



Masterarbeit

# Implementierung einer Webanwendung zur Visualisierung von Programmiergrundlagen im Informatikunterricht

Willi Zobel

Matr.-Nr.: 3662805

Studiengang: Medieninformatik

Betreut durch:

Prof. Dr. paed. habil. Steffen Friedrich

Dr. rer. nat. Holger Rohland

Eingereicht am 02. August 2016

## **Kurzfassung**

Einfache Algorithmen bilden die Grundlage für das Programmieren. Obwohl die meisten Funktionen und Abfolgen logisch und trivial sind, bereiten sie vielen Anfängern Schwierigkeiten. Um diese zu beheben, ist 2008 im Rahmen einer Diplomarbeit ein Werkzeug zur Visualisierung solcher grundlegenden Programmstrukturen geschaffen worden, welches in dieser Arbeit technisch und inhaltlich überarbeitet wurde.

Zu Beginn der Arbeit wurde eine Untersuchung des aktuellen Lehrplans für den Informatikunterricht durchgeführt. Dabei wurden die Anforderungen an das zu erstellende Werkzeug erfasst. Im Anschluss wurden aktuell existierende Werkzeuge getestet und evaluiert, die dem Zweck dienen, Grundlagen von Programmierkenntnissen mithilfe von Visualisierungen zu vermitteln. Dadurch konnten die Anforderungen weiter spezifiziert werden.

Im Hauptteil der Arbeit wurde das zu überarbeitende Werkzeug analysiert und mithilfe aktueller Webtechnologien in eine Webanwendung transformiert. Anschließend wurden zusätzliche Module implementiert, die Objekte und verkettete Listen für den Nutzer verständlich visualisieren. Am Ende der Arbeit erfolgte eine Evaluation des Werkzeugs.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>9</b>
1.1	Ziel der Arbeit . . . . .	10
1.2	Vorgehensweise . . . . .	10
<b>2</b>	<b>Untersuchung des Informatikunterrichts</b>	<b>12</b>
2.1	Aktuelle Situation . . . . .	12
2.2	Empfohlene Lerninhalte . . . . .	14
2.3	Lehrmaterialien . . . . .	18
2.4	Visualisierungen . . . . .	19
<b>3</b>	<b>Analyse von Werkzeugen</b>	<b>20</b>
3.1	Aktuelle Werkzeuge . . . . .	20
3.1.1	DIN EN ISO 9241 - Teil 110 . . . . .	20
3.1.2	Aufgabenspezifische Bewertungskriterien . . . . .	23
3.1.3	Test der Werkzeuge . . . . .	23
3.1.4	Fazit . . . . .	38
3.2	Werkzeugvorlage . . . . .	40

3.2.1	Aufgabenspezifische Bewertungskriterien . . . . .	40
3.2.2	Aufgabenangemessenheit . . . . .	41
3.2.3	Selbstbeschreibungsfähigkeit . . . . .	42
3.2.4	Lernförderlichkeit . . . . .	42
3.2.5	Steuerbarkeit . . . . .	43
3.2.6	Erwartungskonformität . . . . .	43
3.2.7	Individualisierbarkeit . . . . .	43
3.2.8	Fehlertoleranz . . . . .	44
3.2.9	Evaluationsergebnisse . . . . .	44
3.2.10	Erweiterungen . . . . .	45
3.2.11	Fazit . . . . .	46
<b>4</b>	<b>Implementierung</b>	<b>48</b>
4.1	Entwurf . . . . .	48
4.1.1	Anforderungen . . . . .	48
4.1.2	Szenarien . . . . .	51
4.1.3	Aufbau der Anwendung . . . . .	59
4.2	Prototyp . . . . .	60
4.2.1	Umsetzung der Anforderungen . . . . .	61
4.2.2	Technische Realisierung . . . . .	62
<b>5</b>	<b>Evaluation</b>	<b>63</b>
5.1	Auswertung . . . . .	64
5.1.1	Erscheinungsbild . . . . .	65
5.1.2	Funktionen . . . . .	66
5.1.3	Module . . . . .	67
5.1.4	Allgemein . . . . .	69
5.2	Bewertung der Ergebnisse . . . . .	70

<b>6 Zusammenfassung und Ausblick</b>	<b>71</b>
<b>Anhang</b>	<b>75</b>
<b>Literaturverzeichnis</b>	<b>75</b>
<b>Verzeichnis der Webadressen</b>	<b>77</b>
<b>A Bewertung aktueller Werkzeuge</b>	<b>79</b>
A.1 Codecademy . . . . .	79
A.2 Code.org . . . . .	81
A.3 Learn X . . . . .	83
A.4 Coding Dojo . . . . .	85
A.5 TheCodePlayer . . . . .	87
A.6 Khanacademy . . . . .	89
A.7 Udacity . . . . .	91
A.8 Learneroo . . . . .	92
<b>B Bewertung der Werkzeugvorlage</b>	<b>95</b>
<b>C Evaluation</b>	<b>98</b>
C.1 Fragebogen . . . . .	98
C.2 Ergebnisse . . . . .	106
<b>D Dokumentation</b>	<b>114</b>
D.1 Ordnerstruktur . . . . .	114
D.2 Modul hinzufügen . . . . .	117
D.2.1 Benötigte Dateien duplizieren . . . . .	117
D.2.2 HTML-Datei anpassen . . . . .	117
D.2.3 JS-Datei anpassen . . . . .	118

D.2.4	Link in Navigation hinzufügen . . . . .	120
D.3	Bibliotheken . . . . .	120

# Abbildungsverzeichnis

3.1	Codecademy - Ansicht einer Aufgabe / Lesson . . . . .	25
3.2	Code - Ansicht einer Aufgabe . . . . .	28
3.3	Learn X - Ansicht eines Tutorials (Learn JS [31]) . . . . .	30
3.4	Coding Dojo - Algorithm Platform - Ansicht einer Aufgabe . . . . .	31
3.5	TheCodePlayer - Ansicht eines Walkthroughs . . . . .	33
3.6	Khan Academy - Ansicht einer Challenge . . . . .	35
3.7	Udacity - Ansicht eines Kurses . . . . .	36
3.8	Learneroo - Ansicht einer Challenge in einem Kurs . . . . .	38
3.9	Werkzeugvorlage - Ansicht eines Moduls . . . . .	41
4.1	Entwurf - Navigation / Menü . . . . .	59
4.2	Entwurf - Ansicht eines Beispielmoduls . . . . .	60
4.3	EduCode - Ansicht des Moduls <b>Objekte</b> mit Auswahlfeld . . . . .	61
4.4	EduCode - Ansicht des Moduls <b>Operatoren</b> mit Tutorial und ausgeblendeter Einleitung und Aufgabenstellung . . . . .	62
5.1	Bewertung des Erscheinungsbilds . . . . .	64
5.2	Bewertung der Funktionen . . . . .	65

5.3	Gewichtung der Funktionen . . . . .	66
5.4	Bewertung der Modulmetaphern . . . . .	68



# Tabellenverzeichnis

2.1	Grundsätze für den Informatikunterricht [9]	15
3.1	Kurse bei Codecademy	24
3.2	Codecademy: Vor- und Nachteile	26
3.3	Kurse bei Code.org	27
3.4	Code: Vor- und Nachteile	28
3.5	Beispielhafte Lerninhalte in den Tutorials bei Learn X	29
3.6	Learn X: Vor- und Nachteile	30
3.7	Coding Dojo: Vor- und Nachteile	32
3.8	TheCodePlayer: Vor- und Nachteile	34
3.9	Khan Academy: Vor- und Nachteile	35
3.10	Udacity: Vor- und Nachteile	37
3.11	Learneroo: Vor- und Nachteile	38
3.12	Bewertungen aktueller Werkzeuge	39
3.13	Bewertung der Werkzeugvorlage [11]	47
A.1	Bewertungen - Codecademy [20]	81

A.2	Bewertungen - Code.org [18]	83
A.3	Bewertungen - Learn X [28, 29, 31, 30, 32, 33, 34]	85
A.4	Bewertungen - Coding Dojo [22]	87
A.5	Bewertungen - TheCodePlayer [37]	88
A.6	Bewertungen - Khanacademy [27]	90
A.7	Bewertungen - Udacity [38]	92
A.8	Bewertungen - Learneroo [35]	94
B.1	Bewertungen - Werkzeugvorlage [11]	97

# 1 Einleitung

Algorithmen haben in der Informatik eine große Bedeutung. Durch festgelegte Schritte können Probleme dadurch strukturiert gelöst werden. Dies ist bei der Programmierung unabdingbar. Jedoch spielen Algorithmen auch in unserem Alltag eine wichtige Rolle. Bei der Bedienung eines Kaffeekochers muss klar definiert werden, in welcher Reihenfolge bestimmte Schritte ausgeführt werden müssen, damit am Ende ein guter Kaffee entsteht. Diese Algorithmen so abzubilden, dass auch Maschinen sie verstehen, bereitet vor allem Anfängern große Schwierigkeiten. Im Informatikunterricht sollte der Vermittlung dieses Themengebiets daher viel Zeit eingeräumt werden.

Das Verständnis von Algorithmen ist eine wichtige Voraussetzung für das Erlernen einer Programmiersprache. Weiterhin müssen, wie auch bei einer Fremdsprache, Vokabular und dessen Syntax-Regeln gelernt werden. Die Schwierigkeit bei Programmiersprachen liegt darin, dass schon kleine Fehler das gesamte Programm zum Absturz bringen können. Der Lernvorgang erfordert somit viel Zeit und eine verständliche und einfache Herangehensweise.

Wie bereits erwähnt, ist der Lernprozess von den Grundlagen der Programmierung sehr zeitintensiv. Bei Anfängern kommt es zu häufigen Verständnisproblemen. Daher gilt es, möglichst einfache und praxisnahe Beispiele zu wählen, um die Motivation und das Verständnis zu erhöhen. Ein weiteres sehr wichtiges Mittel sind Visualisierungen. Durch sie können den SchülerInnen schwer verständliche Algorithmen grafisch deutlich gemacht werden, was den Lernprozess verkürzt und intensiviert.

## 1.1 Ziel der Arbeit

Ziel dieser Arbeit ist die Erstellung einer Webanwendung, die unterrichtsbegleitend im Informatikunterricht eingesetzt werden kann und den SchülerInnen die Grundlagen der Programmierung mithilfe von Visualisierungen vermittelt. Als Vorlage dient dafür eine Flash-Anwendung, die im Jahr 2008 im Rahmen der Diplomarbeit von Claudia Schindler entstand [11]. Die Umsetzung als Webanwendung ist im Gegensatz zu einer Flash-Anwendung plattformunabhängig und benötigt keine Browser-Erweiterungen. Ein weiterer wichtiger Schritt ist die Überarbeitung der aktuell existierenden Module: Wertzuweisungen, Verzweigungen, Schleifen und Unterprogramme. Ein weiteres Ziel dieser Arbeit ist die Implementierung zusätzlicher Module bezüglich der Arbeit mit verketteten Listen.

Die Webanwendung soll SchülerInnen den Einstieg in die Programmierung erleichtern. Dies soll jedoch nicht nur in Form eines Textes geschehen. Die Motivation und der Spaß beim Lernen sind entscheidend für den Lernprozess. Eine interaktive Umgebung fördert die SchülerInnen und regt sie so zum eigenständigen Nachdenken an. Zudem können mittels Visualisierungen komplizierte Vorgänge auf eine verständliche Weise beschrieben werden. Diese beiden Funktionen, die Interaktivität und die Visualisierungen, sind ein wichtiger Bestandteil der entstehenden Webanwendung.

## 1.2 Vorgehensweise

Zu Beginn dieser Arbeit wird eine Untersuchung bezüglich des Informatikunterrichts durchgeführt. Dies erfolgt mithilfe des sächsischen Lehrplans und den Prüfungsanforderungen für Informatik. Durch diese wird in Erfahrung gebracht, welche Lerninhalte in welchen Klassenstufen vermittelt werden. Außerdem gibt die Untersuchung Aufschluss über die Zielgruppe der Webanwendung. Im Anschluss wird geklärt, welche Lerninhalte für den Informatikunterricht empfohlen werden. Dafür werden zunächst verschiedene Bewertungsmaßstäbe vorgestellt. Durch die Untersuchung verschiedener Studien und Veröffentlichungen werden die sinnvollen Lerninhalte anschließend bestimmt. Dabei spielen im Besonderen die Algorithmen eine wichtige Rolle. Des Weiteren stellt sich die Frage, auf welche Art und Weise die Lerninhalte vermittelt werden können. Dafür wird geklärt, welche Lehrmaterialien sich für den Informatikunterricht eignen und welche Ansprüche sie besitzen. Abschließend geht es um den Einsatz von Visualisierungen im Unterricht. Dabei werden die verschiedenen Typen von Visualisierungen miteinander verglichen.

Nach der Untersuchung des Informatikunterrichts folgt die Analyse aktueller Werkzeuge, die sich mit der Lehre von Programmiergrundlagen beschäftigen. Dafür werden vorerst Bewertungsgrundlagen festgelegt, die zum Teil auf der *DIN EN ISO 9241* beruhen. Anschließend erfolgt der Test verschiedener Werkzeuge anhand dieser Bewertungsrichtlinien. Der Abschnitt stellt die Konkurrenzanalyse in dieser Arbeit dar. Dadurch kann in Erfahrung gebracht werden, wie sich der aktuelle Markt in diesem Bereich in den letzten Jahren entwickelt hat. Zusätzlich werden dadurch neue Ideen gewonnen, die in die Webanwendung integriert werden können.

Anschließend erfolgt die Bewertung der Werkzeugvorlage, die im Rahmen der Diplomarbeit von Claudia Schindler entstand. Dabei werden die gleichen Bewertungsgrundlagen wie bei den aktuellen Werkzeugen verwendet. Dadurch ist auch hier ein direkter Vergleich möglich. Daraufhin werden die Evaluationsergebnisse, die während der Diplomarbeit entstanden, ausgewertet. Durch die dort aufgezeigten Probleme und Verbesserungsvorschläge kann eine differenziertere Implementierung der Webanwendung stattfinden.

Im Anschluss an die Analyse erfolgt die Implementierung. Diese teilt sich wiederum in die Abschnitte **Entwurf** und **Prototyp**. In der Entwurfsphase werden zuerst die Anforderungen an die Webanwendung gestellt und anschließend nach ihrer Priorität sortiert. Aufgrund dieser Anforderungen werden Szenarien für die verschiedenen Module erstellt. In den Szenarien wird festgelegt, was die Nutzer in diesem Modul erlernen, welche Aufgabe sie erledigen müssen und auf welche Weise die Visualisierung erfolgt. Auf der Basis dieser Szenarien wird im Anschluss der Aufbau der Anwendung definiert und die ersten Mockups erstellt. Im zweiten Teil der Implementierung wird erläutert, welche Anforderungen letztendlich im Prototyp umgesetzt worden sind und welche Änderungen es im Bezug zum Entwurf gab. Anschließend wird geklärt, welche Techniken für die Implementierung verwendet wurden und wie die Webanwendung aufgebaut ist.

Um die Qualität der Anwendung zu bewerten, erfolgt nach der Implementierung eine Evaluation. Dadurch können Fehler, Probleme und Verbesserungsvorschläge aufgezeigt werden. Um im Besonderen auch auf den Einsatz im Informatikunterricht einzugehen, wurden bei der Evaluation LehrerInnen und Studierende zu ihrer Meinung bezüglich des Tools befragt.

Abschließend werden die entstandenen Ergebnisse zusammengefasst und zukünftige Schritte aufgezeigt, die zur Verbesserung und Weiterentwicklung der Webanwendung beitragen.

## 6 Zusammenfassung und Ausblick

Das Ziel dieser Arbeit war die Implementierung einer Webanwendung, die SchülerInnen im Informatikunterricht dabei unterstützt, Programmiergrundlagen und Algorithmen mithilfe einfacher Beispiele und Visualisierungen kennenzulernen und zu verstehen. Vor der eigentlichen Implementierung wurde jedoch eine Untersuchung des Informatikunterrichts durchgeführt. Als erstes wurde dafür die aktuelle Situation analysiert. Seit 2007 gelten einheitliche Prüfungsanforderungen im Fach Informatik in allen deutschen Bundesländern. Diese geben Auskunft über das Fachwissen und die Kompetenzen, die von den SchülerInnen in der Abiturprüfung erwartet werden [4]. Weiterhin wurde auch der sächsische Lehrplan für das Fach Informatik am Gymnasium untersucht [7]. Dadurch konnte der Einsatz der Webanwendung zeitlich eingeordnet werden. Ab der 8. Klassenstufe beschäftigen sich die SchülerInnen intensiver mit dem Algorithmusbegriff. In der 9. Klasse erfolgt die formale Beschreibung von Algorithmen und der erste Kontakt mit Programmiersprachen. Nach der aktuellen Situation wurde die Frage untersucht, welche Inhalte im Informatikunterricht vermittelt werden sollten. Der Unterricht sollte sich nicht an kurzfristigen, sondern an konstanten und alltäglichen Entwicklungen orientieren [13]. In einer Untersuchung fanden Zender et al. heraus, dass das Konzept *Algorithmen* eine zentrale Rolle im Informatikunterricht einnehmen sollte. Dabei schnitt es in allen untersuchten Kriterien (Horizontalkriterium, Vertikalkriterium, Sinnkriterium und Zeitkriterium) sehr gut ab [13]. In den im Jahr 2008 veröffentlichten Bildungsstandards der *Gesellschaft für Informatik e.V.* wurden weitere Empfehlungen für den Informatikunterricht festgelegt [9]. Durch diese wurde die Idee der Entwicklung des Tools als Webanwendung gestützt. Durch die plattformunabhängige Verwendbarkeit wird die Chancengleichheit unter den SchülerInnen vergrößert, da die Anwendung auf fast jedem System ausgeführt werden kann. In den Bildungsstandards wurde außerdem eine Empfehlung bezüglich des Umgangs mit Algorithmen ausgesprochen. In den Klassenstufen 5 bis 8 sollten die SchülerInnen lernen, Algorithmen umgangssprachlich zu beschreiben. Ab der 8. Klasse werden Algorithmen mit formalen Sprachen beschrieben, so dass auch Maschinen diese verstehen können. An diesem Punkt kommt das Tool zum Einsatz. Im dritten Teil der Untersuchung des Informatikunterrichts ging es um das Lehrmaterial. Dabei geht es um die Fragestellung, auf welche Weise den SchülerInnen die Lehrinhalte vermittelt werden

sollen. In den letzten Jahren sind viele Lehrmaterialien entstanden, die SchülerInnen beim Lernprozess unterstützen können. Laut Diethelm et al. kommt es aufgrund einer fehlenden Lehrerperspektive häufig nicht zum Unterrichtseinsatz dieser Materialien [1]. Die LehrerInnen müssen zum Beispiel erfahren, wann sie ein Tool einsetzen können, welches Wissen dabei vermittelt wird und wie sie damit umzugehen haben. Im letzten Teil der Untersuchung des Informatikunterrichts wurde der Einsatz von Visualisierungen untersucht. Nils Faltin kam in seiner Dissertation zu dem Schluss, dass es nicht darum geht welche Visualisierungsform verwendet wird, sondern ob die Visualisierung einfach und verständlich ist. Jedoch können interaktive Visualisierungen den Lernprozess fördern, da die Motivation durch entdeckendes Lernen erhöht wird.

Im Anschluss an die Untersuchung des Informatikunterrichts ging es um die Bewertung aktueller Werkzeuge, die das Ziel verfolgen, Algorithmen und Programmiersprachen zu vermitteln. Die Bewertungsgrundlage setzt sich dabei zum einen aus dem Teil 110 der Norm EN ISO 9241 [6] und zum anderen aus der Einsatzmöglichkeit im Informatikunterricht zusammen. Beide Beurteilungen gingen zu 50% in die Gesamtbewertung ein. Damit geben die Ergebnisse nicht nur darüber Auskunft wie gut sie sind, sondern auch ob sie SchülerInnen in der 7.-9. Klassenstufe bei der Vermittlung von Programmiergrundlagen im Informatikunterricht unterstützen können. Die Testergebnisse zeigen, dass viele Tools Grundlagen der Programmierung voraussetzen. Der Quellcode wird oft nicht weiter erläutert. Eine interessante Funktion lieferte die Plattform *Code* [18], die mithilfe blockbasierter Programmierung Kindern ermöglicht, einfache Algorithmen spielerisch zu verstehen. Hervorzuheben ist weiterhin die Plattform *Khan Academy* [27], die sich mit abwechslungsreichen Modulen und Methoden die beste Bewertung sichert. Einige der getesteten Tools setzen auf Lernvideos. Diese sind für den Informatikunterricht jedoch meist ungeeignet.

Als Vorlage für die Webanwendung diente eine Flash-Anwendung, die im Rahmen der Diplomarbeit von Claudia Schindler entstand [11]. Im zweiten Teil der Analyse wurde diese Werkzeugvorlage bewertet. Dies erfolgte mit den gleichen Bewertungskriterien, wie für die aktuellen Werkzeuge. Am besten schnitt das Tool bei den aufgabenspezifischen Bewertungskriterien ab, da es speziell für den Einsatz im Informatikunterricht entwickelt wurde. Im nächsten Schritt wurde die Evaluation aus der Diplomarbeit von Frau Schindler ausgewertet. Dabei wurde festgestellt, dass der Quelltext zu Beginn sehr unverständlich ist. Außerdem wurde kritisiert, dass das Belegungsprotokoll und die Visualisierung nicht gleichzeitig angezeigt werden kann. Die Informationen über die aktuell existierenden Variablen sind jedoch für den Ablauf des Programms von großer Bedeutung. Weiterhin wurde festgestellt, dass die Metaphern in den Modulen zum Teil nicht relevant für den Alltag der SchülerInnen sind. Diese fördern jedoch ihre Motivation.

Im Anschluss an die Analyse wurden in der Entwurfsphase die Anforderungen an die Webanwendung definiert. Diese wurden anschließend in Muss-, Soll- und Kann-Kriterien unterteilt. Die Muss-Kriterien sind dabei so gewählt, dass die Anwendung lauffähig und einsatzbereit ist. Die Soll-Kriterien erweitern das Tool unter anderem um weitere Module und Steuerungsmöglichkeiten. Optionale Ideen wurden hingegen in den Kann-Kriterien festgehalten. Nach der Anforderungsanalyse wurden Szenarien und Inhalte für die einzelnen Module erstellt. Im Anschluss erfolgte die Erstellung von Mockups, die das Layout und das Design der Webanwendung darstellen. Der Prototyp wurde als Webanwendung mit den Sprachen HTML, CSS und Javascript implementiert. Durch den Verzicht auf serverseitige Techniken kann das Tool ohne Webserver im Browser aufgerufen

werden. Dies ermöglicht eine einfache Einbettung in E-Learning Plattformen als Lerninhalt.

Zum Schluss wurde eine Evaluation durchgeführt, bei der LehrerInnen, Studierende und auch SchülerInnen zu Ihrer Meinung bezüglich Educode befragt wurden. Dadurch konnte festgestellt werden, dass das Tool mit einer durchschnittlichen Bewertung von 1,83 einen guten Anklang findet und von den meisten auch im Unterricht eingesetzt werden würde. Zusätzlich wurden durch die Evaluation auch einige Verbesserungsvorschläge gesammelt, die zum Teil im Anschluss umgesetzt wurden.

Algorithmen zu verstehen und so zu formulieren, dass auch Maschinen sie richtig interpretieren, stellt SchülerInnen und LehrerInnen im Informatikunterricht auch in Zukunft vor eine große Aufgabe. Anwendungen wie Educode können die SchülerInnen bei diesem Schritt stark unterstützen. Im Besonderen sind Visualisierungen und alltagsnahe Beispiele dabei eine große Hilfe. Solche Tools sollten somit eine stärkere Einbindung in den Informatikunterricht erfahren. Dabei gibt es jedoch einige Dinge zu beachten. Die Anwendungen sollten sich nicht an kurz- sondern langfristigen Techniken orientieren. Außerdem ist es wichtig, dass auch SchülerInnen und LehrerInnen an der Entwicklung beteiligt werden, um das bestmögliche Ergebnis für alle zu erreichen. Vor allem die Lehrerperspektive wird oftmals nicht berücksichtigt. Die Tools werden nicht ausreichend beschrieben und auch der Einsatzzeitpunkt ist häufig unklar.

Das in dieser Arbeit entstandene Tool *Educode* kann zu diesem Zeitpunkt schon vollständig genutzt werden. Daher wird die Webanwendung voraussichtlich im August 2016 auf den sächsischen Bildungsserver geladen, um LehrerInnen und SchülerInnen aus Sachsen den Zugriff auf dieses Tool zu gewähren. Trotzdem gibt es noch einige Anpassungen, die Educode in Zukunft weiter ausbauen und verbessern können. Neue Module können implementiert werden, wie zum Beispiel für den Umgang mit Variablentypen. Aber auch das sortieren von verketteten Listen und die Einführung in Arrays wären zwei mögliche Module für die Modulgruppe *Datenstrukturen*. Dabei sollte jedoch darauf geachtet werden, dass das Tool derzeit für den Einstieg in die Programmiergrundlagen gedacht ist und die Module daher nicht zu komplex werden sollten. Da das dunkle Farbkonzept sehr ungewöhnlich ist, kann wahlweise eine hellere Variante angeboten werden. Auch die Anpassung der Schriftgröße ist eine gute Idee, um die Darstellung auf Bildschirmen mit unterschiedlichen Auflösungen zu verbessern. Zusätzlich können dem Benutzer noch mehr Freiheiten eingeräumt werden, in dem man ihm die Gestaltung der Oberfläche selbst überlässt. Der Nutzer kann so die benötigten Modulbereiche einblenden und an eine beliebige Position schieben. Dabei könnte die Anwendung auch besser an mobile Geräte angepasst werden.

Bei der Evaluation wurde von einigen Testpersonen die textlastige Einleitung kritisiert. Durch die Erstellung eines Popups, der die SchülerInnen zu Beginn eines Moduls Schritt für Schritt durch die Einleitung führt, könnte auf den Bereich *Einleitung* im Anschluss verzichtet werden. Außerdem können erfahrenere Personen diesen Teil überspringen. Ein wichtiger Punkt ist die Auswahl einer Programmiersprache im Editor, wodurch die LehrerInnen einen besseren Bezug für ihren persönlichen Unterricht herstellen können. Vor allem die Programmiersprachen *Java* und *Pascal* sind in diesem Zusammenhang zu nennen. Für fortgeschrittene SchülerInnen könnte der Editor weiterhin so angepasst werden, dass einzelne Abschnitte selbst eingetippt werden müssen. Um die Inhalte der Module noch verständlicher zu vermitteln, ist die Darstellung eines Flussdiagramms als



Alternative zur Visualisierung möglich. Neben den inhaltlichen Faktoren sollte jedoch auch die Motivation beachtet werden, die den Lernprozess der SchülerInnen maßgeblich verbessert. Dies kann zum Beispiel dadurch erreicht werden, dass abgeschlossene Module in der Navigation gekennzeichnet werden, so dass die SchülerInnen ihren Fortschritt besser nachvollziehen können. Gleichzeitig können dadurch auch die LehrerInnen den Lernfortschritt ihrer SchülerInnen überprüfen.

# Literaturverzeichnis

- [1] I. Diethelm, C. Dörge, A.-M. Mesaros, und M. Dünnebier (2011): Die Didaktische Rekonstruktion für den Informatikunterricht. In INFOS. Seiten 77–86. [https://www.researchgate.net/profile/Ira\\_Diethelm/publication/221208850\\_Die\\_Didaktische\\_Rekonstruktion\\_fr\\_den\\_Informatikunterricht/links/02e7e5258414c49aff000000.pdf](https://www.researchgate.net/profile/Ira_Diethelm/publication/221208850_Die_Didaktische_Rekonstruktion_fr_den_Informatikunterricht/links/02e7e5258414c49aff000000.pdf).
- [2] N. Faltin (2002): Strukturiertes aktives Lernen von Algorithmen mit interaktiven Visualisierungen. Ph.D. thesis, Universität Oldenburg. <http://oops.uni-oldenburg.de/269/1/299.pdf>.
- [3] S. Lehmann (2009): Visualisierung einfacher Algorithmen für den Informatikunterricht - Unterprogrammtechnik. Wissenschaftliche arbeit, Technische Universität Dresden, Fakultät Informatik, Inst. für Software- und Multimediatechnik, AG Didaktik der Informatik/Lehrerbildung.
- [4] o. V. (1989 i. d. F. vom 05.02.2004). Einheitliche Prüfungsanforderungen - Informatik. Kultusministerium. [http://www.kmk.org/fileadmin/Dateien/veroeffentlichungen\\_beschluesse/1989/1989\\_12\\_01-EPA-Informatik.pdf](http://www.kmk.org/fileadmin/Dateien/veroeffentlichungen_beschluesse/1989/1989_12_01-EPA-Informatik.pdf).
- [5] o. V. (2000). National Council of Teachers of Mathematics: Principles and Standards for School Mathematics. (NCTM). <http://standards.nctm.org/>.
- [6] o. V. (2008). Ergonomie der Mensch-System-Interaktion - EN ISO 9241-110. Deutsches Institut für Normung e.v. <http://xd-i.com/wp-content/uploads/2016/01/9241-110.pdf>.
- [7] o. V. (2011). Lehrplan Informatik - Gymnasium. Sächsisches Staatsministerium. [http://www.schule.sachsen.de/lpdb/web/downloads/lp\\_gy\\_informatik\\_2011.pdf](http://www.schule.sachsen.de/lpdb/web/downloads/lp_gy_informatik_2011.pdf).
- [8] o. V. (2014). Sachsen startet ins Klassenzimmer der Zukunft. Medienservice Sachsen. <http://www.medienservice.sachsen.de/medien/news/190949>.

- [9] H. Puhlmann, T. Brinda, M. Fothe, S. Friedrich, B. Koerber, G. Röhner, und C. Schulte (2008). Grundsätze und Standards für die Informatik in der Schule. [http://www.informatikstandards.de/docs/bildungsstandards\\_2008.pdf](http://www.informatikstandards.de/docs/bildungsstandards_2008.pdf).
- [10] G. Röhner, P. D. T. Brinda, V. Denke, D. L. Hellmig, T. Heußner, D. A. Pasternak, P. D. A. Schwill, und M. Seiffert (2016). Bildungsstandards Informatik für die Sekundarstufe II. [http://www.informatikstandards.de/docs/Bildungsstandards\\_SII.pdf](http://www.informatikstandards.de/docs/Bildungsstandards_SII.pdf).
- [11] C. Schindler (2008): Visualisierung einfacher Algorithmen für den Informatikunterricht. Diplomarbeit, Technische Universität Dresden, Fakultät Informatik, Inst. für Software- und Multimediatechnik, AG Didaktik der Informatik/Lehrerbildung.
- [12] H. Witten (2003): Allgemeinbildender Informatikunterricht? Ein neuer Blick auf HW Heymanns Aufgaben allgemeinbildender Schulen. In INFOS. Seiten 59–75. <http://cs.emis.de/LNI/Proceedings/Proceedings32/GI-Proceedings.32-7.pdf>.
- [13] A. Zandler und C. Spannagel (2006): Zentrale Konzepte im Informatikunterricht: eine empirische Grundlegung. *Notes on Educational Informatics - Section A: Concepts and Techniques*, Vol. 2 (Nr. 1):Seiten 1–21. [http://www.ph-ludwigsburg.de/fileadmin/subsites/2e-imix-t-01/user\\_files/Journal\\_NEI\\_-\\_PDFs\\_fuer\\_Webauftritt/Section\\_A/Volume\\_2\\_No\\_1\\_2006/NEI\\_Section\\_A\\_Vol.\\_2\\_No.\\_1\\_2006\\_p.\\_01-21\\_-\\_Zandler\\_Spannagel\\_-\\_Zentrale\\_Konzepte\\_01.pdf](http://www.ph-ludwigsburg.de/fileadmin/subsites/2e-imix-t-01/user_files/Journal_NEI_-_PDFs_fuer_Webauftritt/Section_A/Volume_2_No_1_2006/NEI_Section_A_Vol._2_No._1_2006_p._01-21_-_Zandler_Spannagel_-_Zentrale_Konzepte_01.pdf).

# Verzeichnis der Webadressen

- [14] Animate.css. <https://daneden.github.io/animate.css/>.
- [15] Bootstrap. <http://getbootstrap.com/>.
- [16] Bootstrap Slider. <http://www.eyecon.ro/bootstrap-slider>.
- [17] Bootstrap Tour. <http://bootstraptour.com/>.
- [18] Code. <https://code.org/>. (Zugriff: 2.3.2016).
- [19] Code Prettify. <https://github.com/google/code-prettify>.
- [20] Codecademy. <https://www.codecademy.com/>. (Zugriff: 2.3.2016).
- [21] Coding Dojo. <http://www.codingdojo.com/>. (Zugriff: 2.3.2016).
- [22] Coding Dojo Algorithm Platform. <http://algorithm.codingdojo.com/>. (Zugriff: 2.3.2016).
- [23] Font Awesome. <http://fontawesome.io/>.
- [24] jQuery. <https://jquery.com/>.
- [25] jQuery Custom Scrollbar. <http://manos.malihu.gr/jquery-custom-content-scroller/>.
- [26] jQuery Sortable. <https://johnny.github.io/jquery-sortable/>.
- [27] Khanacademy. <https://www.khanacademy.org/>. (Zugriff: 3.3.2016).
- [28] Learn C. <http://www.learn-c.org/>. (Zugriff: 2.3.2016).
- [29] Learn C-Sharp. <http://www.learncs.org/>. (Zugriff: 2.3.2016).

- [30] Learn Java. <http://www.learnjavaonline.org/>. (Zugriff: 2.3.2016).
- [31] Learn JavaScript. <http://www.learn-js.org/>. (Zugriff: 2.3.2016).
- [32] Learn PHP. <http://www.learn-php.org/>. (Zugriff: 2.3.2016).
- [33] Learn Python. <http://www.learnpython.org/>. (Zugriff: 2.3.2016).
- [34] Learn Shell. <http://www.learnshell.org/>. (Zugriff: 2.3.2016).
- [35] Learneroo. <https://www.learneroo.com/>. (Zugriff: 3.3.2016).
- [36] Raphael.js. <http://dmitrybaranovskiy.github.io/raphael/>.
- [37] TheCodePlayer. <http://thecodeplayer.com/>. (Zugriff: 3.3.2016).
- [38] Udacity. <https://www.udacity.com/>. (Zugriff: 3.3.2016).

## **Erklärung**

Hiermit erkläre ich, Herr Willi Zobel, die vorliegende Masterarbeit zum Thema

### **Implementierung einer Webanwendung zur Visualisierung von Programmiergrundlagen im Informatikunterricht**

selbständig und ausschließlich unter Verwendung der im Quellenverzeichnis aufgeführten Literatur- und sonstigen Informationsquellen verfasst zu haben.

Dresden, 02. August 2016