

Aktivierender Unterricht von Anfang an mit dem PRIMM-Konzept

Reinold, K. und Brichzin, P.

DOI: 10.18420/ibis-02-02-03

Zusammenfassung

PRIMM ist ein Ansatz, der dazu dient, den Programmierunterricht so zu strukturieren, dass die Lernenden selbst von Anfang an aktiv sind. Dabei erhalten sie im Sinne des Scaffolding ein „Lerngerüst“, welches die Lernenden in passenden Schritten bei der Aneignung von Programmierkompetenzen unterstützt. PRIMM steht für Predict (Vorhersagen) – Run (Ausführen) – Investigate (Untersuchen) – Modify (Verändern) – Make (kreatives Erstellen). Die Methode fördert die Analyse von Programmen; vom Lesen und Interpretieren gelangt man über kleinere Variationen hin zum eigenständigen Schreiben eines Programms.

Einleitung

Im Informatikunterricht sind zwei grundlegende Herangehensweisen weit verbreitet: Einerseits werden häufig materialgeleitete Übungen durchgeführt, die die Lernenden Schritt für Schritt zu einem gewünschten Ergebnis führen (z. B. Stem learning: Scratch cards unter <https://kurzlinks.de/primm2>). Andererseits ist der erste Teil einer Unterrichtseinheit oft geprägt von der Instruktion durch die Lehrperson: Neue Konzepte werden häufig – methodisch vermittelt durch Vortrag oder Unterrichtsgespräch – im Plenum eingeführt, dabei gemeinsam im Programm erprobt und erst nach einer Sicherungsphase dürfen die Lernenden im zweiten Teil der Unterrichtseinheit endlich selbst an den Rechnern aktiv werden. Der Übergang zum eigenständigen Modellieren und Programmieren ist bei beiden Herangehensweisen oft mit Unsicherheit und Angst verbunden, da man unvermittelt „ins kalte Wasser geworfen wird“ und zeigen muss, dass man die präsentierten Konzepte verstanden hat und auch selbst syntaktisch und semantisch in anderem Kontext fehlerfrei umsetzen kann. Das PRIMM-Konzept stellt dem einen Ansatz gegenüber, bei dem die Lernenden von Beginn an zu Handelnden werden und ausgehend von der Codeanalyse über die Codevariation hin zu einem tieferen Verständnis sowie einem kreativen und kompetenten Entwickeln von Programmen gelangen.

Ursprung und Aufbau

Das PRIMM-Konzept wurde im Jahr 2017 von Sue Sentence und Jane Waite vorgeschlagen (Sentence/Waite, 2017)

Es fußt auf dem co-konstruktiven Ansatz von Lew Wygotski. Wesentliche Aspekte dieses Ansatzes in Hinblick auf das PRIMM-Konzept sind die folgenden (Textor, 2000):

- Die Lehrperson als Aktivitätengestalter: Sie wählt geeignete Aktivitäten aus, stellt passende Materialien bereit und interagiert nach den Erfordernissen mit den Lernenden.
- Die Lehrperson als Dialogpartner: Im Austausch mit den Lernenden unterstützt die Lehrperson den Lernprozess hin zu einem tieferen Verständnis.
- Die Lehrperson im Lernprozess: Die Lehrperson beteiligt sich am Problemlösungsprozess nur so weit, wie die Lernenden Unterstützung benötigen und unterstützt dabei den Aufbau von Verbindungen zwischen den bereits erworbenen Kompetenzen und neuen Kenntnissen.

Das PRIMM-Konzept besteht nun aus den folgenden Schritten:

- *Predict*: Die Lernenden erhalten ein vorgegebenes Programm und sagen dessen Verhalten voraus. Sie zeichnen ihre Vermutung in geeigneter Form, z. B. stichwortartig, auf.
- *Run*: Die Lernenden führen das vorgegebene Programm aus und überprüfen und diskutieren dabei ihre Vermutung.
- *Investigate*: Die Lernenden beschäftigen sich genauer mit dem Code, beispielsweise durch die Diskussion geeigneter Fragen („Was würde eine bestimmte Änderung ... einer Codezeile bewirken?“, „Was ergibt sich bei veränderter Eingabe ...?“), schrittweises Nachvollziehen, Arbeiten mit dem Debugger oder Kommentieren.
- *Modify*: Die Lernenden variieren das Programm in einer Reihe von zunehmend anspruchsvolleren Übungen. Sie machen sich den Code so nach und nach mehr zueigen und gewinnen dadurch Vertrauen in ihre eigenen Fähigkeiten.
- *Make*: Die Lernenden entwerfen neue Programme unter Nutzung der neu erlernten Konzepte.

Beim PRIMM-Konzept ist die Reihenfolge und Intensität der einzelnen Phasen nicht zwingend vorgeschrieben: „*You may not be able to go through all the stages in one lesson, and you may focus on one stage more than another. Using the PRIMM framework gives you a way of labelling what you are doing when you are teaching programming.*“ (Coleman, 2021, S. 22) So kann es beispielsweise sinnvoll sein, bei komplexen Programmen zunächst mit der Run-Phase zu starten, damit ein Grundverständnis für das Ziel des Programms gewonnen wird und sich die Lernenden dem Programm annähern, um danach in der Phase Investigate die für einen bestimmten Effekt verantwortlichen Codezeilen zu finden und zu interpretieren. Durch eine Variation von Codezeilen oder Eingabewerten kann man im Anschluss in einer Predict-Phase ein tieferes Verständnis gewinnen und sich so über mehrere Zyklen an die Konzepte annähern.

Beispiele aus der eigenen Praxis

Tabellenkalkulationssysteme – Logische Funktionen

Teilaufgabe a) der untenstehenden Beispielaufgabe leitet mit *Predict* ein. Dabei ist noch kein Öffnen der Tabellenkalkulationsdokuments nötig, da die Formeln auch der Abbildung entnommen werden können – dies hilft zu fokussieren.

Bei der Vorhersage unterstützt auch die einleitende Beschreibung der Zielsetzung im Rahmen eines schülernahen Kontexts.

Run/Investigate findet in b) statt: Mit dem Tabellenkalkulationsdokument können die Lernenden ihre Antwort aus a) überprüfen und das Verhalten mit anderen Eingabewerten erkunden.

Das *Modify* ist in dieser Aufgabe nicht nur auf das Implementieren beschränkt (Teilaufgabe d), sondern über c) wird auch eine Darstellung als Modell und eine Begründung als Text gefordert. Dies steigert die Vielfalt der Kompetenzanforderungen.

In Teilaufgabe e) werden die Lernenden abschließend aufgefordert, ihre eigene Perspektive mit einzubringen und das Rechenblatt entsprechend zu erweitern. Dies ist ein erstes kleines *Make*. Das Erstellen von Rechenblättern zu anderen Kontexten mit den neu erlernten Konzepten findet dann in weiteren Aufgaben des Kapitels statt. Typischerweise wird vorher die Einstiegsaufgabe im Unterricht besprochen, um mögliche Fragen zu klären.

Unter den Eltern herrscht Frust: Viele Kinder essen täglich Pommes oder Spaghetti. Deshalb überlegt die Schulleitung des John-von-Neumann-Gymnasiums zusammen mit Mensapächter und Schülervertretung die Einführung einer Empfehlung für das Mensaessen.

Das erste Kriterium ist die biologische Erzeugung, die sichergestellt ist, wenn die Zutaten ein Biosiegel des Handels haben oder direkt von regionalen Biobauern stammen. Für den Genusswert müssen Geschmack, Aussehen und Geruch stimmen. Zum Dritten soll gesichert sein, dass die Ernährung nicht einseitig ist.

	A	B	C	D	E	F	G	H
1	Biologische Erzeugung			Genusswert			Einseitigkeit	
2	Biosiegel	WAHR		guter Geschmack	WAHR		ist einseitig?	FALSCH
3	örtlicher Anbau	FALSCH		sieht gut aus	WAHR			
4				riecht gut	WAHR			
5	empfehlenswert: =ODER(B2;B3)			empfehlenswert: =UND(E2;E3;E4)			empfehlenswert: =NICHT(H2)	

a. Das Ergebnis der Funktionen in den Zellen B5, E5 und H5 sind Wahrheitswerte. Notiere die Werte, die deiner Meinung nach berechnet werden.

b. Experimentiere mit dem vorliegenden Tabellendokument, indem du die Wahrheitswerte in den Spalten B, E, H veränderst. Untersuche die Auswirkungen in den Berechnungen von Zeile 5.

c. Die Schülervertretung wünscht sich eine Auszeichnung „rundum empfehlenswert“, wenn alle drei Kriterien gleichzeitig erfüllt sind. Begründe, welche der Funktionen aus Zeile 5 du dafür benötigst. Erstelle für die Präsentation im Schulforum ein anschauliches Datenflussdiagramm, das auch Schulleitung und Mensapächter überzeugt.

d. Ergänze das Rechenblatt um die Auszeichnung „rundum empfehlenswert“.

e. Welche Kriterien sind für dich beim Mensa-Essen wichtig? Ergänze mindestens ein zusätzliches Kriterium im Rechenblatt und passe die Formeln, soweit nötig, an.

(Übernahme von Brichzin/Jetzinger/Neumeyer/Reinold/Wiedemann, 2021, S. 44 mit geringer Adaption)

Datenmodellierung und Datenbanksysteme – Einfache Datenbankabfragen

Sicher hast du schon einmal in einem Online-Shop nach einem Kleidungsstück, einem Buch oder etwas Ähnlichem gesucht. Nach der Eingabe des Suchbegriffes bekommst du sofort die passenden Artikel angezeigt. Aber was steckt eigentlich hinter der Webseite des Shops? Viele Anwendungen, die große Mengen an Daten bereitstellen und verarbeiten, verwenden zur Datenverwaltung eine sogenannte Datenbank. Unter <https://www.dbiu.de/shop/> kannst du einen Blick hinter einen Webshop werfen.

a Öffne die oben angegebene Seite und betrachte die Ansicht im Reiter DB-Backend. Beschreibe, wie die Artikel hier dargestellt sind und vergleiche mit der Darstellung im User-Frontend.

b Stelle im User-Frontend die Filter so ein, dass dir nur die Farbe und der Preis aller T-Shirts angezeigt werden. Wende die Filter an und wechsle in das DB-Backend. Analysiere die hier angegebene Datenbankabfrage.

c Formuliere im DB-Backend eine weitere Datenbankabfrage, welche die Art und den Preis aller Artikel in der Farbe Blau zurückgibt! Überprüfe deine Eingabe, indem du nach dem Senden der Abfrage wieder zurück in das User-Frontend wechselst.

(Brichzin/Jetzinger/Neumeyer/Reinold/Wiedemann, 2021, S. 60)

Diese Aufgabe, die in einem schülernahen Kontext sehr praktisch das Kapitel Datenbank einleitet, startet direkt mit Run/Investigate. Durch den Wechsel zwischen Backend und Frontend ist sehr einfach der Blick hinter die Kulissen von Webseiten mit Datenbankabfragen möglich. Durch den Wechsel zwischen den Ansichten sind Elemente des Predict auch enthalten.

Über den Modify-Auftrag in Teilaufgabe b) können sich die Schüler*innen den Aufbau und Bestandteile einfacher SQL-Abfragen erschließen. Auf dieser Basis sind in c) neue Datenbankabfragen (Make) möglich.

The screenshot shows the 'DB-Backend' view of a web application. At the top, there are tabs for 'User-Frontend' and 'DB-Backend'. Below the tabs, the 'Datenbankmodell:' section shows a list of attributes: 'E, art:ZEICHENKETTE, typ:ZEICHENKETTE, geschlecht:ZEICHENKETTE, farbe:ZEICHENKETTE, pre'. A 'Klassendiagramm' button is visible. The 'Datenbankabfrage:' section contains a SQL query: '1 SELECT * 2 FROM Kleidung'. An 'Abfrage senden' button is present, along with a 'Schriftgröße:' control. Below the query, the 'Ergebnistabelle (148 Datensätze):' is displayed as a table with the following data:

artikelnummer	art	typ	geschlecht	farbe	preis	bildverweis
2110532	Hose	Jeans	m	blau-washed	89.95	bilder/2110532.jpg
2112044	Hose	Jogginghose	m	schwarz	17.95	bilder/2112044.jpg
2112497	Hose	Jogginghose	m	rot	24.95	bilder/2112497.jpg
2112860	Hose	jeans	w	orange	89.95	bilder/2112860.jpg

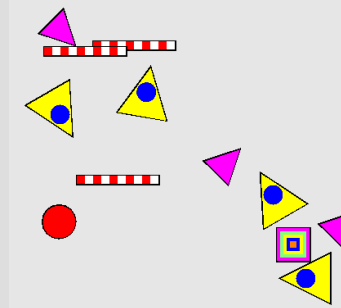
Abbildung 1: Abfragemöglichkeit im Datenbankbackend

The screenshot shows the 'User-Frontend' view. At the top, there are tabs for 'User-Frontend' and 'DB-Backend'. Below the tabs, there is a 'Filter:' section with a 'Filter anwenden' button and a list of filter options: 'Bild', 'Artikelnummer', 'Art', 'Typ', 'Geschlecht', 'Farbe', 'Preis'. The 'anwenden' button is also present. Below the filters, the 'Produkte (148):' section displays two product cards. The first card shows a pair of blue jeans with the following details: 'Artikelnummer: 2110532', 'Art: Hose', 'Typ: Jeans', 'Geschlecht: m', 'Farbe: blau-washed', 'Preis: 89.95'. The second card shows a plaid shirt with the following details: 'Artikelnummer: 3248249', 'Art: T-Shirt', 'Typ: Langarm', 'Geschlecht: w', 'Farbe: orange', 'Preis: 24.95'.

Abbildung 2: Ergebnis der Abfrage im Frontend

Objektorientierte Modellierung und Programmierung – Überschreiben von Methoden

Beim Spiel „Don't touch“ muss der rote Ball durch die Pfeiltasten gesteuert werden, so dass er nicht gegen die verschiedenartigen Hindernisse stößt.



a. Erprobe das Spiel und beobachte das unterschiedliche Aussehen und Bewegungsverhalten insbesondere von Schlagbaum und Quadrat. Vergleiche deine Beobachtung mit dem Code der Methoden `Bewegen` und `Zeichnen` in den Klassen `SCHLAGBAUM` und `QUADRAT`. Ergänze als Ergebnis deiner Beobachtungen einen kurzen Kommentar im Quelltext vor die Methode `Bewegen`, der die Bewegung beschreibt.

b. Erstelle ein Klassendiagramm zu den Klassen `HINDERNIS`, `QUADRAT` und `SCHLAGBAUM` mit den jeweiligen Methoden.

c. Der Kopf der Methoden `Bewegen` und `Zeichnen` in der Klasse `QUADRAT` bzw. `SCHLAGBAUM` hat in den meisten Programmiersprachen ein neues Syntaxelement wie `@override` bzw. `@Override`. Erkunde die Bedeutung des englischen Begriffs und versuche dann zu erklären, was bei den betrachteten Methoden gemeint ist.

Tipp: Wirf zum Vergleich auch einen Blick in die Klasse `HINDERNIS`.

d. Erstelle selbst eine neue Unterklasse von `HINDERNIS` mit neuem Aussehen und Bewegungsverhalten. (Übernahme von Brichzin/Jetzinger/Neumeyer/Reinold/Wiedemann, 2021, S. 130 mit geringer Adaption)

In Teilaufgabe a) ist *Predict* mit *Run/Investigate* eng verzahnt. *Predict* ergibt sich aus der Betrachtung des statischen Quelltexts; sobald das Programm ausgeführt wird, ist man bereits in der *Run/Investigate-Phase*.

Herausfordernd bei Quelltextanalysen für die Phasen *Predict/Investigate* ist häufig, dass das Programm einen gewissen Umfang benötigt, damit es Jugendlichen als motivierendes, interessantes Betätigungsfeld wahrnehmen. Der Umfang mag eine Gefahr sein, dass sich die Schülerinnen und Schüler im Quelltext verlieren bzw. überfordert fühlen. Deshalb ist es

wichtig, den Quelltext gut zu strukturieren und bei den Schüler*innen die Kompetenz aufzubauen, zwischen Blackbox-Sicht und Whitebox-Sicht zu unterscheiden. Betrachtet man beispielsweise die Methode *Bewegen* (Whitebox-Sicht), ist diese in den Unterklassen sehr übersichtlich. Die aufgerufene Methode *EntfernenWennAußerhalb* muss dabei nicht näher betrachtet werden (Blackbox-Sicht).

Klasse SCHLAGBAUM	Klasse QUADRAT
<pre>/** * Lässt das Hindernis um 5 Einheiten * gehen */ @Override void Bewegen() { Gehen(5); EntfernenWennAußerhalb(); }</pre>	<pre>/** * Lässt das Hindernis um 20 * Einheiten gehen */ @Override void Bewegen() { Gehen(20); EntfernenWennAußerhalb(); }</pre>

Quelltext 1: Quelltexte der Klassen für die Aufgabe

Codierung – Binärsystem

Bei einer Show des Wahlkurses Zaubertricks sollen Zahlen per „Gedankenübertragung“ übermittelt werden. In Wirklichkeit aber ver-raten kleine Strahler in der Bühnenbeleuchtung dem „Zauberer“ die jeweiligen Werte. Das System ist in der Vorlage dargestellt: Durch Klicken auf die einzelnen Leuchten kann man sie an- bzw. ausschalten. Die dadurch heimlich übermittelte Zahl wird darunter angezeigt.



13

a. Finden Sie heraus, nach welchem System die Zahlen dargestellt werden und wie viele Werte mit den vier Leuchten insgesamt beschrieben werden können. Geben Sie die Darstellung der Zahl 10 an.

b. Folgern Sie, wie die Zahlen 32 und 47 nach diesem System dargestellt werden müssten, und geben Sie beide Darstellungen an. (Hinweis: Der Zahlenbereich wird erweitert, es sind mehr als vier Lampen nötig.)

c. Für Schnelle: Die Leuchtmittel werden durch LEDs ausgetauscht, die neben gelb auch noch in rot leuchten können. Erschließen Sie sich das veränderte System und geben Sie eine Darstellung der Zahlen 17 und 64 an. Wie viele Werte könnte man mit den vier vorhandenen Leuchten nun insgesamt darstellen?

(Übernahme von Brichzin/Janus/Jetzinger/Neumeyer/Reinold/Seegerer/Wiedemann 2023, S. 78 mit geringer Adaption)

Vorlage zur Aufgabe siehe <https://kurzlinks.de/primm3>

Das letzte Beispiel soll zeigen, dass Intensionen von PRIMM, insbesondere die kognitive Aktivierung, auch bei Lerngerüsten ohne Quelltexte für den (Informatik-)Unterricht nutzbar sind: Teilaufgabe a bringt die Lernenden sofort in eine aktive, explorierende Rolle (*Run/Investigate*), denn sie müssen sich einen neuen Inhalt (hier das Binärsystem) erschließen. Bewusst wurde hier ein Kontext gewählt, der zumindest auf den ersten Blick nichts mit Mathematik zu tun hat, um eventuell vorhandene Barrieren zu umgehen. Auf den Erkenntnissen aus Teilaufgabe a aufbauend muss zur Darstellung der Zahlen 32 und 47 das System erweitert werden (*Modify*). Bei beiden Teilaufgaben empfiehlt es sich, von den Schüler*innen eine schriftliche Zusammenfassung einzufordern, die Basis für eine gemeinsame Besprechung und Ergebnissicherung im Plenum ist. Geht man regelmäßig so vor, kann dabei der Mehrwert von Skizzen mit Legenden gegenüber rein textlichen Zusammenfassungen deutlich gemacht werden. Da beim beschriebenen konstruktivistischen Prozess die Lerngeschwindigkeit recht unterschiedlich ist, ist die binnendifferenzierende Teilaufgabe c sehr wichtig. Diese bietet für die Leistungsstärkeren eine Herausforderung, indem über das

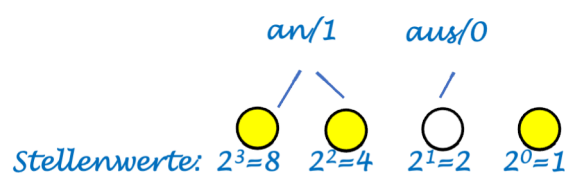


Abbildung 3: Beispielnotiz zur Aufgabe

Binärsystem hinaus auf das Ternärsystem erweitert wird; das Ergebnis der Bearbeitung kann an späterer Stelle im Unterricht wertschätzend eingebracht werden, beispielsweise wenn das Hexadezimalsystem angesprochen wird.

Fazit

Dieser Artikel zeigt Einsatzmöglichkeiten von PRIMM, auch über Programmieraufgaben hinaus, und dessen Potenzial in einem konsequent konstruktivistisch geführten Informatikunterricht. Der Ansatz unterstützt die Lehrperson in der Strukturierung von Lernaufgaben hin zu einem konsequent schüleraktivierenden Unterricht, der durch die Abfolge der Phasen den Lernenden der verschiedenen Leistungsniveaus passend entgegenkommt. Wie bei allen Unterrichtsmethoden ist auch PRIMM kein Selbstläufer bei einem vereinzelt Einsatz. Eigenverantwortliches Arbeiten und die Fähigkeit der Lernenden, zwischen Blackbox- und Whitebox-Sicht wechseln zu können, sind ebenso wichtige Voraussetzungen für einen erfolgreichen Einsatz, wie die passende Reduzierung der Aufgabenkomplexität durch die Lehrenden.

Material

Vorlagen zu den vorgestellten Beispielen (und vielen anderen Einstiegsaufgaben) können unter informatikschulbuch.de heruntergeladen werden. Das Material für die Beispiele 1, 3 und 4 ist unter informatikschulbuch.de/primm zusammengefasst.

Quellen

Alle Webseiten/Links wurden zuletzt geprüft am 16.06.2024.

Brichzin, Peter; Jetzinger, Franz; Neumeyer, Johannes; Reinold, Klaus; Wiedemann, Albert (2021): Informatik 3: Funktionale Modellierung, Datenmodellierung und objektorientierte Modellierung, Cornelsen Verlag, Berlin

Brichzin, Peter; Janus, Florian; Jetzinger, Franz; Neumeyer, Johannes; Reinold, Klaus; Seegerer, Stefan; Wiedemann, Albert (2023): Informatik Gymnasium Oberstufe 1: Algorithmen, Codierung, Kommunikation in Netzwerken, Künstliche Intelligenz, Cornelsen Verlag, Berlin

Coleman, Gemma (Hg.): The PRIMM Approach. In: Raspberry Pi Foundation (2021): The Big Book of Computing Pedagogy. <https://www.raspberrypi.org/hello-world/issues/the-big-book-of-computing-pedagogy>

Sentance, Sue; Waite, Jane (2017): PRIMM: Exploring pedagogical approaches for teaching text-based programming in school. in Proceedings of the 12th Workshop on Primary and Secondary Computing Education, 113 – 114. Nijmegen, The Netherlands: ACM.

Sentance, Sue; Waite, Jane; Kallia, Maria (2019): Teaching computer programming with PRIMM: a sociocultural perspective, Computer Science Education, DOI: 10.1080/08993408.2019.1608781

Stem Learning: Scratch cards: <https://www.stem.org.uk/resources/elibrary/resource/35110/scratch-cards>

Textor, Martin: Lew Wygotski – der ko-konstruktive Ansatz. In: Fthenakis, Wassilios E., Textor, Martin R. (Hg.): Pädagogische Ansätze im Kindergarten. Weinheim, Basel: Beltz 2000, S. 71-83

Lizenz



Dieser Artikel steht unter der Lizenz CC BY NC 4.0 zur Verfügung.

Kontakt

Klaus Reinold
Rupprecht-Gymnasium, München
E-Mail: reinold@rupprecht-gymnasium.de

Peter Brichzin
Erasmus-Grasser-Gymnasium, München
E-Mail: schule@brichzin.de