

ProgrammingWiki in der Praxis - ein Erfahrungsbericht

Michael Hielscher
Zentrum für Bildungsinformatik
Pädagogische Hochschule Bern
mail@michael-hielscher.de

Christian Wagenknecht
Fakultät Elektrotechnik und Informatik
Hochschule Zittau/Görlitz
c.wagenknecht@hs-zigr.de

Veit Berger
Fachlehrer für Physik, Mathematik und Informatik
Geschwister-Scholl-Gymnasium Löbau
v.berger@gmx.de

Thomas Grebedüinkel
Fachlehrer für Mathematik, Physik,
Astronomie und Informatik
Philipp-Melanchthon-Gymnasium Bautzen
thomas.grebeduenkel@schulen.sachsen.de

Abstract: ProgrammingWiki ist ein spezielles Wiki-System. Es soll den Anfangsunterricht im Programmieren unterstützen. Das System hat eine gute technische Reife erlangt und wurde in ausgewählten Veranstaltungen des Informatikstudiums didaktisch erprobt. Dieser Phase folgte nun eine Erprobung im Informatikunterricht zweier Gymnasien. Dabei verdichten sich Erfahrungen, über die im Folgenden berichtet wird. Zur Konkretisierung werden ausgewählte Themen mit zugehörigen Beispielen aus der Unterrichtspraxis einbezogen. Das umfangreiche Unterrichtsmaterial steht auf dem ProgrammingWiki unter <http://programmingwiki.de> frei zur Verfügung.

1 Einleitung

Die Nutzung von Web 2.0-Technologien für Lehr- und Lernzwecke, sowohl an Hochschulen als auch an Schulen, ist ein stetig wachsendes Thema für Lehrende in der Praxis wie auch für Forschende an Universitäten und Hochschulen (vgl. beispielsweise [Has11] und [Kal11]). Eine solche Web 2.0-Technologie ist das Wiki, welches für den Informatikunterricht vielfältig genutzt werden kann (vgl. [Hon07]). Wikis werden vorrangig als Medium und Werkzeug für die computerunterstützte Mensch-Mensch-Kommunikation verwendet, um verschiedenste Materialien kollaborativ zu erarbeiten und zu verteilen. Das ProgrammingWiki entstand aus dem von der Wikipedia her bekannten MediaWiki-System, indem ein bestimmter Satz von Software-Erweiterungen hinzugefügt wurde. Dadurch ist das ProgrammingWiki in der Lage, interaktive Programmier-Elemente zu verarbeiten (s. [HW09]): Eingebettete Programmtexte können unmittelbar editiert, gespeichert und interpretiert (ausgeführt) werden. Ein der ausgewählten Programmiersprache (z.B.:

Java, Pascal, SQL, XSLT, Scheme oder Prolog) entsprechender Interpreter arbeitet „unsichtbar“ im Hintergrund, um Programmtexte zu evaluieren. Die Ergebnisse dieses Prozesses werden im Arbeitsblatt an vorgesehener Stelle eingefügt. Dies geschieht innerhalb des Webbrowsers mit Hilfe eingebauter Java-Applets. Separate Interpreter-Installationen und deren Anpassungen auf jedem Clientrechner erübrigen sich. Die einzige Voraussetzung ist ein permanenter Web-Zugang eines Computers mit Java-fähigem Webbrowser. Der Webserver ist am Übersetzungs- und Ausführungsprozess von entwickelten Programmen unbeteiligt. Folglich entstehen keine Performance-Engpässe am Server, auch wenn viele Lernende gleichzeitig ihre erstellten Programme ausführen.

Das ProgrammingWiki wird seit Oktober 2008 im Informatikstudium an der Hochschule Zittau/Görlitz eingesetzt (vgl. [HW10]). Von November 2009 bis Februar 2011 wurde ein Pilotprojekt an zwei sächsischen Gymnasien durchgeführt. Darin wurden konkrete Themen des Informatikunterrichts in den Klassen 9, 11 und 12 für den ProgrammingWiki-Einsatz erschlossen und qualitative Einsatzbewertungen vorgenommen. In Bezug auf die Wiki-Nutzung war der Vorkenntnisstand sowohl der beteiligten Lehrenden als auch der Lernenden zunächst sehr gering. Die Auswertungsgespräche der Lehrpersonen fanden vierteljährlich statt. Für das Projekt wurde ein leeres ProgrammingWiki-System ohne Zugriffsschutz eingerichtet. Von den rund 100 registrierten Benutzern entstanden zusammen über 350 ProgrammingWiki-Seiten, auf die über 50000 Mal zugegriffen wurde.

Im Folgenden werden ausgewählte Beispiele aus dem Informatikunterricht skizziert und im Hinblick auf ihren Platz in einer ProgrammingWiki-basierten Didaktik kommentiert. Dabei ist zu beachten, dass das ProgrammingWiki keine kosmetische Ergänzung traditioneller didaktischer Ansätze darstellt, sondern als Form kooperativen computergestützten Unterrichts tief in den Lernprozess eingreift und kontinuierlicher Integration bedarf. Die zitierten Beispiele sind eher mathematiklastig, basierend auf einschlägiger Literatur (vgl. [ASS01]). Auch wenn dies der konkreten Unterrichtssituation entsprach, ist dies keineswegs ProgrammingWiki-typisch.

2 ProgrammingWiki im Unterricht

Für viele Schülerinnen und Schüler ist der tägliche Umgang mit einem Computer (zunehmend Notebook) ganz selbstverständlich. Zu den Hauptanwendungsfeldern gehören die Recherche im Internet, Onlinespiele, Mitgliedschaft in sozialen Netzwerken sowie die Kommunikation via Chat und Email. Medientechnisch sind die Schülerinnen und Schüler somit gut für den ProgrammingWiki-Einsatz vorgebildet. Die Arbeit in einem Wiki stellt dennoch neue und teilweise unerwartete Anforderungen an die Lernenden: Die Möglichkeiten des gemeinsamen Erarbeitens von Dokumenten, der „offene“ Charakter der Plattform (jeder kann rückverfolgbare Änderungen vornehmen) müssen in ihrer Tragweite zunächst erfasst und anschließend akzeptiert und verinnerlicht werden.

Deshalb beginnt der Unterricht nicht damit, das Potenzial des ProgrammingWiki-Systems auszuschöpfen, sondern vielmehr mit Aufgabenstellungen, die auf die Verwendung, Mo-

difikation und schließlich Erarbeitung „normaler“ Wiki-Dokumente abzielen. Auf diese Weise werden die Grundfunktionen des MediaWiki-Systems eingeübt.

In diesem Sinne wurden Wiki-Dokumente im Lernbereich „Sicherheit von Informationen“ (vgl. [Leh07]) als Zusammenfassungen behandelter Themen gestaltet: Die Schülerinnen und Schüler erhielten hierfür verschiedene Teilaufträge, um den kooperativen Weg zum Ergebnis zu erleben. Als Nebenprodukt machten sie sich mit der MediaWiki-Syntax (z.B. für Tabellen, Aufzählungen, Links usw.) vertraut.

Die früher verwendeten Werkzeuge (Delphi 7 zur Programmierung und Access für Datenbanken) wurden vollständig durch das ProgrammingWiki ersetzt und die Unterrichtsmaterialien entsprechend angepasst. Dabei ist zu beachten, dass die vom ProgrammingWiki bereitgestellten Interpreter nicht etwa im Sinne einer Notlösung, d.h. als mehr oder weniger gute Alternative zu den o.g. Systemen, verstanden werden. Vielmehr definiert das ProgrammingWiki eine reglementierte Arbeitsumgebung, in der konkrete Software mit ihren spezifischen Eigenschaften, wie Evaluationsreihenfolge und Gültigkeitsbereiche im Dokument, vorgehalten wird.

Analog unterscheidet sich eine ProgrammingWiki-Systemumgebung von einer Entwicklungsumgebung (IDE), wie sie in der professionellen Programmierung verwendet wird. Besonders im Bereich „Algorithmen“ zeigte sich die Stärke des vereinfachten Editors im ProgrammingWiki: Der Fokus konnte klar auf den jeweiligen Lerninhalt gelegt werden, ohne Rahmenbedingungen, wie Hauptprogramme, GUI-Elemente oder Projekteinstellungen, berücksichtigen zu müssen. Ähnliches gilt für die Modellierung und Nutzung von Datenbanken. Über das Lehrerinterface im ProgrammingWiki konnte die Schülerarbeit beobachtet und durch Kontrolle, Korrektur und Hilfestellungen beeinflusst werden. Da ein entwickeltes Programm jederzeit und von jedem Ort (von jedem Computer) ausgeführt und angepasst werden kann, sind Hilfestellungen nicht an das Arbeitsgerät des Lernenden gebunden: Ebenso wie jeder Schülerrechner kann ein am Lehrercomputer angeschlossener Beamer jede beliebige ausgewählte Schüler-Lösung anzeigen, ohne dass irgendein Umbau, wie Laptop- oder Stick-Transport, notwendig ist.

Einen Schwerpunkt des Lernbereiches „Algorithmen und Programmierung“ bildet die Behandlung von Iteration und Rekursion. Neben der Berechnung der Fakultät wird ein Programm für das Potenzieren erarbeitet. Abb. 1 zeigt links die Vorgabe des Lehrenden; rechts ist die von einem Schüler durch Ausfüllen der Codebox bearbeitete Seite dargestellt. Im ProgrammingWiki ist eine Schülerlösung an dem roten Kreuz in der linken oberen Ecke der betrachteten Eingabebox zu erkennen. Es dient außerdem als Schaltfläche zum Zurücksetzen der Schülerlösung auf die Vorgaben der Lehrperson.

Durch die Einbettung interaktiver Elemente in die Aufgabenstellungen bzw. Lehrtexte können und sollen Schülerinnen und Schüler an den passenden Stellen der Wissensvermittlung Computertexte durchführen. Die Unterschiede iterativer und rekursiver Programme werden im Direktvergleich entsprechender Codeabschnitte gut sichtbar. Gezielte Modifikationen des jeweiligen Programmtextes führen bei dessen Interpretation zu Konsequenzen, die vom Lernenden unmittelbar erlebt und bewertet werden können.

Nach dem Erarbeiten der ersten Beispiele wurden analoge Übungen (Summe aufeinander folgender Zahlen, Fibonacci-Zahlen usw.) von den Lernenden selbstständig durch Adap-



Abbildung 1: Darstellung der Thematik im ProgrammingWiki

tion der Einführungsbeispiele bearbeitet. Die automatische Speicherung der entwickelten Lösungen im ProgrammingWiki ermöglicht es, die Arbeit an einer beliebigen Stelle zu unterbrechen und zu einem späteren Zeitpunkt und/oder Ort fortzusetzen. Der Aufwand zur Herstellung eines interaktiven Arbeitsblattes entspricht in etwa dem für ein Arbeitsblatt auf Papier.

Für den Schwerpunkt „Modellierung von Datenbanken“, einschl. Umsetzung mit SQL, stellt das ProgrammingWiki-System jedem Schüler (jedem Webbrowser) eine eigene Datenbank zur Verfügung. Die ansonsten notwendige Installation eines SQL-Servers, wie MySQL, erübrigt sich, wodurch der Administrationsaufwand für die Lehrperson deutlich abnimmt. SQL-Abfragen und Anweisungen lassen sich in der üblichen Form angeben (s. Abb. 2).

Im Ergebnis der Erstellung eines Entity-Relationship-Diagrams (kurz: ERD) wurden von den Schülerinnen und Schülern verschiedene Lösungen angeboten. Es lohnt sich sehr, diese Lösungen im Unterricht vorzustellen und zu kommentieren. Das ProgrammingWiki bietet hierfür eine hervorragende Unterstützung, indem die entwickelten ERD-Bilder individuell hochgeladen und in Tabellenform zusammengefasst werden können. Auf diese einfache Weise können die Lösungsvorschläge der gesamten Klasse zugänglich gemacht werden (s. Abb. 3).

Insbesondere Projektaufgaben erfordern oftmals den Entwurf eines Datenmodells. Projektorientierte Arbeitskontexte, die in der Dimension angemessen sind, erweisen sich als sehr

```

1 DROP TABLE IF EXISTS leser;
2 CREATE TABLE leser (
3   L_Nr varchar(255),
4   Name varchar(255),
5   Vorname varchar(255),
6   Geb_datum varchar(30),
7   PLZ varchar(255),
8   Ort varchar(255),
9   Straße varchar(255)
10 );
11
12
13 INSERT INTO leser VALUES ('001', 'Franz', 'Werner', '12.08.1969', '01259', 'Dresden', 'Ottost.12');
14 INSERT INTO leser VALUES ('002', 'Bert', 'Kuni', '01.01.2010', '02625', 'Bautzen', 'Bismarkstr. 10');
15 INSERT INTO leser VALUES ('003', 'Hummel', 'Elfriede', '05.12.1940', '02222', 'Doosdorf', 'Nr. 5');
16 INSERT INTO leser VALUES ('004', 'Stramm', 'Berta', '01.02.1988', '01226', 'Dresden', 'Heideweg 7');
17 INSERT INTO leser VALUES ('010', 'Bunt', 'Sebastian', '11.11.1985', '02625', 'Bautzen', 'Reichenstr. 17');
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

```

1 SELECT buch2.Schlagwort, buch1.Titel, buch1.Verlag, buch1.Jahr
2 FROM buch1, buch2
3 WHERE buch1.T_Nr = buch2.T_Nr and ((buch2.Schlagwort)='LISP' );
4
5 select T_Nr, Titel, Verlag, Aus_datum
6 from leser, ausleihe, bestand, buch1
7 where leser.L_Nr=ausleihe.L_Nr and ausleihe.reg_nr=bestand.reg_nr and bestand.T_nr=buch1.T_Nr and L_Nr='001';
8

```

speichern & ausführen:

| SCHLAGWORT | TITEL | VERLAG | JAHR |
|------------|--|----------------|------|
| LISP | Struktur und Interpretation von Computerprogrammen | Springer | 1991 |
| LISP | Lehr- und Übungsbuch Informatik, Bd. 3 | Fachbuchverlag | 1997 |

2 rows in set.

Abbildung 2: Erstellen von Tabellen und Abfragen mit SQL

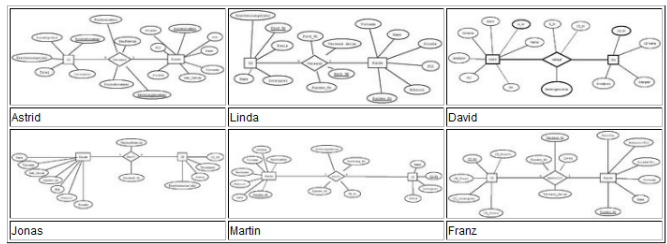


Abbildung 3: Lösungsansätze von Schülern als Thumbnails - Grundlage für Diskussionen

motivierend und führen bei Kleingruppenarbeit oft zu beachtlichen Resultaten. Dem gegenüber ist eine personenbezogene Bewertung (Wiki-ferner) individueller Beiträge meist sehr schwierig. Dieses Problem kann jedoch deutlich entschärft werden, wenn die Anfertigung eines Belegs im ProgrammingWiki erfolgt. Im Rahmen der hier zitierten Erprobung entstanden je Schülerin und Schüler eine bzw. mehrere ProgrammingWiki-Seiten und trotz gleicher Aufgabenstellung individuelle Belege mit unterschiedlichen Erfüllungs- und Schwierigkeitsgraden.

Anspruchsvollere Aufgaben, deren Lösung komplexere Programme erfordern, lassen die Grenzen der interaktiven ProgrammingWiki-Programmierung erkennen: Größere Programme erfordern spezielle Editoren und eine Arbeitsumgebung, wie sie nur moderne IDE bereitstellen. Für diese Aufgabendimension ist die ProgrammingWiki-Arbeit nicht vorgesehen.

3 ProgrammingWiki als interaktives Lehrbuch

Die o.g. Beispiele illustrieren, dass das ProgrammingWiki ein universelles und einfach zu handhabendes didaktisches Werkzeug ist. Es überzeugt durch seine multimedialen Eigenschaften (Einbindung von Text, Bild, Ton und Video), permanente Verfügbarkeit und vielfältige didaktisch-methodische Einsatzmöglichkeiten. In den vergangenen beiden Jahren wurde beispielsweise ein „ProgrammingWiki-Lehrbuch“ für den Informatikunterricht in Jahrgangsstufe 11 ansatzweise entwickelt. Auf der Basis funktionsorientierter Programmierung mit Scheme reichen dessen Inhalte von einfachen Einführungsbeispielen bis zu ausgewählte Algorithmen der Kryptologie, vgl. [Wag04] und [WB05]. Dabei werden leistungsfähige Programmierkonzepte, wie verschiedene Formen der Rekursion, Funktionen höherer Ordnung und Konzepte verzögerter Evaluation ebenso thematisiert, wie die Effizienz etablierter Such- und Sortierverfahren mit Blick auf die praktischen Grenzen des Problemlösens, vgl. [Wag03].

Programmtexte werden grundsätzlich in Code-Boxen für die Lernenden zum Experimentieren bereitgestellt. Umfangreichere Programmmodule, etwa für aufwendige Berechnungen oder die Erzeugung grafischer Darstellungen und Auswertungen, sind häufig als Inhalt versteckter Elemente definiert und so für die Schülerinnen und Schüler ausgeblendet. Dadurch reduziert sich das optische Ablenkungspotenzial vom eigentlichen Behandlungsgegenstand. Die Verwendung dieser Programme wird mittels Aufrufmuster (API) vermittelt und angeleitet.

Die Schülerinnen und Schüler implementieren eine Prozedur höherer Ordnung zur Berechnung der ersten Ableitung einstelliger Funktionen. Zur Visualisierung der Graphen wurde von der betreffenden Lehrperson ein einfacher Funktionsplotter (s. Abb. 4) als (verstecktes) Modul zur Verfügung gestellt und das Nutzungsmuster (Schnittstelle) innerhalb der dafür vorgesehenen Seite beschrieben. Mit dieser Anwendung lassen sich willkommene Bezüge zum laufenden Mathematikunterricht herstellen.

Im Allg. werden die ProgrammingWiki-Seiten des interaktiven Lehrbuches von themenbezogenen Aufgabenstellungen begleitet. Diese werden von Schülerinnen und Schülern, die am ProgrammingWiki angemeldet sind, bearbeitet und präsentiert. Hier wird ein Leistungsmerkmal des ProgrammingWikis besonders sichtbar: Mit vergleichbaren Aufgabenstellungen können vielfältige didaktische Funktionen realisiert werden. Unterrichtsbegleitende Schülerübungen und Hausaufgaben sind genauso denkbar, wie zu bewertende Leistungskontrollen.

Zur Verwaltung steht der Lehrperson das bereits erwähnte Lehrerinterface zur Verfügung. Es informiert nicht nur grob über den jeweiligen Erfüllungsstand, sondern ermöglicht auch den Blick auf jede individuelle Schülerseite (vgl. Abb. 5). Ausgewählte Seiten können gesperrt, d.h. einer Weiterbearbeitung entzogen werden.

Mit der Check-Box (mit Smiley) wird dem Lernenden eine gewisse Rückkopplung über den individuellen Erfolg beim Aufgabenlösen gegeben. An vorbereiteten Testaufgaben wird überprüft, ob das bearbeitete Programm erwartungsgemäß arbeitet. Didaktisch darf eine Check-Box nicht überbewertet werden, denn die Ja/Nein-Antwort bietet keinerlei Lösungshilfe an. Die Auswahl von Testbeispielen muss sorgsam erfolgen und beispielsweise

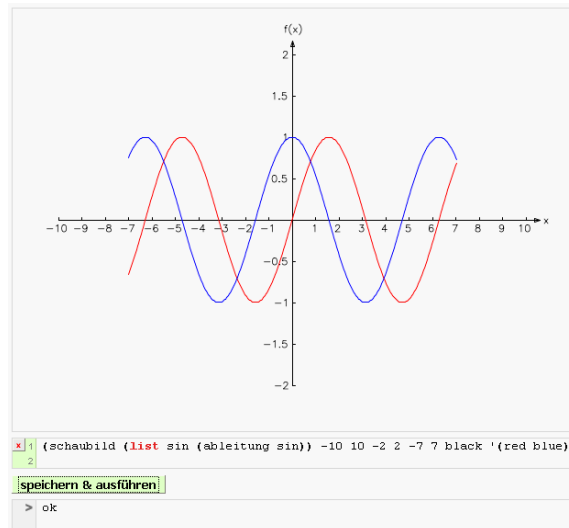


Abbildung 4: Schaubild von Funktionen und ihren Ableitungen










| Administration Lehrperson: | | |
|---|---|---|
|  |  Michael Schneider | Status:  |
|  |  Patrick Bürgermeister | Status:  |
|  |  Alex | Status:  |

Abbildung 5: Lehrerinterface mit dem Erfüllungsgrad eine ProgrammingWiki-Aufgabenseite

numerische Effekte beachten. Eventuelle Fehlbewertungen ziehen oft unberechtigte Misserfolgsenerlebnisse nach sich. In ausgewählten Fällen kann die Check-Box auch zur Kontrolle von Eingaben, Vorbedingungen usw. herangezogen werden.

```
x| 1 (define p 97)
  2 (define q 103)
  3 (define e 107)
  4 (define d 3203)
  5
```

 Prima, deine Lösung scheint zu stimmen.

jetzt prüfen & speichern!

Abbildung 6: Prüfen der Parameter zur RSA-Verschlüsselung mit einer Check-Box

Zur Demonstration des RSA-Algorithmus soll geprüft werden, ob die Werte p und q Primzahlen sind und die Variablen e und d die Bedingungen für den privaten bzw. öffentlichen Schlüssel erfüllen. Diesen vielschichtigen Test übernimmt eine wie in Abb. 6 dargestellte Check-Box.

4 ProgrammingWiki zur Binnendifferenzierung

Die Aufgabenstellungen im ProgrammingWiki werden üblicherweise mit aufsteigendem Schwierigkeitsgrad angeordnet. Darüber hinaus bietet das ProgrammingWiki verschiedene Möglichkeiten zur Ausprägung von Binnendifferenzierung. Wie in Abb. 7 gezeigt, wurden von den Lehrpersonen auf vielen ProgrammingWiki-Seiten unter der Rubrik „Zum Weiterarbeiten“ weiterführende, komplexe Aufgabenstellungen angeboten, zu denen bereits separate Lösungsseiten angelegt wurden.

Zum Weiterarbeiten

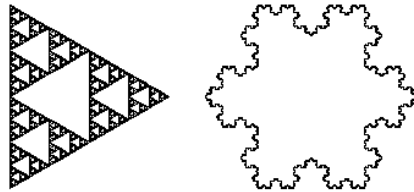
[\[Bearbeiten\]](#)

Fraktale

[\[Bearbeiten\]](#)

Mit selbstähnlichen Figuren bzw. Objekten beschäftigt sich heute ein ganzes Teilgebiet der Mathematik, die *fraktale Geometrie*. Ihre Untersuchungsergebnisse reichen in die Funktions- und Berechnbarkeitstheorie hinein und haben einen wesentlichen Einfluss auf die *Chaosforschung* komplexer dynamischer Systeme.

Klassische selbstähnliche oder fraktale Figuren sind das Sierpinski-Dreieck und die Koch-Schneeflocke:



Beschreiben Sie beide [Fraktale](#) verbal und implementieren Sie diese mit der Turtle-Grafik.

[Zur Problemlösung.](#)

Abbildung 7: Binnendifferenzierung durch weiterführende Aufgaben

Mit dem Konzept der verzögerten Evaluation lassen sich ausgewählte Zahlenfolgen als potenziell unendliche mathematische Objekte implementieren. Vom Lehrenden sind hierfür Sprachelemente für streams bereitzustellen. Besonders gute Teillösungen wurden von den Schülerinnen und Schülern auf separaten Lösungsseiten veröffentlicht.

Bei Untersuchungen zur Zeitkomplexität ausgewählter Algorithmen tritt die Programmierfähigkeit der Lernenden in den Hintergrund. Doch auch dann sind vielfältige Ansatzpunkte zur Binnendifferenzierung gegeben. Mit der Bereitstellung leistungsfähiger Programmmodule besteht beispielsweise die Möglichkeit, Bausteine für empirische Untersuchungsstrategien zu entwickeln und anzuwenden.

In Abb. 8 ist der mittlere Zeitaufwand bei der binären Suche einer Zahl in verschiedenen Intervallen dargestellt. Leistungsstarke Schülerinnen und Schüler können bereits hier einen logarithmischen Zusammenhang vermuten, der beispielsweise mit einem Tabellenkalkulationsprogramm verifiziert werden kann.

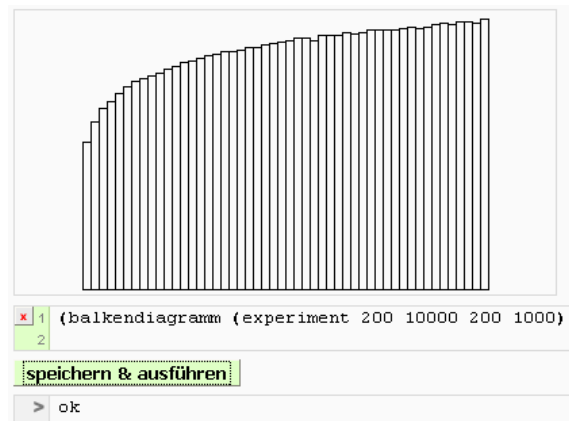


Abbildung 8: Empirische Effizienzuntersuchungen zur binären Suche

5 Zusammenfassung und Ausblick

Besonders zu Beginn des Unterrichts mit dem ProgrammingWiki zeigten die Schülerinnen und Schüler viel Freude beim Ändern von Seiten. Die dafür notwendige Kenntnis der Wiki-Syntax stellte sich zu keinem Zeitpunkt als Problem dar. Die anfänglich starke Motivation ging nach den ersten Wochen auf Normalmaß zurück und das ProgrammingWiki wurde mehr und mehr als reines Werkzeug, gern auch außerhalb des Unterrichts, wahrgenommen und genutzt. Die Möglichkeit, Aufgabentexte des Lehrenden durch Lernende zu modifizieren, führte zu einem völlig neuen Erlebnis der Lehrer-Schüler-Interaktion. Über den gesamten Versuchszeitraum gab es dennoch keinen einzigen Fall von Vandalismus.

Das ProgrammingWiki eignet sich besonders für Einstiegsbeispiele in der jeweils verwendeten Programmiersprache. Ein großer Vorteil bei der Arbeit mit Pascal oder Java ist das Fehlen einer expliziten Hauptprogrammbindung bzw. Hauptklassenbindung, wodurch sofort modulares Programmieren mit Funktionen und Prozeduren thematisiert werden kann. Das ProgrammingWiki-Konzept eignet sich daher besonders für funktionsorientiertes Programmieren. Für die Verarbeitung von Ein-/Ausgabe-Operationen zur Laufzeit von Programmen wurden spezielle Pop-up-Formulare entwickelt.

Programmieraufgaben wurden im Anspruchsniveau gestaffelt: Zunächst wurden Beispiele oder Codefragmente vorgegeben, die von den Schülerinnen und Schülern angewandt, verstanden bzw. ergänzt oder korrigiert werden sollten. Die Checkboxes haben sich als ein wertvolles Mittel der Selbstreflexion bei Standardinhalten und in Festigungsphasen für

die Lernenden erwiesen. Das ProgrammingWiki bietet vielfältige didaktische Ansätze zur Binnendifferenzierung. Die Bandbreite dafür beginnt bei Anwendungsaufgaben mit differenziertem Anforderungsniveau, reicht über attraktive, multimediale Informationsseiten bis hin zur komplexen, im ProgrammingWiki zu repräsentierenden Belegarbeit.

Die an das Lehrmaterial gestellten Design-Anforderungen entsprechen denen aktueller Lehrbücher und sind grundsätzlich erfüllbar. Durch die Möglichkeit, z.B. mathematische Ausdrücke im ProgrammingWiki mit \LaTeX zu schreiben, erzielt man eine ansprechende Darstellungsqualität.

Kooperative Entwicklungen einer ProgrammingWiki-Seite von mehreren Schülern bzw. Schülergruppen können das didaktische Einsatzspektrum bereichern. Der in der Erprobung rasch aufgetretene Wunsch nach „Sammelseiten“ für wiederverwendbaren Code, führte zum Einbau einer Importfunktion für Programmbausteine von einer auf eine andere ProgrammingWiki-Seite. Damit lässt sich das Konzept der Modularisierung im Zusammenhang mit der Schnittstellenbeschreibung anschaulich im Unterricht thematisieren.

Im Rahmen einer Fortbildungsveranstaltung für Lehrpersonen (Sächsisches Bildungsinstitut) wurde im Januar 2011 das ProgrammingWiki als modernes Unterrichtswerkzeug mit guter Resonanz vorgestellt.

Literaturverzeichnis

- [ASS01] Harold Abelson, Gerald J. Sussman und J. Sussman. Struktur und Interpretation von Computerprogrammen: Eine Informatik-Einführung (Springer-Lehrbuch). Springer, 2001.
- [Has11] Joachim P. Hasebrook. Online-Lernen in Banken. Online-Lernen – Handbuch für Wissenschaft und Praxis, Hrsg. Paul Klimsa und Ludwig J. Issing:457–469, 2011.
- [Hon07] Beat Döbeli Honegger. Wiki und die fundamentalen Ideen der Informatik. Didaktik der Informatik in Theorie und Praxis, 12. GI-Fachtagung Informatik und Schule. INFOS’07:207–216, 2007.
- [HW09] Michael Hielscher und Christian Wagenknecht. Programming-Wiki: Online programmieren und kommentieren. Zukunft braucht Herkunft. 13. GI-Fachtagung Informatik und Schule, INFOS’09:281–292, 2009.
- [HW10] Michael Hielscher und Christian Wagenknecht. Dokumentation von Projektarbeiten mit dynamischen Inhalten mittels Web 2.0-Werkzeugen. Proceedings, eLBA Science Conference:225–235, 2010.
- [Kal11] Jesko Kaltenbaek. Hochschule online – Online Lehren und Lernen in der Hochschule. Online-Lernen – Handbuch für Wissenschaft und Praxis, Hrsg. Paul Klimsa und Ludwig J. Issing:367–388, 2011.
- [Leh07] Sächsischer Lehrplan Informatik Gymnasium Klassenstufe 11, 2007.
- [Wag03] Christian Wagenknecht. Algorithmen und Komplexität. Informatik interaktiv. Hanser, 2003.
- [Wag04] Christian Wagenknecht. Programmierparadigmen: Eine Einführung auf der Grundlage von Scheme. Teubner, 2004.
- [WB05] Christian Wagenknecht und Veit Berger. Programmierparadigmen mit Scheme. Unterrichtskonzepte für informatische Bildung, INFOS’05:219–229, 2005.