

# Benutzen – Analysieren – Gestalten – Verankern

Ein didaktischer Vierschritt im Informatikunterricht

von Lutz Hellmig und Tino Hempel

Informatik bietet Schülerinnen und Schülern – ähnlich wie der naturwissenschaftliche Unterricht – vielfältige Möglichkeiten, ihr Wissen unter Einbeziehung experimenteller Methoden durch das eigene Handeln selbstständig und individuell zu entwickeln. (vgl. Micheuz u. a., 2011; Hellmig/Gramm, 2013). In den Naturwissenschaften sind dem selbstbestimmten Handeln der Schülerinnen und Schüler jedoch Grenzen gesetzt: Arbeitsschutzvorschriften grenzen ihren Handlungsspielraum (vernünftigerweise!) ein; Untersuchungsobjekte und Materialien stehen nicht unbegrenzt zur Verfügung, sodass Experimentialgorithmen weitgehend vorgegeben werden müssen, um Schaden für Menschen und Material zu vermeiden. Das ist ein gravierender Unterschied zum Informatikunterricht, in dem derartige Einschränkungen kaum eine Rolle spielen. Nur im Informatikunterricht haben Schülerinnen und Schüler die Möglichkeit, unbeschwert und risikofrei Informatiksysteme zu erkunden, um daraus Erkenntnisse zu gewinnen. Das sich daraus ergebende einzigartige Potenzial des Informatikunterrichts kann mithilfe der fachspezifischen didaktischen Schrittfolge

- ▷ Benutzen,
- ▷ Analysieren,
- ▷ Gestalten,
- ▷ Verankern

erschlossen werden.

Informatiklehrkräften in Mecklenburg-Vorpommern ist die folgende Passage aus den Rahmenplänen bereits seit Jahren bekannt: „Als didaktisches Prinzip bei der Umsetzung des Rahmenplanes im Unterricht eignet sich im Allgemeinen das Prinzip *Benutzen, Analysieren, Gestalten*“ (MBWK, 2008, S.2). Auf sechs Zeilen erfolgt daraufhin eine kurze Inhaltsbeschreibung zu den einzelnen Phasen (MBWK, 2008, S.2):

In der ersten Phase, dem *Benutzen*, verwenden die Schülerinnen und Schüler Informatiksysteme. Mit den in dieser Phase gesammelten Erfahrungen wird in der zweiten Phase, dem *Analysieren*, der Aufbau und die Arbeitsweise des Informatik- bzw. seines konkreten Anwendungssystems untersucht. Da-

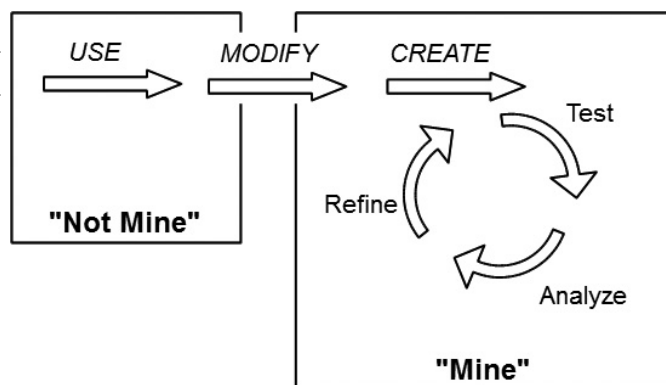
durch werden die Voraussetzungen geschaffen, dass die Schülerinnen und Schüler in der dritten Phase *Gestalten*, Lösungen selbstständig finden und die untersuchten Informatik- und deren Anwendungssysteme in neuen Zusammenhängen sinnvoll und effizient einsetzen.

Die Suche nach einer ausführlicheren Erläuterung des Prinzips oder einer lerntheoretischen Begründung desselben führte ins Leere. Immerhin konnte mit Norbert Breier der Urheber der Idee ausfindig gemacht werden, der den Autoren dieses Artikels bestätigte, dass zu dem Thema bislang keine Publikationen erschienen sind. Mit dem vorliegenden Artikel soll diese Lücke geschlossen werden: Nach einer Einordnung des Konzepts in einen theoretischen Kontext und dem Hinzufügen des vierten Schritts, des *Verankerns*, werden zunächst alle Phasen an einem prototypischen Beispiel beschrieben und allgemeine Merkmale für deren methodische Gestaltung und den Entwurf der Arbeitsmaterialien herausgearbeitet.

## Theoretischer Kontext

Die Beschreibung der Methode „Benutzen – Analysieren – Gestalten“ in den Rahmenplänen Mecklenburg-Vorpommerns bezweckt offenbar eine Abkehr vom klassisch-deduktiven Unterricht, wie er beispielsweise in den neun Schritten des Instruktionsdesigns nach Robert Gagné beschrieben wird (vgl. Gagné/Briggs, 1974), bei dem erst die Theorie präsentiert werden muss, bevor praktische Übungen unter Nutzung von Informatiksystemen folgen können. Trotzdem lohnt ein Blick auf die Ideen Gagnés. Interessanterweise wird der vierte Punkt seiner *9 Events of Instruction* im Deutschen häufig als pure Präsentation des zu erlernenden Wissens verstanden, entsprechend übersetzt und in die pädagogische (Un-)Tat umgesetzt. Im Original ist bei Gagné (zumindest in den späteren Fassungen seines Modells) aber von „*Present stimulus material*“ die Rede. Wenn den Schüle-

Quelle: Lee u. a., 2011, S. 35



**Bild 1: Use – Modify – Create.**

rinnen und Schülern tatsächlich *stimulierendes*, also eine Handlung anregendes Material angeboten wird, sind die Voraussetzungen für die weitgehend selbstständige Erschließung der informatischen Konzepte mithilfe der Nutzung konkreter Artefakte durch sie gegeben – und somit konsequenterweise eine Zuwendung zu induktiven Lernszenarien, bei denen die Theorie erst im Laufe des Lernprozesses durch die Schülerinnen und Schüler entwickelt wird.

Bemerkenswert ist, dass die intuitiv erstellte didaktische Stufung *Benutzen – Analysieren – Gestalten* auch empirisch gestützt wird. Die drei Stufen finden ihre Entsprechung im Kompetenzbereich K1 „Bewältigung von systembezogenen Aufgabenbereichen“ des im MoKoM-Projekt anhand von Experteninterviews ausdifferenzierten Kompetenzstrukturmodells (vgl. Magenheimer u. a., 2012, S.116):

Phasen des forschend-entwickelnden Unterrichts		B-A-G-V
Problemgewinnung	Problemgrund	Benutzen
	Problemfindung	Analysieren
	Problemformulierung	Analysieren
Überlegungen zur Problemlösung	Analyse des Problems	Analysieren
	Vorschläge zur Problemlösung	Gestalten
	Entscheidung für einen Lösungsvorschlag	Gestalten
Durchführung des Problemlösungsvorschlags	Planung des konkreten, praktischen Tuns	Gestalten
	Praktische Durchführung des Lösungsvorschlags	Gestalten
	Erörterung und Zusammenfassung der Ergebnisse	Verankern
Abstraktion der gewonnenen Erkenntnisse (Theoriebildung)	Ikonische Abstraktion	Verankern
	Verbale Abstraktion	Verankern
	Symbolhafte Abstraktion	Verankern
Wissenssicherung	Anwendungsbeispiele (Transfer)	Verankern
	Wiederholung des Inhalts und der Denkphasen	Verankern
	Lernzielkontrolle	Verankern

- ▷ K1.1 Systemanwendung,
- ▷ K1.2 Systemverständnis und
- ▷ K1.3 Systemgestaltung.

Zu den Ergebnissen der Expertenbefragungen gehörte unter anderem, dass insbesondere der Kompetenzbereich K1.1 „zu wenig zusammen mit den fundamentalen Ideen der Informatik thematisiert und geübt wird“ (Magenheimer u. a., 2012, S.116).

Irene Lee und andere betonen ebenfalls die Bedeutung einer Einstiegsphase, in der der Umgang mit informatischen Artefakten den Ausgangspunkt für das zielgerichtete, selbstständige Gestalten informatischer Systeme und der Gewinnung informatischer Erkenntnisse bildet (vgl. Lee u. a., 2011; siehe auch Bild 1).

Im Modell von Irene Lee u. a. ist gut ersichtlich, dass die Abfolge des Modifizierens, Analysierens und Gestaltens für gewöhnlich ein zyklischer Prozess ist, in dem der Eigenanteil der Schülerin bzw. des Schülers am Gesamtergebnis zusehends steigt.

Aus welchen Gründen spielt der direkte Umgang mit informatischen Artefakten eine so geringe Rolle in der Unterrichtspraxis? Eine mögliche Ursache könnte sein, dass adäquate Phasen im Unterricht anderer Fächer nicht bekannt, ja sogar vielfach gar nicht möglich sind. Eine Abbildung von *Benutzen – Analysieren – Gestalten* auf die Phasen des forschend-entwickelnden Unterrichts in den Naturwissenschaften zeigt (vgl. Schmidkunz/Lindemann, 2003), dass der Aspekt des Benutzens nur für die zeitlich knappe Phase der Darstellung des Problemgrunds und ggf. der Problemfindung genutzt werden kann (siehe Tabelle 1).

Aus der Gegenüberstellung erwächst im Weiteren die Erkenntnis, dass das dreistufige Modell für die Gewinnung und Vertiefung neu erworbenen Wissens zu kurz greift. Die Zuordnung der Phasen *Benutzen, Analysieren und Gestalten* zum Modell des forschend-entwickelnden Unterrichts endet schon auf halber Strecke mit dem Umsetzen eines Lösungswegs. Die wichtigen Schritte der Erörterung und Zusammenfassung eines Ergebnisses, eine Abstraktion der gewonnenen Erkenntnisse sowie die Wissenssicherung erfordern die Erweiterung der Schrittfolge um das *Verankern* als vierten didaktischen Schritt. (siehe Tabelle 1)

Bevor die einzelnen Phasen des Prinzips allgemein charakterisiert werden, wird die Schrittfolge anhand eines prototypischen Unterrichtsbeispiels im Folgenden verdeutlicht.

**Tabelle 1.**

	A	B	C	D	E	F	G
1	Vorstellung		öffentliche Generalprobe	Premiere	2. Vorstellung	3. Vorstellung	4. Vorstellung
2	Eintrittspreis		2	5	4	4	
3	Zuschauer	Lehrer	7	10			
4	Zuschauer	Schüler	8	30			
5							
6	Einnahmen		30	200	0		

**Bild 2 (oben): Datenansicht der Kalkulationstabelle.**

**Bild 3 (unten): Formelansicht der Kalkulationstabelle.**

	A	B	C	D	E	F	G
1	Vorstellung		öffentliche Generalprobe	Premiere	2. Vorstellung	3. Vorstellung	4. Vorstellung
2	Eintrittspreis		2	5	4	4	
3	Zuschauer	Lehrer	7	10			
4	Zuschauer	Schüler	8	30			
5							
6	Einnahmen		=C2*(C3+C4)	200	=E2*(E3+E4)		

## Erstes Beispiel: Formeln in Tabellenkalkulationen

Das zentrale Merkmal, das die Tabelle eines Kalkulationsprogramms von einer Tabelle in einer Textverarbeitung unterscheidet, ist die Möglichkeit, funktionale Abhängigkeiten zwischen Werten abzubilden und die abhängigen Werte beim Verändern der Parameter automatisch aktualisieren zu lassen. Daraus lässt sich für die Schülerinnen und Schüler das *Lernziel* ableiten, Formeln in Tabellenkalkulationen unter Verwendung von Zellbezügen und Grundrechenarten aufstellen zu können.

Während in den *Bildungsstandards Informatik* das Aufstellen von Formeln in einer Tabellenkalkulation überraschenderweise keine explizite Erwähnung findet (vgl. GI, 2008), rechtfertigen landesspezifische Curricula dieses Thema (z.B. Tabellen analysieren; Formeln zur Problemlösung entwickeln und an die Tabellenstruktur anpassen; vgl. MBWK, 2002a und 2002b).

Zur Einordnung der Unterrichtseinheit sind die *Lernvoraussetzungen* der Schülerinnen und Schüler zu berücksichtigen. Für die Realisierung dieses Unterrichtsbeispiels sollten die Schülerinnen und Schüler schon das Eintragen von Zahlen und Zeichenketten in die Kalkulationstabelle beherrschen (motivierbar beispielsweise durch die Diagrammerstellung aus statischen Daten), numerische Werte als Währung formatieren und die Feldbezeichnung in A1-Notation angeben können.

### Benutzen

Die Lernenden erhalten als *Material* eine präparierte Kalkulationstabelle, anhand derer sie die Aufgaben des Arbeitsblatts bearbeiten (siehe Bild 2).

Würde man die Formeln der Tabelle einblenden, ergäbe sich das Bild 3.

Die *Aufgaben* des Arbeitsblatts beziehen sich auf einen vorangestellten Kontext – in diesem Falle den Betrieb eines Schülertheaters:

Die Mitglieder einer Theatergruppe haben im Wahlpflichtfach „Darstellendes Spiel“ ein Stück einstudiert, das sie im Schultheater mit 40 Zuschauerplätzen aufführen wollen, solange ausreichend Besucher kommen. Um den Überblick über die Einnahmen zu behalten, möchten sie eine Tabellenkalkulation nutzen. Von einer anderen Theatergruppe haben sie dazu die Datei *eintrittsgelder.ods* erhalten – leider ohne Informationen darüber, wie die Tabelle zu benutzen ist.

1. In der Datei *eintrittsgelder.ods* befinden sich einige Daten. Formatiere dort, wo es sinnvoll ist, die Daten mit Währungssymbolen.
2. Ändere Eintrittspreise und Zuschauerzahlen. Beobachte die Inhalte in Zeile 6.
3. Beschreibe Probleme, die Dir beim Benutzen der Tabelle auffallen.

### Hinweise zur Aufgabe

Aufgabe 1 dient dem Hineinfinden in die Aufgabe durch das Erfassen der Tabellenstruktur und der Art der verwalteten Daten. Lediglich in Zeile 2 und 6 sind die Zelleneinträge als Währung zu formatieren.

Eine erhöhte Anforderung an das inhaltliche Erfassen der Tabelle stellt die im Bild 4 (nächste Seite) wiedergegebene Variante dar.

In die Zelle H2 wurde mit einer Summe der Einzelkartenpreise pro Veranstaltung eine völlig sinnlose Angabe lanciert. Dies stellt einen guten Anlass dar, um den Informationsgehalt von Daten kritisch zu hinterfragen.

Nachdem die Schülerinnen und Schüler ihr Wissen um das Verhältnis von Daten und deren Formatierung

	A	B	C	D	E	F	G	H
1	Vorstellung		öffentliche Generalprobe	Premiere	2. Vorstellung	3. Vorstellung	4. Vorstellung	Gesamt
2	Eintrittspreis		2	5	4	4		15
3	Zuschauer	Lehrer	7	10				17
4	Zuschauer	Schüler	8	30				38
5								
6	Einnahmen		30	200	0			230

**Bild 4: Anreicherung des Problems – Daten ohne Aussage in einer Kalkulationstabelle.**

in Aufgabe 1 reaktivieren konnten, erkennen sie durch die Bearbeitung von Aufgabe 2 die funktionalen Abhängigkeiten der Werte in Zeile 6 von den darüber eingetragenen Werten. Eine Ausnahme bildet die Zelle D6, die nicht wie die anderen der Zeile reagiert und einen konstanten Wert anzeigt. Das unterschiedliche Verhalten der Zellen wird in Aufgabe 3 sprachlich beschrieben und motiviert den nächsten Schritt, das Analysieren.

## Analysieren

Den unformatierten Inhalt der markierten Tabellenzelle kann man in der Bearbeitungszeile über den Spaltenbeschriftungen sehen (siehe Bild 5).

4. Betrachte den tatsächlichen Inhalt der Zellen in Zeile 6 in der Bearbeitungszeile und versuche, die Ursache für die in Aufgabe 3 gefundenen Probleme zu beschreiben.

## Hinweise zur Aufgabe

Durch den Impuls, den Inhalt der Bearbeitungszeile mit dem in der Zelle sichtbaren Wert zu vergleichen, erkennen die Schülerinnen und Schüler, dass in den Feldern der Zeile 6, die immer den richtigen Wert anzeigen, keine Werte, sondern Ausdrücke stehen, die mit einem „=“-Zeichen eingeleitet werden. Die Werte in Zeile 6 sind offenbar funktional von den darüberstehenden Werten abhängig.

A1		$\Sigma$	=	7,89
	A	B		
1	7,89 €			

**Bild 5: Anzeige eines unformatierten Zelleninhalts in der Bearbeitungszeile.**

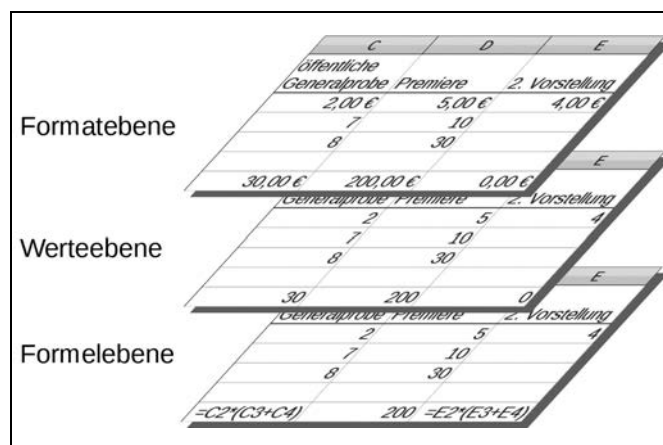
Modellhaft dringen die Schülerinnen und Schüler auf die unterste Ebene des Drei-Ebenen-Modells der Tabellenkalkulation nach Michael Gieding vor (siehe Bild 6; vgl. Gieding, 2003). Sie erkennen, dass Zellwerte sowohl als Konstanten eingetragen werden als auch das Ergebnis einer automatischen und stets aktuellen Berechnung sein können. Diese trivial klingende Aussage ist ein entscheidender kognitiver Schritt – wie oft sieht man, dass funktional abhängige Werte mithilfe eines Taschenrechners ermittelt werden und dann säuberlich als Konstante (und nicht als Formel) in eine Zelle eingetragen werden.

## Gestalten

5. Behebe die erkannten Probleme in Zeile 6 der Tabelle.
6. Für die Statistik ist auch die Gesamtzahl aller Zuschauer interessant. Die automatisch ermittelte Gesamtzahl der Zuschauer jeder Vorstellung soll in Zeile 5 zu sehen sein. Löse diese Aufgabe.
7. Mittlerweile hat die Theatergruppe beschlossen, dass Lehrer den doppelten Eintrittspreis bezahlen sollen. Verändere die Tabelle entsprechend.
8. Entwickle weitere Ideen, um die Tabelle zu verbessern und realisiere diese.

## Hinweise zur Aufgabe

Die Behebung des Fehlers in Zelle D6 erfolgt durch Analogiebildung. Die Auswahl einer Formel, die nur



**Bild 6: Das Drei-Ebenen-Modell der Tabellenkalkulation (nach: Gieding, 2003).**

aus Grundrechenarten besteht, erfolgte bewusst als allererstes Beispiel. Die Schüler sind so gezwungen, explizit über die Zusammenhänge zwischen den Daten nachzudenken. Zum Gestalten gehört neben der Berichtigung von Fehlern auch die Weiterentwicklung der Tabelle um sinnhafte Daten. Dabei wurde darauf geachtet, auch Fragestellungen einzubeziehen, die mehr als eine simple Addition von Werten erfordern.

Aufgabe 8 besitzt den größten Grad an Offenheit. Hier können insbesondere Schülerinnen und Schüler gefordert werden, die die „Pflichtaufgaben“ problemlos bewältigen. Die Berücksichtigung des Umstandes etwa, dass das Theater insgesamt nur 40 Zuschauern Platz bietet, dürfte für einiges Nachdenken sorgen.

## Verankern

9. Du hast das vorstehende Problem durch die Verwendung von Formeln in einer Tabellenkalkulation gelöst. Formuliere Deine Erkenntnisse dazu und halte sie im Heft fest. Kannst Du ...
- erklären, welche Vorteile der Einsatz von Formeln mit sich bringt?
  - beschreiben, was beim Eingeben von Formeln zu beachten ist?
  - verschiedenartige Beispiele für korrekte Formeln angeben?
  - fehlerhafte Formeln angeben und aufzeigen, was nicht beachtet wurde?
10. Welche offenen oder weiterführenden Fragen sind durch die Beschäftigung mit den Formeln entstanden?

## Hinweise zur Aufgabe

Die hier verwendete unverbindliche Art der Fragestellung ist für Schülerinnen und Schüler geeignet, die ihren Lernfortschritt bereits relativ selbstständig reflektieren können. Sollte das nicht vorausgesetzt werden können, sind zielgerichtete, geschlossenere Fragestellungen und Aufträge zum Festhalten der Lernergebnisse, beispielsweise in Form von Zusammenfassungen oder Systematisierungen, nötig.

## Allgemeine Beschreibung der Methode

Anhand des ersten Beispiels lassen sich wesentliche Merkmale der didaktischen Schrittfolge allgemein beschreiben.

Um alle Phasen der Schrittfolge für die Schülerinnen und Schüler plausibel zu motivieren, Anregungen für eine kreative und ergebnisoffene Auseinandersetzung mit dem Thema zu geben und Möglichkeiten für eine kritische Betrachtung der verwendeten Modelle zu eröffnen, sollte eine *kontextuelle Einbindung* erfolgen.

Das *Benutzen* trägt motivierenden Charakter und dient den Schülerinnen und Schülern zur Erfassung der Situation und dem Bewusstmachen einer Problemstellung. Dies kann durch den Umgang mit einem unvollständigen oder fehlerhaften Informatiksystem bzw. durch die Bearbeitung der gleichen Aufgabe mit verschiedenen Informatiksystemen geschehen. Ein Nebeneffekt dieser Phase ist, dass die Lernenden Routine im Umgang mit Informatiksystemen erwerben. Einen vertieften und modellhaften Einblick in die Problemlage können die Schülerinnen und Schüler gewinnen, wenn sie zuvor einen Sachverhalt ohne Zuhilfenahme von Informatiksystemen nachvollziehen – etwa durch das Nachspielen und Protokollieren einer Situation oder die Lösung eines Problems mit althergebrachten Mitteln wie dem Anfertigen und Ausfüllen einer Tabelle mit Stift und Papier. Dies eröffnet die Möglichkeit, informatische Vorgänge auf der enaktiven Ebene im wahrsten Wortsinne zu begreifen und die Beziehungen zwischen Prozessen aus der Lebensumwelt der Schülerinnen und Schüler und der Informatik zu erkennen.

Zur zielgerichteten Anregung des Erkenntnisprozesses bei den Schülerinnen und Schülern bedarf es geeigneter Aufgabenstellungen, die die innewohnenden Probleme des abstrakten Modells oder dessen konkreter Umsetzung offenbaren. Zielloses Interagieren mit Informatiksystemen kostet Zeit und garantiert nicht die Entdeckung eines Problems, das Aufdecken von Grenzen und Fehlern, was die notwendige Voraussetzung für die nächste Phase ist.

Im zweiten Schritt, dem *Analysieren*, stellen die Schülerinnen und Schüler Vermutungen zu den Ursachen und zu Möglichkeiten der Behebung des Problems auf. Das gelingt umso besser, je mehr sie über methodische Kompetenzen verfügen – einschließlich der Bereitschaft, selbstständig forschend zu arbeiten. Bei Bedarf können hier unterstützende Impulse gegeben werden, die ihnen Wege weisen, so selbstständig wie möglich die fachlichen Hintergründe des Problems zu entdecken und die zugrunde liegenden Modelle zu entwickeln. Für die Analyse können beispielsweise Informatiksysteme verglichen, Bezüge zwischen Informatiksystemen und nichtinformatischen Konzepten und Verfahren hergestellt, Systemmeldungen und Code untersucht sowie ggf. geeignete Literatur verwendet werden.

Das verwendete Informatiksystem muss zwei wichtige Merkmale aufweisen:

1. Um die Zahl der Instruktionen zu minimieren, sollten die zur Analyse nötigen Informationen im Informatiksystem formalsprachlich zugänglich sein: In einer Kalkulationstabelle können Formeln analysiert werden, für Probleme der Programmierung stehen Quelltexte und Fehlermeldungen zur Verfügung, in Textverarbeitungen kann das Einblenden nicht-druckbarer Zeichen Hinweise auf den Problemgrund liefern.
2. Das untersuchte Informatiksystem muss durch die Schülerinnen und Schüler manipulierbar sein, um Hypothesen durch systematisches Experimentieren (vgl. Hellmig/Gramm, 2014) überprüfen zu können und die Phase des Gestaltens zu ermöglichen. Im ersten Beispiel ist also nicht das Tabellenkalkulati-

onsprogramm das zu untersuchende Artefakt, sondern die gegebene Kalkulationstabelle, die analysiert wird und deren Inhalte zielgerichtet verändert werden können.

In der Phase des *Gestaltens* sollten die Schülerinnen und Schüler möglichst viele Freiräume haben, die identifizierten Probleme zu beheben und an der weiteren Vervollkommen der informatischen Lösung für den gegebenen Kontext zu arbeiten. Voraussetzung für einen hohen Anteil eigenverantwortlichen Arbeitens ist, dass die zur Weiterentwicklung des Produkts nötigen Techniken bereits exemplarisch im gegebenen Artefakt verwendet worden sein sollten, um den Schülerinnen und Schülern das selbstständige Herstellen von Analogien und einen Transfer des Wissens zu ermöglichen. Im Tabellenkalkulationsbeispiel erfolgt die Verbesserung des Systems über das Implementieren weiterer Formeln. Konzepte, die in der Vorgabe keine Rolle spielten – wie im Beispiel das Verwenden von Funktionen – liegen außerhalb der Zone der nächsten Entwicklung (vgl. Vygotsky, 1978) und sind zu vermeiden.

Die Phasen des Analysieren und des Gestaltens lassen sich selten trennscharf unterscheiden. Wie im nächsten Beispiel deutlich wird, können sich diese gegenseitig durchdringen oder in zyklischer Folge wiederholen.

Mit dem letzten Schritt, dem *Verankern*, abstrahieren die Schülerinnen und Schüler vom entstandenen Produkt und werden sich ihres allgemeinen Lernfortschritts bewusst. Dies kann durch das Reflektieren und Zusammenfassen, das Formalisieren, das Systematisieren und Verknüpfen der Erkenntnisse mit vorhandenem Wissen oder Überlegungen zu möglichen Verallgemeinerungen erfolgen.

## Diskussion weiterer Unterrichtsbeispiele

Die Verzahnung des Prozessbereichs *Modellieren und Implementieren* der Bildungsstandards mit dem Inhaltsbereich *Algorithmen* spiegelt sich in den Curricula wider. Im Rahmenplan Informatik Klasse 10 in Mecklenburg-Vorpommern wird in den Hinweisen zum verpflichtend zu unterrichtenden Thema „Sprachen und Sprachkonzepte“ eine Umsetzung beschrieben, die dem informationsorientierten didaktischen Ansatz folgt und explizit den Dreischritt *Benutzen – Analysieren – Gestalten* umsetzt (vgl. u.a. MBWK, 2002a; Hempel, 2007).

### Imperatives Problemlösen im Kontext des Nim-Spiels

#### *Benutzen und Analysieren I*

Das Nim-Spiel ist bekanntlich ein Spiel für zwei Personen, bei dem abwechselnd eine Anzahl von Gegenständen, etwa Streichhölzer, weggenommen werden. Gewonnen hat beim Standardspiel derjenige, der das

letzte Hölzchen nimmt, bei der Misère-Variante verliert dagegen derjenige, der das letzte Hölzchen nehmen muss. 1940 stellte die Firma Westinghouse auf der New-Yorker Weltausstellung ihr Gerät *Nimatron* aus, und 1951 beeindruckte ein in England gebauter elektronischer Rechner namens *Nimrod* die Öffentlichkeit dadurch, dass er auf der Berliner Industrieausstellung den damaligen Wirtschaftsminister Ludwig Erhard im Nim-Spiel schlug (siehe u.a. LOG IN Nr. 157/158, S.65).

„Benutzen und Analysieren bedeutet, dass die Schüler z.B. das gewählte Spiel zunächst ohne und dann mit Computer spielen, sich dabei die Spielregeln zu eigen machen und den Spielablauf Schritt für Schritt protokollieren. Sie unterscheiden Ein- und Ausgaben und beobachten die schrittweise Abfolge und eventuelle Wiederholung einzelner Schritte oder Schrittfolgen“ (MBWK, 2002a, S.45). Dieser enaktive Einstieg dient auch dazu, die Grenzen des Spiels auszutesten. Neben einer verbalen Beschreibung des Spielalgorithmus sollten die Schülerinnen und Schüler bereits hier in einer Übersicht die offensichtlichen Programmfehler sowie mögliche Verbesserungen erfassen.

Auf dem Spieltisch liegt eine zufällige Anzahl von Hölzern. Die beiden Spieler dürfen abwechselnd ein bis drei Hölzer wegnehmen. Der Spieler, der das letzte Holz nehmen kann, hat gewonnen.

#### **Auftrag 1:** Benutzen und Analysieren des realen Spiels

- Spiele mit Deinem Partner das Spiel mehrfach.
- Beschreibe den Spielverlauf mit eigenen Worten.
- Ermittle die Bedingungen unter denen ein Spieler noch einen Zug machen kann.
- Auf welche Kriterien müssen die Spieler achten, damit nicht geschummelt werden kann?

#### **Auftrag 2:** Benutzen und Analysieren des Computerspiels

- Spiele mit Deinem Partner das Spiel auf dem Computer mehrfach.
- Prüfe, ob der Spielverlauf mit dem realen Spiel übereinstimmt. Überarbeite/konkretisiere ggf. Deine Ablaufbeschreibung.
- Ermittle die Informationen, die ein- und ausgegeben werden und die sich das Programm merken muss.
- Prüfe, ob sich der Computer überlisten lässt.
- Erstelle eine Übersicht über Fehler in der Computerversion und mache Verbesserungsvorschläge.

Entscheidend ist das den Schülerinnen und Schülern zur Verfügung gestellte Programm. Dieses ist spielbar, solange die Spieler freiwillig die Regeln einhalten. Allerdings wird nicht geprüft, ob die Zahl der genommenen Hölzer im vorgeschriebenen Rahmen (1 bis 3) liegt. Mehr noch: Durch die „Wegnahme“ einer negativen Zahl von Hölzern können wieder Hölzer auf den Haufen gelegt werden. Weiterhin endet das Spiel nur, wenn genau 0 Hölzer auf dem Tisch liegen. Nimmt man mehr Hölzer weg als da sind, gerät das Programm in eine Endlosschleife. Dies bietet ausreichend Anlass für die Analyse und Verbesserung. Das dafür nötige Handwerkszeug finden die Schülerinnen und Schüler bereits an anderen Stellen im Programm vor und haben so die

Möglichkeit, ohne Instruktion durch die Lehrkraft einen Transfer herzustellen.

## Analysieren II und Gestalten I

„Die Schüler öffnen dann die Programmierumgebung und darin den Quelltext des zuvor benutzen Programms, vergleichen den Quelltext mit dem protokollierten Spielverlauf und lernen durch dieses Rückwärtsarbeiten die zur Lösung der Aufgabe erforderlichen Elemente der Programmiersprache sowie das zugrundeliegende Programmierparadigma kennen“ (MBWK, 2002a, S.45). Je nach Rahmenbedingung können dabei visuelle Programmiersprachen wie SCRATCH oder textuelle Systeme zum Einsatz kommen. Für den Lerneffekt ist entscheidend, dass sich die Lehrkraft an dieser Stelle mit Erläuterungen zu algorithmischen Grundstrukturen, logischen Bedingungen oder Variablen zurückhält, sondern allenfalls kurze Impulse gibt.

### Auftrag 3: Teilanalyse der Implementation und Beseitigung logischer Fehler

- Gib die Aufgabe der Variable *hölzer* und der Variable *Antwort* an.
- Beschreibe den Algorithmus aus der folgenden Abbildung, die einen Programmausschnitt des Nim-Spiels enthält, mit eigenen Worten.



- Nenne den Teil des Regelwerks, der durch diesen Ausschnitt implementiert werden soll.
- Begründe, warum dies nicht vollständig gelingt.
- Beschreibe mit eigenen Worten notwendige Veränderungen in diesem Ausschnitt, um die Regel zu vervollständigen.
- Implementiere diese Veränderungen.
- Prüfe, ob die durchgeführten Veränderungen das Regelwerk für beide Spieler korrekt umsetzen. Korrigiere gegebenenfalls weiter.

Die Bearbeitung der oben genannten Aufgaben führt zum ersten Verständnis von logischen Bedingungen. Hier wird der Fehler, dass mehr Hölzer genommen werden können als noch vorhanden sind, durch die Erweiterung der Boole'schen Bedingung beseitigt. Außerdem erfolgt die Übertragung der gefundenen Lösung in den Teilalgorithmus für Spieler 2.

## Gestalten II

Auf diese Weise lassen sich nun vorhandene logische Fehler beheben und im Anschluss die gewünschten

Verbesserungsvorschläge, wie ein Rundenzähler oder die Möglichkeit der Spielwiederholung, umsetzen. Die dafür notwendigen Variablen, algorithmischen Grundstrukturen und logischen Bedingungen können erneut durch Analogiebetrachtungen und Wissenstransfer eingebaut werden. Die Anforderungen dafür sind jedoch höher, da die Lösungen nicht mehr an anderer Stelle im Quelltext stehen.

### Auftrag 4: Ausgestaltung der Spielimplementation

- Beschreibe eine Möglichkeit, mithilfe einer neuen Variable *runde*, die Anzahl der gespielten Runden zu zählen und am Ende auszugeben.
- Erweitere das Projekt um eine solche Option.
- Erweitere das Projekt um die Möglichkeit der Spielwiederholung.

## Verankern und Gestalten III

„Erste kleine Veränderungen am Quelltext (z.B. Einbau eines noch fehlenden Schrittzählers beim Zahlenraten-Spiel, verbesserter Bildschirmausgaben oder einer Abfrage, ob das Spiel noch einmal wiederholt werden soll) vertiefen die Kenntnisse der Schüler und sind erste Schritte auf dem Weg zum selbständigen Entwerfen von Programmen für vergleichbare Probleme, zu deren Lösung die Schüler bereits Bekanntes aufgreifen können und nur wenige neue Gestaltungselemente erforderlich sind“ (MBWK, 2002a, S.45). Daher bieten sich einfache Glücks- und Würfelspiele, wie etwa Zahlenraten, Pasch, Einundzwanzig oder Gerneklein an. Die Verankerung der neuen Erkenntnisse und Kompetenzen sollte durch deren gezielte Bewusstmachung, etwa durch Aufträge zur Gestaltung von Lernplakaten oder Übersichten erfolgen, auf denen die Schülerinnen und Schüler die Begriffe *Variable*, *logische Bedingung* oder *Schleifenstruktur* und deren Umsetzung darstellen.

### Auftrag 5: Verankern

- Dein Nim-Spiel ist nun fertig. Beschreibe in maximal vier Sätzen den Weg zu Erzeugung des fehlerfreien Spiels.
- Erstelle eine Übersicht über die Verwendung einer Variablen zur Speicherung und Anzeige der Spielrunden.

## Fazit

Durch die Schrittfolge *Benutzen – Analysieren – Gestalten – Verankern* werden die Schülerinnen und Schüler stark in den Lernprozess integriert. Um die Motivation nicht durch eine Überforderung zu gefährden, muss sich der Zuschnitt der einzelnen Lernstufen an der Zone der nächsten Entwicklung nach Lev Semyonovitch Vygotsky ausrichten – ein Prinzip, das durch Wood, Bruner und Ross (1976) mit dem Begriff des

*Scaffolding* (von *scaffold*; deutsch: Gerüst) beschrieben wurde und das eine Unterstützung des Lernprozesses durch die Bereitstellung einer ersten – vollständigen – Orientierungsgrundlage in Form von Anleitungen, Denkanstößen und anderen Hilfestellungen beinhaltet. Der optimale Zuschnitt dieser Lernstufen ist das Ergebnis fortwährenden Probierens und differiert zwischen unterschiedlichen Lerngruppen.

Einer universellen Anwendung des Verfahrens sind gewissen Grenzen gesetzt. Abgesehen davon, dass dies der erwünschten didaktischen und methodischen Vielfalt im Unterricht entgegensteht, sind nicht alle Lerninhalte im Informatikunterricht unmittelbar mit einem für die Schülerinnen und Schüler transparentem Informatiksystem verknüpft, das „lesbar“ ist und leicht zu erfassende Rückmeldungen gibt. Im Weiteren ist die Methode recht zeitintensiv, sodass ihre Anwendung nur maßvoll möglich sein wird. Ein Curriculum, das Raum für einen induktiven, forschenden Zugang zu den Inhalten bietet, ist eine gute Voraussetzung für den regelmäßigen Einsatz im Unterricht und für das Zustandekommen eines Gewöhnungseffekts bei den Schülerinnen und Schülern.

Dr. Lutz Hellmig  
Didaktik der Informatik  
Lehrstuhl für Praktische Informatik  
Universität Rostock  
18051 Rostock

E-Mail: lutz.hellmig@uni-rostock.de

Tino Hempel  
Richard-Wossidlo-Gymnasium Ribnitz-Damgarten  
Schulstraße 15  
18311 Ribnitz-Damgarten

E-Mail: t.hempel@wossidlogymnasium.de

## Literatur und Internetquellen

Gagné, R.M.; Briggs, L.J.: Principles of Instructional Design. New York (USA): Holt, 1974.

GI – Gesellschaft für Informatik (Hrsg.): Grundsätze und Standards für die Informatik in der Schule – Bildungsstandards Informatik für die Sekundarstufe I. Erarbeitet vom Arbeitskreis „Bildungsstandards“ unter Federführung von Hermann Puhmann – Empfehlungen der Gesellschaft für Informatik e.V. vom 24.01.2008. In: LOG IN, 28. Jg. (2008), Nr. 150/151, Beilage.  
<https://metager.to/hj1ha>

Gieding, M.: Programming by Example – Überlegungen zu Grundlagen einer Didaktik der Tabellenkalkulation. In: mathematica didactica, 26. Jg. (2003), Bd. 2, S.42–72.  
<https://metager.to/-1yko>

Hellmig, L.; Gramm, A.: Lernaufgaben mit experimentellem Charakter. In: LOG IN, 33. Jg. (2013), Nr. 176/177, S.96–106.

Hempel, T.: Sprachen und Sprachkonzepte – Erfahrungen und Umsetzungsideen für eine Unterrichtseinheit für die Sekundarstufe I nach dem informationsorientierten didaktischen Ansatz. In: P. Stechert (Hrsg.): Informatische Bildung in der Wissensgesellschaft. Praxisband der 12. Fachtagung „Informatik und Schule – INFOS 2007“ der Gesellschaft für Informatik e.V. 19. bis 21. September 2007 an der Universität Siegen. Reihe „Medienwissenschaften“, Band 6. Siegen: universi, 2007, S.55–56.

Lee, I.; Martin, F.; Denner, J.; Coulter, B.; Allan, W.; Erickson, J.; Malyn-Smith, J.; Werner, L.: Computational Thinking for Youth in Practice. In: ACM Inroads, 2. Jg. (2011), Nr. 1, S.32–37.  
<https://dl.acm.org/citation.cfm?doid=1929887.1929902>

Magenheim, J.; Schubert, S.; Schaper, N.: Entwicklung von qualitativen und quantitativen Messverfahren zu Lehr-Lern-Prozessen für Modellierung und Systemverständnis in der Informatik (MoKoM). In: H. Bayrhuber, U. Harms, B. Muszynski, B. Ralle, M. Rothgangel, L.-H. Schön, H.J. Vollmer, H.-G. Weigand (Hrsg.): Formate Fachdidaktischer Forschung. Empirische Projekte – historische Analysen – theoretische Grundlegungen. Reihe „Fachdidaktische Forschungen“, Band 2. Münster u. a.: Waxmann, 2012, S.109–128.

MBWK – Ministerium für Bildung, Wissenschaft und Kultur Mecklenburg-Vorpommern (Hrsg.): Rahmenplan Informatik – Regionale Schule, Verbundene Haupt- und Realschule, Hauptschule, Realschule, Integrierte Gesamtschule – Jahrgangsstufen 7 bis 10. 2002a.  
<https://metager.to/g6ux4>

MBWK – Ministerium für Bildung, Wissenschaft und Kultur Mecklenburg-Vorpommern (Hrsg.): Rahmenplan AWT – Gymnasium, Integrierte Gesamtschule – Jahrgangsstufen 7 bis 10. 2002b.  
<https://metager.to/12nhg>

MBWK – Ministerium für Bildung, Wissenschaft und Kultur Mecklenburg-Vorpommern (Hrsg.): Rahmenplan für das Fach Datenverarbeitung und Informatik in den Jahrgangsstufen 11 bis 13 am Fachgymnasium. 2008.  
<https://metager.to/v5bq4>

Micheuz, P.; Bischof, E.; Sabitzer, B.: Forschend-entdeckender Informatikunterricht – warum nicht? In: LOG IN, 31. Jg. (2011), Nr. 168, S.10–20.

Prince, M.J.; Felder, R.M.: Inductive Teaching and Learning Methods – Definitions, Comparisons, and Research Bases. In: JEE – Journal of Engineering Education, 95. Jg. (2006), Heft 2, S.123–138.

Schmidkunz, H.; Lindemann, H.: Das forschend-entwickelnde Unterrichtsverfahren. Hohenwarsleben: Westarp Wissenschaftenverlagsgesellschaft, 2003.

Vygotsky, L.S.: Mind in Society – The Development of Higher Psychological Processes. Cambridge (MA, USA); London (UK): Harvard University Press, 1978.  
<https://metager.to/o3i5i>

Wood, D.; Bruner, J.S.; Ross, G.: The Role of Tutoring in Problem Solving. In: Journal of Child Psychology and Psychiatry, 17. Jg. (1976), Heft 2, S.89–100.  
<https://metager.to/35c0r>

Die Internetquellen wurden zuletzt am 27. Dezember 2017 geprüft und können auch aus dem Service-Bereich des LOG IN Verlags (<http://www.log-in-verlag.de/>) heruntergeladen werden.