen. Dazu kommt, dass gerade im Bereich der Algorithmik analytisches Denken gefragt ist, das den Schülerinnen und Schülern unterschiedlich schwer fällt.

Die dadurch notwendige Differenzierung sollte neben verschiedenen Lernniveaus im Idealfall unterschiedliche vorrangig geförderte prozess- oder inhaltsbezogene Kompetenzen unterstützen.

Strichcode

Die *European Article Number* (kurz: EAN) stellt eine international unverwechselbare Produktkennzeichnung für Handelsartikel dar (vgl. Dresch, 1986; Spolwig, 1990, S.50f.). Die Nummer besteht aus 8 bzw. 13 Ziffern, wobei die Codes EAN-8 und EAN-13 die gleiche Codierung besitzen. Die codierte Information wird durch eine numerische Klarschriftangabe unter dem Barcode wiederholt.



Ein EAN-8-Barcode.

Der hier abgebildete EAN-8-Code besteht aus einer Folge von schwarzen und weißen Strichen, wobei jeder der Ziffern 0 bis 9 eine Folge von vier Strichbreiten abwechselnd in schwarz und weiß zugeordnet ist (siehe Tabelle). Die längeren, grau unterlegten Striche begrenzen den Code und dienen der Orientierung (drei Striche am Anfang und Ende, fünf in der Mitte).

Ziffer	Strichbreiten	Ziffer	Strichbreiten
0	3211	5	1231
1	2221	6	1114
2	2122	7	1312
3	1411	8	1213
4	1132	9	3112

Wird ein Strich der Breite n als n Striche der Breite 1 dargestellt, besteht jede Ziffer aus sieben schwarzen oder weißen Strichen, was bei acht Ziffern 56 Striche ausmacht. Die gegenüber der EAN-13 verkürzte Version EAN-8 ist speziell für kleine Artikel gedacht, auf denen eine EAN-13 mehr als 25 % des Platzes auf der Vorderseite benötigen würde. Der Aufbau der EAN-8 setzt sich wie folgt zusammen:

- ▷ GS1-Länderpräfix (2–3 Stellen),
- ▷ Artikelnummer (Reststellen, also 4–5),
- ▷ Prüfziffer (1 Stelle).

Seit 2009 sind die Benennungen für EAN-Codes durch die Bezeichnungen für die Globale Artikelidentnummer (*Global Trade Item Number* – GTIN) abgelöst worden, sodass der EAN-8 nunmehr mit *GTIN-8* bezeichnet wird.

Literatur und Internetquellen zum Thema „EAN“

Dresch, P.-J.: Unterrichtseinheit „Strichcodes und Computerkassen“. In: LOG IN, 6. Jg. (1986), Heft 3, S.18–24.

Spolwig, S.: Warenwirtschaft im Unterricht (Teil 1) – Zu einem handlungsorientierten und EDV-gestützten Unterricht über „Warenwirtschaftssysteme“ im Ausbildungsberuf „Kaufmann/Kaufrau im Einzelhandel“. In: LOG IN, 10. Jg. (1990), Heft 3, S.46–51.

Wikipedia – Stichwort „European Article Number“:
http://de.wikipedia.org/wiki/European_Article_Number

Wikipedia – Stichwort „Global Trade Item Number“:
http://de.wikipedia.org/wiki/Global_Trade_Item_Number

Um wirklich verschiedene Lernniveaus mit einer Aufgabe zu bedienen und verschiedene Herangehensweisen zu ermöglichen, ist oft ein *Aufgabenvorrat* erforderlich, in dem sich die Aufgabenstellungen je nach vorrangig geförderter Kompetenz oder gefordertem Lernniveau leicht voneinander unterscheiden. Ein gut gewählter *Kontext* kann dabei ermöglichen, dass diese „unterschiedlichen“ Aufgaben trotzdem zusammenhängend sind.

Im folgenden Beispiel-Aufgabenvorrat geht es um den Kontext „Strichcode“, bzw. die Rekonstruktion des Informatiksystems „Strichcode-Scanner“ (siehe auch Informationskasten „Strichcode“). Als Werkzeug wird SCRATCH verwendet, bzw. in einer Aufgabe werden auch NXT-Roboter eingesetzt. Die einzelnen Aufgaben fordern von den Schülerinnen und Schülern, jeweils einen Teilbereich eines Barcode-Scanners oder -Generators zu rekonstruieren, doch jede Aufgabe hat einen unterschiedlichen Anspruch. Durch das Zusammenfügen der einzelnen Lösungen ergibt sich später am Ende der Bearbeitungszeit die Rekonstruktion des betrachteten Informatiksystems.

Die Stunde könnte so beginnen, dass mit einem echten Strichcode-Scanner mitgebrachte Produkte eingescannt werden und ein Programm den zugehörigen Produktnamen und Preis – für alle Schülerinnen und Schüler sichtbar – ausgibt. Aufgaben für die Teilgruppen könnten dann wie im Kasten „Aufgabe Gruppe 1: „Recherche Strichcode““ aussehen.

Bei dieser Teilaufgabe 1 geht es um Inhalte aus dem Inhaltsbereich *Information und Daten*, aber auch *Sprachen und Automaten*. Die vorrangig geförderte prozessbezogene Kompetenz ist *Kommunizieren*. Das Ergebnis bildet später den Rahmen, um die Ergebnisse der anderen Gruppen sinnvoll zu einem Ganzen zusammenzufügen. Es ist die einzige Teilaufgabe, in der kein Algorithmus gefunden werden muss; der Anspruch der Aufgabe ist eher gering.

Dagegen wird bei der Teilaufgabe 2 (siehe Kasten „Aufgabe Gruppe 2: „Strichcode scannen““, nächste Seite) das selbstständige Entwickeln von Algorithmen gefordert.

Es ergeben sich bei der Teilaufgabe 2 aus diesem reduzierten Barcode (die Striche zur Orientierung fehlen) 56 Einträge in die Liste. Durch diese Menge werden leicht Fehler verursacht. Natürlich kann auch ein kürzerer Code verwendet oder nur eine Zahl codiert

Aufgabe Gruppe 1: „Recherche Strichcode“

Ziel: Aufbau des EAN-8-Strichcodes verstehen und diesen in eine Ziffernfolge übersetzen können.

Recherchiert im Netz Informationen über den EAN-8-Code. Versucht mit eurem gewonnenen Wissen die Strichcodes der Produkte am Lehrertisch selbst in die darunter stehende Ziffernfolge zu übersetzen. Bei Fragen stehen euch Hilfefkarten zur Verfügung.

Fertigt eine Präsentation an, mit der ihr euer Wissen, wie die schwarz-weißen Strichfolgen in Zahlen übersetzt werden, den anderen erläutern könnt.

Aufgabe Gruppe 2: „Strichcode scannen“

Ziel: Lesen eines Strichcodes mit Lichtsensoren und Speichern des Ergebnisses in einer Tabelle.



Der EAN-8-Strichcode besteht abwechselnd aus weißen und schwarzen Strichen. Jede Ziffer 0 bis 9 besteht aus einer Folge von vier Strichen, die aber eine unterschiedliche Strichbreite haben können. Die Ziffer 4 beispielsweise besteht aus einem weißen Strich (der Breite 1), einem schwarzen Strich (der Breite 1), einem weißen Strich der Breite 3 und einem schwarzen Strich der Breite 2.

Verwendet den NXT-Roboter mit zwei Lichtsensoren und einen Berührungssensor. Den vergrößerten EAN-Code findet ihr auf dem Lehrertisch. Ein Lichtsensor kann einen Strich der Breite 1 einlesen. Haltet die beiden Lichtsensoren auf die ersten beiden Striche und lest die Daten der Sensoren auf Knopfdruck ein. Haltet danach die Lichtsensoren auf die nächsten beiden Striche usw.

Wandelt die eingelesenen Daten in die Information „s“ für Schwarz oder „w“ für Weiß um und tragt sie in eine Liste ein. Da der Strichcode-Scanner Striche der Breite 1 einliest, entspricht ein Strich der Breite 3 also 3 Einträgen in der Liste.



Beispiel:

Für die Ziffer 4 würde die Liste wie im rechten Bild aussehen:

Lösungsskizze:

- ▷ Bei Tastendruck Werte der Lichtsensoren abfragen.
- ▷ Mit einem Schwellwert vergleichen.
- ▷ Abhängig vom Vergleich entweder „s“ oder „w“ der Liste hinzufügen.

Steckbrief zur Aufgabengruppe „Strichcode“

(1) Jahrgangsstufe/Schulart/Autor (Bundesland)

Jahrgangsstufe: 8–10.

Schulart: Gymnasium.

Autor (Bundesland): Kerstin Strecker (Niedersachsen).

(2) Fachliche Thematik

Rekonstruktion eines Strichcode-Scanners.

(3) Vorwissen

Umgang mit SCRATCH und NXT-Robotern, eigenständiges Arbeiten.

(4) Einsatzbereich der Aufgabe

Siehe Text, Seite 90ff.

(5) Zeitlicher Umfang

Insgesamt mit abschließender Zusammenführung der einzelnen Teile zwei Doppelstunden.

(6) Aufgabenstellung,

(7) Lösungsskizze,

(8) Einordnung in die Bildungsstandards

Siehe jeweils den entsprechenden Text, Seite 90ff.

Aufgabe Gruppe 3: „Gescannten Code zusammenfassen“

Ziel: Der Strichcode-Scanner liefert eine Folge von Werten w für weiße Striche und s für schwarze Striche. Folgen gleiche Farben aufeinander, heißt das, dass der Strich hier breiter war. Diese Strichbreiten müssen rekonstruiert werden.



Kontext:

Der EAN-8-Strichcode besteht abwechselnd aus weißen und schwarzen Strichen. Jede Ziffer 0 bis 9 besteht aus einer Folge von vier Strichen, die aber eine unterschiedliche Strichbreite haben können. Die Ziffer 4 beispielsweise besteht aus einem weißen Strich, einem schwarzen Strich, einem weißen Strich der Breite 3 und einem schwarzen Strich der Breite 2, in Kurzschreibweise:

Ziffer 4 entspricht: 1-1-3-2

Der Strichcode-Scanner liest aber fortlaufend Striche der Breite 1 ein und wandelt die Daten in die Buchstaben s (Schwarz) oder w (Weiß) um, die in einer Liste gespeichert werden. Ein Strich der Breite 3 entspricht also 3 Einträgen in die Liste.

Beispiel:

Für die Ziffer 4 würde die Liste wie nebenstehend abgebildet aussehen:



Aus dieser Liste w-s-w-w-w-s-s muss ein Programm nun die Liste 1-1-3-2 machen.

Aufgabenstellung:

Implementiert ein Programm, das eine Liste mit Einträgen w und s so zusammenfasst, dass in einer neuen Liste nur die Anzahlen stehen.

Beispiel 1:



Beispiel 2:



Lösungsskizze:

Ähnlich einer Lauflängencodierung muss beim Durchlaufen der Liste eine Variable hochgezählt werden, bis sich der Listeneintrag von „w“ nach „s“ oder „s“ nach „w“ ändert. In diesem Fall muss dann der Wert der Variablen in der Zusammenfassung gespeichert und im Anschluss wieder auf Null zurückgesetzt werden. Wird das Ende des Feldes erreicht, muss ebenfalls der Wert der Variablen gespeichert werden.

werden, es geht ja schließlich nur um das Prinzip. Statt Roboter kann auch das *PicoBoard* unter SCRATCH verwendet werden. Die Roboter können zur Lösung der Aufgabenstellung mit der SCRATCH-Modifikation ENCHANTING programmiert werden.

Hinsichtlich der Bildungsstandards (vgl. AKBSI, 2008) kann diese 2. Teilaufgabe den Inhaltsbereichen *Algorithmen* und *Sprachen und Automaten* zugeordnet werden. Die vorrangig geförderte prozessbezogene Kompetenz ist *Modellieren und Implementieren*. Der Anspruch ist eher im mittleren Bereich zu sehen.

Bei der Teilaufgabe 3 im Kasten „Aufgabe Gruppe 3: ‚Gescannten Code zusammenfassen‘“ werden die Inhaltsbereiche *Algorithmen* sowie *Sprachen und Automaten* angesprochen. Die vorrangig geförderte prozessbezogene Kompetenz ist *Modellieren und Implementieren*. Der Anspruch dieser Teilaufgabe ist sehr hoch.

Aufgabe Gruppe 4: „Gescannten Code zusammenfassen – Teil 2“

Ziel: Umwandlung der Ziffernfolge des Strichcodes in die Ziffernfolge der European Article Number (EAN).

Die vom Strichcode-Scanner aufgenommenen Daten wurden inzwischen so weit verarbeitet, dass sie nun in die zugehörigen Ziffern übersetzt werden können. In der Datei xxx findet ihr eine Liste, in der sich für jede Ziffer jeweils vier Einträge befinden, die den Strichbreiten des zugehörigen EAN-Codes entsprechen.

Ziffer	Listeneinträge	Ziffer	Listeneinträge
0	3211	5	1231
1	2221	6	1114
2	2122	7	1312
3	1411	8	1213
4	1132	9	3112



Fasst jeweils vier Einträge entsprechend der Tabelle zu einer EAN-Code-Ziffer zusammen und gebt diese aus. Die Abbildung links zeigt Ein- und Ausgabe exemplarisch an zwei Ziffern.

Lösungsskizze:

Die vier ersten Listeneinträge verketteten und mithilfe von zehn IF-Anweisungen vergleichen, bis die passende Ziffer gefunden ist. Die Ziffer wird an das Ergebnis angehängt, und die vier ersten Listeneinträge werden gelöscht. Dieser Vorgang wird wiederholt, bis das Listeneinde erreicht ist.

Der Anspruch der Teilaufgabe 4 (siehe Kasten „Aufgabe Gruppe 4: ‚Gescannten Code zusammenfassen – Teil 2‘“) ist eher im mittleren Bereich zu sehen. Mit Bezug zu den Bildungsstandards kann die Aufgabe den Inhaltsbereichen *Algorithmen* sowie *Sprachen und Automaten* zugeordnet werden. Die vorrangig geförderte prozessbezogene Kompetenz ist hier ebenfalls *Modellieren und Implementieren*.

Aufgabe Gruppe 5: „Gescannten Code übersetzen“

Ziel: Ein Strichcode-Scanner scannt eine Ziffernfolge von 8 Ziffern ein. Diese muss in den Produktnamen und Preis umgewandelt werden.

Schließt den Strichcode-Scanner an den Rechner an. Wenn ihr in SCRATCH eine Frage stellt, dann könnt ihr die Strichcodes auf den „realen“ Verpackungen auf dem Lehrertisch mit dem Strichcode-Scanner einscannen, und die zugehörige Zahl wird als *answer* in SCRATCH gespeichert. Baut den Teil des Strichcode-Scanners nach, der diese Zahl in einen Produktnamen und Preis umwandelt.

answer 40084039



Lösungsskizze:

Für eine beliebige Anzahl an Produkten jeweils eine bedingte Anweisung erzeugen, die bei Übereinstimmung mit dem eingescannten EAN-Code den Produktnamen und Preis ausgibt.

Bei der Teilaufgabe 5 (siehe Kasten „Aufgabe Gruppe 5: ‚Gescannten Code übersetzen‘“) muss zumindest noch Folgendes beachtet werden: Wenn kein Strichcode-Scanner vorhanden ist, kann der EAN-Code alternativ auch per Hand eingegeben werden.

Der Inhaltbereich dieser Aufgabe ist *Informatiksysteme*. Die vorrangig geförderte prozessbezogene Kompetenz ist *Modellieren und Implementieren*. Der Anspruch der Aufgabe ist gering.

Jede der bislang vorgestellten fünf Teilaufgaben kann für sich allein bearbeitet werden und stellt unterschiedliche Ansprüche an die Schülerinnen und Schüler. Nur durch das Vernetzen der einzelnen Lösungen nach einer Präsentation durch die jeweiligen Gruppen ist das Informatiksystem *Strichcode-Scanner* mithilfe der Algorithmik rekonstruiert. Durch das verteilte Arbeiten wird die Kompetenz *Kooperieren* gefördert, da jede Schülerin und jeder Schüler einen Anteil am Gesamtergebnis hat. Eine gute Präsentation, die unbedingt erforderlich ist, um den Mitschülerinnen und Mitschülern die noch fehlenden Puzzleteile zu ihrer Aufgabenlösung zu liefern, fördert zudem noch die Kompetenz *Kommunizieren*. Der Kontext liefert den Zusammenhang der einzelnen Aufgaben.

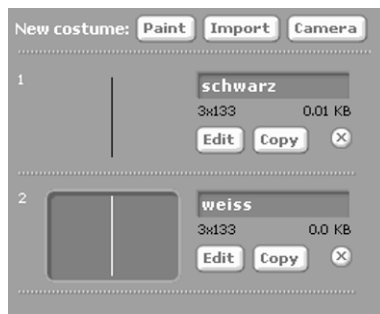
Ergänzend zu den Teilaufgaben 1 bis 5 kann eine sehr leistungsstarke Gruppe z.B. den Barcode-Generator in SCRATCH implementieren (siehe Kasten „Aufgabe Gruppe 6: ‚Strichcode-Generator‘“, nächste Seite).

In dem Beispiel „Strichcode-Scanner“ wird die Differenzierung durch den Unterrichtenden vorgenommen, weil er die Gruppen nach Leistungsstärke zusammensetzen oder eine Zusammensetzung so wählen kann, dass defizitäre Kompetenzbereiche gestärkt werden. Unterschiedlich komplexe Aufgabenstellungen sind hier möglich, weil der Kontext „Strichcode“ den „Zusammenhalt“ der einzelnen Aufgaben bildet. Wie

Aufgabe Gruppe 6: „Strichcode-Generator“

Ziel: Erzeugung des Strichcodes nach Eingabe der Artikelnummer.

Das Objekt *Strich* bekommt zwei Kostüme (ein weißes und ein schwarzes), wie in der Abbildung ersichtlich. Dann gibt der Bediener den EAN-8-Code ein.



Der Strich läuft nun von links nach rechts über eine graue Bühne und hinterlässt mittels **stamp** Abdrücke. Dabei ist ein Strich der „Breite 3“ dasselbe wie 3 Striche der „Breite 1“. Achtet also darauf, wann der Strich das Kostüm wechseln muss!

Ziffer	Strichbreiten	Ziffer	Strichbreiten
0	3211	5	1231
1	2221	6	1114
2	2122	7	1312
3	1411	8	1213
4	1132	9	3112

Hinweise:

- ▷ Den Anfang der Codierung bilden die Strichkombinationen Schwarz-Weiß-Schwarz. Dann folgt die Codierung von 4 Zahlen, wie in der Tabelle ersichtlich. In der Mitte kommen 5 einzelne Striche mit abwechselndem Kostüm hinzu und nach der Codierung 4 weiterer Zahlen am Ende noch einmal 3 einzelne Striche abwechselnd in Schwarz und Weiß. Hilfefkarten liegen auf dem Lehrerpult.
- ▷ Jedes Strichobjekt in der Abbildung ist 3 Pixel breit. Also muss das Strichobjekt nach jedem Abdruck 3 Pixel weiter gehen und eventuell sein Kostüm ändern.



bei der Aufgabe „Fieberscan“ (siehe Kasten „Aufgabe: Fieberscan“, Seite 89) steht auch hier der Zweck, einen Strichcode-Scanner gemeinsam zu rekonstruieren, im Vordergrund, und die Rekonstruktion eines Strichcode-Scanners liefert eine bessere Motivation als die sich ergebenden innerinformatischen Fragestellungen.

Offene kontextbezogene Aufgaben

Differenzierung kann auch durch *offene Aufgaben* erreicht werden, indem nicht die Lehrkraft die Aufgaben-

stellungen differenziert, sondern die Schülerinnen und Schüler selbst die Lösung eines gegebenen Problems individuell so gestalten können, dass unterschiedlich komplexe Lösungen möglich sind (siehe auch die Beiträge von Jürgen Poloczec, Seite 33ff., bzw. Peter Brichzin und Martin Hölzel, Seite 111ff., in diesem Heft). Dazu muss die Aufgabenstellung einen offenen Anteil beinhalten, zu dem die Schülerinnen und Schüler ihre eigenen Lösungsideen kreativ umsetzen können. Stellen einige von ihnen ihre Aufgabenlösungen im Anschluss an die Bearbeitungsphase vor, wird deutlich, dass die nun vorgestellten Produkte im Anspruch divergent sind, aber auch schwache Schülerinnen und Schüler durch inhaltlich zwar wenig komplexe, aber beispielsweise einfallsreiche Zusätze für alle anderen gewinnbringende, eigene Lösungen vorzuweisen haben.

Bei selbstdifferenzierten Aufgaben kann der *Kontext* für zwei positive Effekte sorgen: Einerseits bedarf es keiner langatmigen Erklärungen oder Beschreibungen, die in eine bestimmte Lösungsrichtung weisen. Andererseits ist der Kontext ein Impuls dazu, dass die Schülerinnen und Schüler eigene Ideen finden, entwickeln und umsetzen.

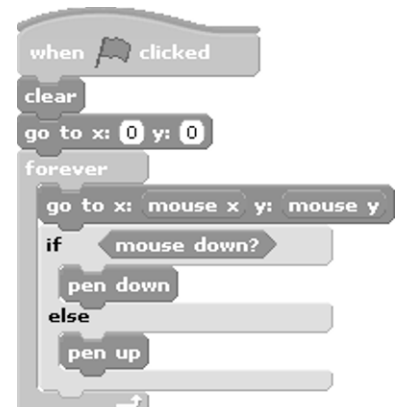
Die Aufgabe zum Zeichenprogramm *Microsoft Paint* für Schülerinnen und Schüler mit wenig Vorerfahrung in der Programmierung mit SCRATCH (siehe Kasten „Aufgabe: Zeichenprogramm ‚Paint‘“) hat als Kontext die Rekonstruktion von Teilen eines bekannten Softwaresystems.

Die Lösungen der Schülerinnen und Schüler sind immer sehr unterschiedlich. Schon die vorgegebenen Funktionen werden unterschiedlich implementiert: Ob durch einfarbige rechteckige Objekte, die man zum Farbenwechseln anklicken muss, oder Tastenkombinationen zum Wechseln der Farbe oder Stiftdicke finden sich völlig unterschiedliche Ansätze. Weitere Zusätze, die die Schülerinnen und Schüler realisieren, sind beispielsweise das Clear des Bildschirms auf ein Ereignis hin, was vom Anspruch eher gering ist. Es finden sich aber auch immer wieder sehr komplexe Lösungen, wie

Aufgabe: Zeichenprogramm „Paint“

Ziel: Implementierung eines Programms, mit dem man wie in „Paint“ auf der Bühne zeichnen kann. Implementiert den nebenstehenden Algorithmus für ein ein-Pixel großes Objekt.

Führt man das Programm aus, kann man mit der Maus auf der Bühne zeichnen. Für ein Zeichenprogramm ist diese eine Funktionalität noch sehr spärlich. Deshalb soll das Programm erweitert werden, in dem man z.B. die Farbe wählen kann oder die Stiftdicke oder ...



Steckbrief zur Aufgabe „Zeichenprogramm ‚Paint‘“

- (1) *Jahrgangsstufe/Schulart/Autor (Bundesland)*
Jahrgangsstufe: 8–10.
Schulart: Gymnasium.
Autor (Bundesland): Kerstin Strecker (Niedersachsen).
- (2) *Fachliche Thematik*
 Ein bekanntes Softwaresystem rekonstruieren, Programmierung mit SCRATCH.
- (3) *Vorwissen*
 Wenig Vorerfahrung in der Programmierung mit SCRATCH.
- (4) *Einsatzbereich der Aufgabe*
 Übungsaufgabe.
- (5) *Zeitlicher Umfang*
 Mit Vorstellung einer Lösung eine Doppelstunde.
- (6) *Aufgabenstellung, (7) Lösungsskizze*
 Siehe Text, Seite 93f.
- (8) *Einordnung in die Bildungsstandards*
 Schwerpunkte sind der Prozessbereich *Implementieren* sowie der Inhaltsbereich *Algorithmen*.

beispielsweise das „Aufziehen“ von Rechtecken mit der Maus (wie in *Paint*). Eine bemerkenswerte Lösung (nicht besonders komplex, aber beeindruckend) war die eines Schülers, bei der der Programmcode vier verschiedenen Objekten zugewiesen und die Anweisung

```
go to x: mouse x y: mouse y
```

bei den anderen drei Objekten jeweils durch

```
go to x: -1 * mouse x y: -1 * mouse y
```

```
go to x: -1 * mouse y y: -1 * mouse x
```

```
go to x: mouse y y: mouse x
```

ersetzt wurde und auf diese Weise sehr beeindruckende symmetrische Figuren erzeugt wurden. Natürlich ist das bei der nächsten Lerngruppe nicht auch zu erwarten. Die Lerngruppe hat dann aber wieder ihre ganz eigenen kreativen Lösungen, die eine Lehrperson an dieser Stelle selbst vielleicht gar nicht im Blick hat.

Was aber an diesem Beispiel verdeutlicht werden soll, ist Folgendes: Der Kontext, das Zeichenprogramm *Paint*, ist den Schülerinnen und Schülern bekannt. Deshalb haben sie auch sofort Ideen, was sie umsetzen können. Durch die Vorgabe des Wechsels der Farbe und der Stiftstärke sind zwei sehr einfache Erweiterungen vorgegeben, die jede Schülerin bzw. jeder Schüler umsetzen kann. Somit ist jeder der Lernenden auch erfolgreich, niemand ist überfordert, weil die Grundfunktionalität schnell hergestellt werden kann. Irgendeine Erweiterung ist in jedem Fall möglich. Bei der anschließenden Vorstellung der Produkte sind alle Lösun-

gen Unikate und so auch die Präsentation weniger komplexer Lösungen für die anderen Schülerinnen und Schüler ein Gewinn, möglicherweise durch ihren Einfallreichtum, wie die Erzeugung symmetrischer Bilder. Der Kontext *Paint* aus dem Erfahrungsbereich der Schülerinnen und Schüler hat es ermöglicht, dass alle sofort Ideen zur Umsetzung hatten.

Fazit

Bei einer kontextbezogenen Aufgabe

- ▷ steht der Zweck deutlich im Vordergrund und nicht das innerinformatische Problem.
- ▷ sollte der Kontext Differenzierung ermöglichen,
 - entweder, indem der Kontext der Lehrperson das Aufteilen in unterschiedliche Aufgabenstellungen mit unterschiedlichen Anforderungen ermöglicht,
 - oder indem der Kontext ein Problem umreißt, für das die Schülerinnen und Schüler selbst individuelle Lösungen finden können.

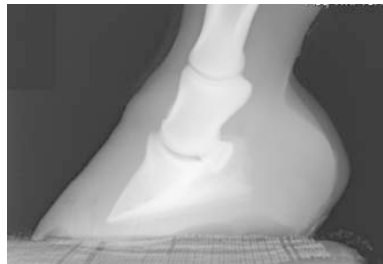
Kontexte können aus verschiedenen (Fach-)Bereichen gewählt werden. Der Kontext sollte neugierig machen und vielleicht auch einmal aus anderen als den typischen (Fach-)Bereichen kommen. Da Informatik oft eher anhand technischer und mathematischer Problemstellungen unterrichtet wird, könnte z.B. ein sozialer und medizinischer Kontext wie bei der Aufgabe „Fieberscan“ (siehe Seite 89) helfen, auch andere Zielgruppen zu motivieren. Abschließend dazu sei ein weiteres Beispiel für eine Aufgabenstellung mit einem veterinärmedizinischen Kontext gezeigt (siehe Kasten „Aufgabe: Hat das Pferd Hufrehe?“, nächste Seite).

Steckbrief zur Aufgabe „Hat das Pferd Hufrehe?“

- (1) *Jahrgangsstufe/Schulart/Autor (Bundesland)*
Jahrgangsstufe: 8/9.
Schulart: Gymnasium.
Autor (Bundesland): Kerstin Strecker (Niedersachsen).
- (2) *Fachliche Thematik*
 Ein bekanntes Softwaresystem rekonstruieren, Programmierung mit SCRATCH.
- (3) *Vorwissen*
 Vorerfahrung in der Programmierung mit SCRATCH.
- (4) *Einsatzbereich der Aufgabe*
 Übungsaufgabe.
- (5) *Zeitlicher Umfang*
 Eine Doppelstunde.
- (6) *Aufgabenstellung, (7) Lösungsskizze*
 Siehe Text, nächste Seite
- (8) *Einordnung in die Bildungsstandards*
 Schwerpunkte sind der Prozessbereich *Modellieren* und *Implementieren* sowie der Inhaltsbereich *Algorithmen*.

Aufgabe: Hat das Pferd Hufrehe?

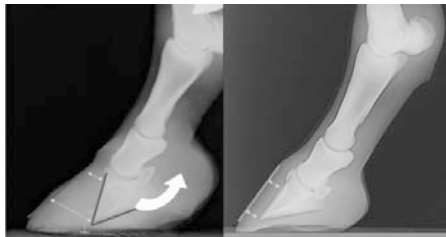
Mit Hufrehe bezeichnet man beim Pferd eine Entzündung im Huf, die solche Schmerzen verursacht, dass das Pferd nicht mehr laufen kann. Erkrankt das Pferd öfter daran oder wird die Hufrehe



nicht therapiert, dann kann es zu einer Absenkung des Hufbeins (Knochen) kommen. Auf dem Röntgenbild zeigt sich diese Absenkung, wenn das Hufbein nicht mehr parallel zur Hufoberfläche verläuft.

Die Abbildungen unten zeigen links eine Absenkung und rechts einen gesunden Huf.

Ladet in SCRATCH das Röntgenbild als Bühne und schreibt ein Programm, das den Tierarzt beim Stellen der Diagnose „Hufbeinabsenkung“ unterstützt.



Fotos: Pferdeklinik Großostheim

Tipp:

Der Tierarzt ist von anderen Untersuchungen gewohnt, mit der Maus oder mit Tasten Objekte auf dem Bildschirm zu bewegen.

Lösungsskizze:

Eine Lösungsmöglichkeit ist, ein Punkt-Objekt mit der Maus o. Ä. zu steuern und auf Tatendruck hin dessen Koordinaten zu speichern sowie einen Abdruck des Objekts auf der Bühne zu hinterlassen. Mithilfe der Koordinaten verschiedener Punkte kann der Abstand zwischen Hufbein und Hufoberfläche bestimmt werden. Ist der Abstand immer in etwa gleich groß, deutet das auf eine Parallelität hin. Die Diagnose ist ein gesunder Huf. Alternativ kann beispielsweise auch mit Linien gearbeitet werden, die man mit der Maus drehen kann und bei denen ein aussagekräftiger Winkel angezeigt wird.

Hier ist der Zweck, den Tierarzt bei seiner Arbeit zu unterstützen und das Pferd vor Schmerzen zu schützen, offensichtlicher als das innerinformatische Problem.

Außerdem gibt es viele unterschiedlich komplexe Möglichkeiten dies Problem zu lösen, was für eine Binnendifferenzierung sorgt. Eine einfache Lösung wird im Kasten links (unten) skizziert.

Dr. Kerstin Strecker
Hainberg-Gymnasium
Friedländer Weg 19
37085 Göttingen

E-Mail: Kerstin.Strecker@gmx.de

Literatur und Internetquellen

AKBSI – Arbeitskreis „Bildungsstandards“ der Gesellschaft für Informatik (Hrsg.): Grundsätze und Standards für die Informatik in der Schule – Bildungsstandards Informatik für die Sekundarstufe I. Empfehlungen der Gesellschaft für Informatik e.V. vom 24. Januar 2008. In: LOG IN, 28. Jg. (2008), Nr. 150/151, Beilage.
http://www.gi.de/fileadmin/redaktion/empfehlungen/Bildungsstandard_s_2008.pdf

Brichzin, P.; Hölzel, M.: Diagnoseaufgaben – Diagnosepotenziale bei Aufgaben zum Leisten ausschöpfen. In: LOG IN, 33. Jg. (2013/2014), Nr. 176/177, S. 111–121 (in diesem Heft).

Diethelm, I.; Koubek, J.; Witten, H.: IniK – Informatik im Kontext – Entwicklungen, Merkmale und Perspektiven. In: LOG IN, 31. Jg. (2011), Nr. 169/170, S. 97–105.

Koubek, J.; Schulte, C.; Schulze, P.; Witten, H.: Informatik im Kontext (IniK) – Ein integratives Unterrichtskonzept für den Informatikunterricht. In: B. Koerber (Hrsg.): Zukunft braucht Herkunft – 25 Jahre „INFOS – Informatik und Schule“. INFOS 2009 – 13. GI-Fachtagung Informatik und Schule, 21.–24. September 2009 in Berlin. Reihe „GI-Edition Lecture Notes in Informatics“, Band P-156. Bonn: Köllen Verlag, 2009, S. 268–279.
<http://subs.emis.de/LNI/Proceedings/Proceedings156/268.pdf>

Magenheim, J.: Informatiksystem und Dekonstruktion als didaktische Kategorien – Theoretische Aspekte und unterrichtspraktische Implikationen einer systemorientierten Didaktik der Informatik. 2000.
http://ddi.uni-paderborn.de/fileadmin/Informatik/AG-DDI/Veroeffentlichungen/Paper/2000/sytemorientierter_ansatz.pdf

Modrow, E.: Informatik als technisches Fach – Kann ein technisch orientierter Informatikunterricht die Einstellung zu technischen Berufen beeinflussen? In: LOG IN, 30. Jg. (2010), Nr. 163/164, S. 38–42.

Poloczek, J.: Zur Konstruktion von Aufgaben – Aufgabentypisierung und das Konzept der Blütenaufgaben. In: LOG IN, 33. Jg. (2013/2014), Nr. 176/177, S. 33–39 (in diesem Heft).

Alle Internetquellen wurden zuletzt am 14. März 2014 geprüft.