

Entwicklung und Konzeption eines E-Learning Kurses für Programmieranfänger

Diplomarbeit

zur Erlangung des Grades eines Diplom-Informatikers
im Studiengang Informatik

vorgelegt von
Kasjen Kramer

Erstgutachter: Prof. Dr.-Ing. Stefan Müller
(Institut für Computervisualistik, AG Computergraphik)

Zweitgutachter: Bastian Kraye, MSc.
Institut für Computervisualistik

Koblenz, im September 2017

Institut für Computervisualistik
AG Computergraphik
Prof. Dr. Stefan Müller
Postfach 20 16 02
56 016 Koblenz
Tel.: 0261-287-2727
Fax: 0261-287-2735
E-Mail: stefanm@uni-koblenz.de



U N I V E R S I T Ä T
K O B L E N Z · L A N D A U

Fachbereich 4: Informatik

Aufgabenstellung für die Diplomarbeit
Kasjen Kramer
(Mat. Nr. 205210183)

Thema: Konzeption und Realisierung eines interaktiven Java Onlinekurses für Programmier- und Studienanfänger des Fachbereiches Informatik

Die Universität Koblenz bietet für Studienanfänger des Fachbereiches für Informatik einen Java-Vorbereitungskurs mit Anwesenheit für Programmierung an. Da nicht jeder Student diesen Kurs wahrnehmen kann, soll ein zeitlich unabhängiger Onlineprogrammierkurs entwickelt werden.

Ziel der Arbeit ist die Erstellung eines Konzeptes und die prototypische Realisierung eines Onlineprogrammierkurses, welcher die grundlegenden Eigenschaften der Programmiersprache Java und ein besseres Logikverständnis für Studienanfänger vermitteln soll.

Die inhaltlichen Schwerpunkte der Arbeit sind:

1. Auseinandersetzung mit E-Learning
2. Recherche zu Java Onlineprogrammierkursen
3. Konzeption des interaktiven Onlineprogrammierkurses
4. Entwicklung und Implementation des interaktiven Onlineprogrammierkurses
5. Evaluation und Dokumentation der Ergebnisse

Koblenz, den 27.04.17

Kasjen Kramer

Prof. Dr. Stefan Müller

Danksagungen

Ich möchte mich insbesondere bei meiner Freundin Ivanna für die Unterstützung der letzten 2 Jahren meines Studiums bedanken. Ich möchte mich bei meinen Eltern Fokko und Karin Kramer danken, die nicht den Glauben verloren haben, dass ich mein Studium doch noch schaffen kann. Ich möchte mich bei Herrn Prof. Dr. Müller für die regelmäßigen Treffen und die Unterstützung danken.

Zusammenfassung

E-Learning Anwendungen sind heutzutage im Trend. Jedoch nicht nur aus diesem Grund sind sie vom großen Interesse. Im Vordergrund steht ihr didaktisches Konzept.

Heutzutage ist es technisch möglich, die Teilnehmer interaktiv in einen E-Learning Kurs einzubinden. Die Lernschritte können durch die Anwendung kontrolliert werden. Im Gegensatz zu einer Vorlesung, kann die Theorie direkt mit der Praxis verknüpft werden.

Das Ziel der Arbeit ist die Erstellung eines Konzeptes und die Entwicklung eines Prototypen einer interaktiven Entwicklungsumgebung von Java für Programmieranfänger. In dieser Arbeit wurden aktuelle E-Learning Plattformen zu Java analysiert. Aus den gewonnen Erkenntnissen wurde ein Konzept für eine E-Learningplattform entwickelt. Dieses Konzept wurde erfolgreich als Prototyp implementiert. Für die Implementation wurden bewährte Technologien, wie das Django-Framework und das Javascript-Framework Vue.js eingesetzt.

Der Kurs wurde von neun Teilnehmern getestet. Zusätzlich wurde eine Umfrage zu der Kursanwendung durchgeführt. Das Testergebnis bestätigt, dass die Teilnehmer den Kurs bestanden haben.

Abstract

E-learning applications are now the trend. However, not only from this is why they are of great interest. The focus is on their didactic concept.

Nowadays it is technically possible to integrate the participants into an E-Learning course. The learning steps can be controlled by the application. In contrast to a lecture, the theory can be directly linked to the practice.

The aim of the thesis is the creation of a concept and the development of a prototype of an interactive development environment of Java for beginners. Current E-learning Platforms to Java were analyzed in this thesis. From the knowledge gained developed a concept for an E-learning platform. This concept has been successfully implemented as a prototype. For the implementation proven technologies such as the Django framework and the JavascriptFramework Vue.js.

The course was tested by nine participants. In addition, a Survey on the course application. The test result confirmed, that the participants have passed the course.

Inhaltsverzeichnis

Danksagungen

Zusammenfassung

Abstract

1	Einleitung	1
2	Grundlagen	3
2.1	Begriff E-Learning	3
2.2	Lernen, Lehren und E-Learning	3
2.3	Blended Learning	4
2.4	E-Learning Kritik	4
2.5	Lerntheoretische Grundlagen	4
2.6	Lernfähigkeit von Menschen	6
2.6.1	Kurzzeit-, Langzeit- und Arbeitsgedächtnis	6
2.6.2	Cognitive Load Theory	6
2.6.3	Cognitive Load Theory Effekte	6
2.7	Wissen	7
2.7.1	Kognitive Prozesse und Lernziele	8
2.8	Interaktivität	9
2.9	Sinn und Ziele von E-Learning	9
2.10	Fazit	10
3	Verwandte Arbeiten	11
3.1	Einleitung	11
3.2	Vergleich von Online Kursen	11
3.3	E-Learning Unterstützung für Java Programmierung	11
3.4	Verwendung von Micro Worlds	12
3.5	Online Learning Environment : Javala Experiences	12
3.6	GoJava - A Java Development Tool for Beginners	13
3.7	Extreme Apprenticeship Method in Teaching Programming for Beginners	13
4	E-Learningplattformen	14
4.1	Allgemeine Klassifikation von E-Learning Programmen	14
4.1.1	Programmierungsumgebungen	15
4.1.2	Mikrowelten (<i>en: Micro Worlds</i>)	15
4.1.3	Simulation	15
4.1.4	Spiel	16
4.2	E-Learning für Programmier(er)(-anfänger)	16
4.2.1	Online-Buch	16
4.2.2	Tutorials	17

4.2.3	Zusätzliche Aufgaben	17
4.3	Interaktivität bei E-Learninganwendungen	17
4.3.1	Interaktivität von Videos	17
4.3.2	Quiz und der Test	17
4.3.3	Interaktive Aufgabe	18
4.4	Beispiele interaktiver E-Learning Webseiten für Java mit Programmierumgebungen	18
4.4.1	gailer-net.de	18
4.4.2	learnjavaonline.org	19
4.4.3	Codecademy	20
4.4.4	Mobile Applikation	22
4.5	Fazit	23
5	Konzept	24
5.1	Einleitung	24
5.2	Entwicklung	25
5.3	Lernlösung (Vision)	25
5.4	Inhalt	26
5.5	Sprache	27
5.6	Beispiele	27
5.7	Aufgaben	28
5.8	Schritt für Schritt in der Übung	30
5.9	Design von Entwicklungsumgebung	33
5.10	Förderung der Motivation der Benutzer	34
5.11	Lernziele	34
6	Entwurf	35
6.1	Anwendungsfalldiagramm	35
6.2	Anforderungen	36
6.2.1	Administrator	36
6.2.2	Navigation und Übersicht	36
6.2.3	Gestaltung	36
6.2.4	Benutzer	36
6.2.5	Benutzer und Aufgabe	37
6.2.6	Nicht Funktionale Anforderungen	37
6.3	Architektur	38
6.3.1	Server und Client	38
6.3.2	Ablauf: Server und Client	39
6.3.3	Entity-Relationship Diagramm server-seitig	39
6.3.4	Klassendiagramm client-seitig	40

7	Implementation	43
7.1	Technik	43
7.1.1	Django	43
7.1.2	Javascript Frameworks: Vue-js vs JQuery	43
7.2	Server	44
7.3	Client	45
7.4	Überblick der wichtigsten Funktionen	45
7.4.1	Bilder erstellen	46
7.5	Sicherheit	47
7.6	Content Erstellung während Entwicklung	49
7.7	Installation der Anwendung	49
8	Anwendung und Umsetzung	50
8.1	Benutzer	50
8.2	Admin und Content Manger	52
9	Auswertung	54
9.1	Fragebogen	54
9.2	Auswertung	55
9.3	Beobachtung	60
9.4	Schlussfolgerung	61
10	Fazit und Ausblick	62
A	Anhang	63
A.1	Beispiel von der HelloWorld Aufgabe	63
A.2	Installation der Webanwendung	64
A.3	Umfragebogen	66
A.4	Ergebnisse der Umfrage	70

Abbildungsverzeichnis

1	Reiz-Reaktions-Schema als Person und BlackBox	5
2	Lernziele Einteilung nach Bloom	8
3	Lernziele neue Einteilung nach Krathwohl	9
4	Layout von Javala	12
5	Klassifikation von Lernprogrammen nach Midoro, Olimpo .	14
6	Java Kara (Screenshot)	15
7	Codingame (Screenshot)	16
8	learnjava.org_screen	18
9	learnjava.org Kapitel Lerneinheit „Variables and Types“ und Aufgabe, Codeeditor und Ausgabefenster	19
10	Codeacedemy Übersicht	20
11	Codeacedemy Ansicht bei der Übung	21
12	Learn Java Solo Fülltext	22
13	Learn Java Solo Quiz	22
14	Javvy: Eingabe über Tastatur	23
15	Jaavy: Auswahl des Strings	23
16	Theorie - Beispiel - Aufgabe	30
17	Information - Aufgabe	31
18	Info - Aufgabe - Info - Aufgabe	31
19	Info - Aufgabe - Information (Erkenntnis)	31
20	Mockup der Webpage	33
21	Zweites Mockup der Webpage	33
22	Anwendungfalldiagramm	35
23	ER Diagramm Serverseitig	38
24	Ablauf Client- und Serverseitig	40
25	ER Diagramm Serverseitig	41
26	Klassen Diagramm Clientseitig	42
27	Technologie Stack	43
28	Funktionen von Server und Client	46
29	Bild mit den drei Grundfarben	48
30	Startseite der Lernanwendung	50
31	Übersichtsseite der Lernanwendung	50
32	Aufgabe,Code und Ausgabe der Lernumgebung	51
33	Änderung der Anordnung zum klassischen Editor	51
34	Übersicht des Admins Menü	52
35	Die Teilübung kann hier geändert werden	53
36	Test und Speicherung von regulären Ausdrücken	53
37	Frage aus der Umfrage nach der Likert Skala	54
38	Frage 1	55
39	Frage 2	55
40	Frage 3	55
41	Fragen zu Java - 4 bis 9	56

42	Aussage 10	56
43	Aussage 11	57
44	Aussage 12	57
45	Aussage 13	57
46	Aussage 14	58
47	Aussage 15	58
48	Aussage 16	58
49	Note des Kurses	59

1 Einleitung

Motivation

E-Learning, das Lernen über ein elektronisches Gerät, bietet heutzutage die Möglichkeit zu Hause selbständig und zeitlich unabhängig zu lernen. Im Internet gibt es eine Vielzahl von E-Learning Kursen für Programmierer. Die Kurse sind nicht erweiterbar oder änderbar und sind didaktisch nicht immer für Anfänger geeignet. Ein Kurs, dessen Inhalt geändert werden kann und didaktisch aufbereitet werden kann, wird benötigt. Zum einen soll ein E-Learning-Kurs Studieninteressierten einen Einblick in die Programmierung, damit auch einen Teil des Informatikstudiums, geben. Zum anderen soll dieser Kurs Studienanfänger, welche keinen Informatikunterricht in der Schule hatten, Defizite gegenüber Studienanfängern, die Informatikunterricht hatten, ausgleichen. Dadurch soll der Einstieg in das Informatikstudium erleichtert werden und eine anfängliche Barriere überwunden werden. Die Teilnehmer können, im Gegensatz zum Einsteigerkurs mit Anwesenheit, zeitlich immer den E-Learning Kurs besuchen.

Aktuell gibt es in vielen Universitäten einen Vorbereitungskurs für Programmierung mit Anwesenheit. Ein von Seiten der Universität angebotener "E-Learning Kurs für Programmierer" bietet viele Vorteile für Studienanfänger. Defizitausgleich, eine grundlegende Orientierung und zeitliche Unabhängigkeit sind offensichtliche Argumente, die für einen solchen Kurs sprechen.

Ziel der Arbeit

Das Ziel der Arbeit ist es, durch die Einarbeitung gegebener Literatur und der Analyse von aktuellen E-Learning Plattformen, folglich durch deren Erkenntnisse einen Prototyp für eine interaktive E-Learning Plattform für Programmieranfänger, welcher die Programmiersprache Java behandelt, zu entwickeln.

Aufbau der Arbeit

In Abschnitt 2 werden die Grundlagen zum Begriff E-Learning erläutert. In Abschnitt 3 werden aktuelle Publikationen beschrieben, welche sich mit einer E-Learning Plattform für Programmieranfängern auseinandersetzen. In Abschnitt 4 werden aktuelle E-Learning Plattformen für Java Programmieranfänger untersucht. In Abschnitt 5 wird ein daraus entwickeltes Konzept der E-Learning Plattform des interaktiven E-Learning Kurses für Programmieranfänger, vorgestellt. In Abschnitt 7 wird auf die konkrete Implementation der Anwendung eingegangen. Es folgt eine Beschreibung der Architektur. Zusätzlich werden die Funktionen kurz erläutert. In Abschnitt

8 wird die Benutzung der Anwendung gezeigt. In Abschnitt 9 werden die Ergebnisse der ausgeführten Umfrage zur der E-Learning Anwendung ausgewertet. In Abschnitt 10 wird eine abschließende Bewertung der Arbeit formuliert und eine konstruktive Diskussion des Ausblicks verfasst.

2 Grundlagen

In diesem Kapitel werden einige grundlegende Begriffe im Bereich E-Learning erläutert.

2.1 Begriff E-Learning

Der Begriff E-Learning steht für „electronic learning“, auf deutsch übersetzt „elektronisches Lernen“. Es handelt sich dabei nicht um das Lernen von elektronischen Geräten von einer künstlichen Intelligenz, sondern um den Menschen, welcher elektronische oder digitale Medien für das Lernen nutzt. Der Ausdruck E-Learning ist vielseitig und taucht unter anderen Synonymen wie „Online-Learning“ oder „computergestütztes Lernen / Training“ auf. In der Wissenschaft hat sich der Begriff des „multimedialen Lernens“ durchgesetzt. Die digitalen Medien zeigen dem Lernenden Lerninhalte in unterschiedlicher medialer Form [Arn13]. Eine Interaktion und Bearbeitung der Lerninhalte, beispielsweise durch die Beantwortung von Fragen oder dem Lösen von Aufgaben, kann ein Bestandteil von E-Learning sein. E-Teaching wird auch im Zusammenhang mit E-Learning verwendet. Es gibt jedoch einen Unterschied zwischen den Begriffen. Bei E-Teaching steht die lehrende Person im Mittelpunkt. Diese Person erstellt elektronische Lehrinhalte und konzipiert im Detail eine Lehrveranstaltung. Die zentrale Person ist der Lehrende. Im allgemeinen Verständnis werden die beiden Begriffe aber gleich behandelt.

2.2 Lernen, Lehren und E-Learning

Lernen ist die Aktivität, sich neues Wissen anzueignen und die bereits erworbenen Kompetenzen zu erweitern. Die Motivation um etwas zu lernen, könnte eine Aufgabe oder ein Problem sein, durch die Aneignung von erweitertem Wissen und Training in Zukunft lösbar sein wird. Eine Teilhabe und erfolgreiches Mitwirken an der Gestaltung gesellschaftlicher Prozesse, ist auch eine Motivation ([Hol95], S.188).

Lernen ist eine Handlung im Zusammenhang mit einem gewissen gesellschaftlichen Kontext zu betrachten [Arn13]. Daraufhin möchte ich zwei einfache Beispiele erwähnen. Ein Schüler lernt aus dem Grund, sich Basiskenntnisse und Kompetenzen für den Berufsalltag anzueignen. Ein Student lernt an einer Hochschule für sein Fachgebiet relevantes Wissen, um später im Beruf Aufgaben und Probleme in dem entsprechendem Fachbereich lösen zu können. Dabei gibt es verschiedene Formen des Wissens zu vermitteln. Dem Lernenden können möglicherweise Inhalte über eine Vorlesung oder über einen Unterricht präsentiert werden. Der Lernende eignet sich die Inhalte autodidaktisch zum Teil selbst an. Das kann durch einfaches Lesen des Inhaltes erfolgen oder durch das Lösen von Aufgaben,

die er dann mit einer Beispiellösung vergleicht. Für den autodidaktischen Teil des Lernens tritt die Motivation und Selbstdisziplin noch mehr in den Vordergrund.

Der **Lehrende** hat die Aufgabe, Wissen dem Lernenden in möglichst didaktischer Form zu vermitteln. Die Inhalte können in unterschiedlicher Art und Weise präsentiert werden. Es gibt die einfache textuelle Form, beispielsweise das Anschreiben an der Tafel. Es besteht die Möglichkeit den Inhalt multimedial zu präsentieren unter Verwendung digitaler Folien, die der Lernende sich anschließend bei der Nachbearbeitung anschauen kann. Das wäre eine einfache Form des E-Learnings. Eine erweiterte Form des E-Learnings ist in dem Fall, wenn der Lehrende multimediale Programme oder Anwendungen anbietet, die den Lernprozess steuern können. In Bezug auf diese Arbeit, ist es wichtig, dass die Lerninhalte der selbständigen E-Learning Anwendung gut ausgewählt, aufbereitet und präsentiert werden.

2.3 Blended Learning

Blended Learning ist die hybride Form zwischen dem Lernen mit Anwesenheit eines Tutors und E-Learning. Es gibt einen Kurs oder eine Übung, indem der Lernende mit seiner Anwesenheit teilnimmt. Zeitlich außerhalb dieses Kurses, bekommt er über eine E-Learninganwendung Lernmaterialien oder Aufgaben zur Verfügung gestellt. Die Aufgaben müssen gelöst und über die E-Learninganwendung abgegeben werden.

Bei Blended-Learning wird die E-Learning-Komponente als zusätzliches Hilfsmittel gesehen, um den Lernenden noch besser vorbereiten zu können.

2.4 E-Learning Kritik

Es gibt auch kritische Kommentare zum E-Learning ([Arn13] S. 24,26ff). E-Learning kann das Lernen mit einer Person in einem Raum nicht ersetzen ([Arn13] S. 33). Die Person ist direkt ansprechbar und zu einem Diskurs verfügbar. Man kann also sofort in einer klassischen Lernsituation interagieren.

2.5 Lerntheoretische Grundlagen

Drei wichtige wissenschaftliche lerntheoretische Konzepte, für diese Arbeit, sind der Behaviorismus, der Kognitivismus und der Konstruktivismus. Der **Behaviorismus** ist ein älteres Konzept, das seine Blütezeit in den 50er und 60er hatte. Das Konzept geht von dem „Reiz-Reaktionsschema“ (siehe [Dri17a]) aus. Der Reiz ist das, was der Lernende lernen soll. Darauf

zeigt der Lernende eine Reaktion. Der Lernende wird dabei als „BlackBox“ (siehe Abb. 1) gesehen, wobei die bewußtseinsmäßigen Prozesse von ihm nicht in Acht genommen werden.

Bei dem Konzept des Behaviorismus soll durch eine bestimmte Aufgabe ein bestimmtes Verhalten oder Lernziel erreicht werden. Als Beispiel seien die Hunde genannt, welche stark nach ihrem Verhalten konditioniert werden. Bei Menschen wäre es das Prinzip des „Drill-and-Practice“. Durch das „Drill-and-Practice“ Verfahren (siehe [Dri17b]) werden durch wiederholte Übungen bestimmte Kompetenzen angeeignet. Das Lernen von Vokabeln und das ständige Wiederholen ist eine Form davon.

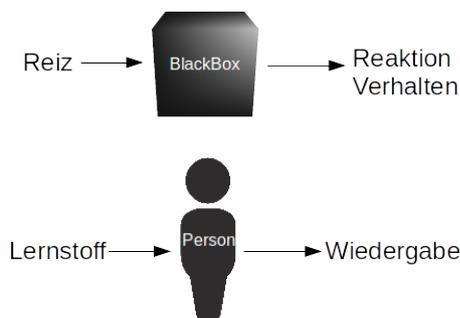


Abbildung 1: Reiz-Reaktions-Schema als Person und BlackBox

Die Lerntheorie des **Kognitivismus** beschreibt selbständige „Denk- und Verstehensprozesse“ ([Arn13] S. 106) des Menschen. Insbesondere spielt die Umgebung eine große Rolle. Der Lernende entdeckt, findet, ordnet und exploriert neue Informationen. Das Individuum kann den bewußtseinsmäßigen Lernprozess selbst steuern, weil er den Freiraum und die entstandene Neugier besitzt. Auf E-Learning bezogen, bieten Lernumgebungen, wie Simulationen oder Mikrowelten, diesen Freiraum an. Weiteres dazu im nächsten Abschnitt 4.1.2 auf Seite 15. Der **Konstruktivismus** beschreibt in der Lerntheorie eine Ansicht, dass das Wissen nicht objektiv ist, sondern subjektiv. Jedes Individuum baut sein Wissen auf sein vorheriges Wissen auf. Das Wissen wird miteinander verknüpft. Der Lernweg ist dadurch individuell. Dem Lernenden werden die Freiheiten gewährt, Probleme selbst zu entdecken. Er eignet sich das Wissen selbstständig an.

Bei dem konstruktivistischen „Anchored Design“ ([Ker01] S.79, [NDH⁺08] S.24ff) geht es darum, einen „Anker“ zu finden der die Lernenden auf einen richtigen Lernweg hinweist und sie motiviert. Man gibt also eine Instruktion durch einen Denkanstoß.

Bei der **instruktionistischen** Lerntheorie wird die lernende Person aufgefordert etwas zu tun. Eine übliche Lernweise ist „vormachen, erklären, nachmachen, üben“. Das Wissen wird passiv aufgenommen und dann durch Übungen vertieft. Der Lernende trägt wenig Eigenverantwortung, weil der Inhalt klar vorgegeben ist. Dadurch ist sein Lernweg nicht individuell. Der Instruktor entspricht einem Modell des Behaviorismus. Bei dem **Instruktorismus** ist der Lernende nur Empfänger des Wissens. Der Lehrende präsentiert nur seinen Lehrinhalt und nimmt dabei keine Rücksicht auf den Lernenden. Ein Beispiel wäre der Frontalunterricht.

Schlussfolgernd kann gesagt werden, dass die Begriffe der Lerntheorien sich nicht genau trennen lassen. Ihr Übergang ist fließend.

2.6 Lernfähigkeit von Menschen

2.6.1 Kurzzeit-, Langzeit- und Arbeitsgedächtnis

Eine Bekannte und ältere Annahme über die Struktur des Gedächtnisses, ist die Einteilung in das Kurz- und Langzeitgedächtnis. Das Arbeitsgedächtnis kann nur maximal sieben Schritte gleichzeitig verarbeiten (siehe [NDH⁺08] S.43). Wenn die Informationen nicht wiederholt werden, löscht das Kurzzeitgedächtnis diese innerhalb weniger Sekunden. Das Langzeitgedächtnis speichert Informationen für längere Zeit. Es kann sich dabei um etwas auswendig Gelerntes handeln oder eine komplexe Information, die auf eine andere Information aufbaut.

2.6.2 Cognitive Load Theory

Eine der häufigsten Theorien, die mit dem Lernvermögen eines Menschen in Verbindung gebracht wird, ist die „Cognitive Load Theory“ ([NDH⁺08] S.45). Bei dieser Theorie spielt das „Arbeitsgedächtnis“ eine zentrale Rolle. Man geht davon aus, dass das Arbeitsgedächtnis über eine beschränkte Kapazität verfügt. Es kann nur eine bestimmte Anzahl von Prozessen gleichzeitig verarbeiten. Bei der genannten Theorie wird die Belastung in drei Arten unterteilt. Die **intrinsische kognitive Belastung** ([NDH⁺08] S.46) nimmt stark zu wenn die Komplexität des zu lernenden Objektes sehr hoch ist. Das Gegenbeispiel für eine geringe intrinsische Belastung, wäre das Lernen von Vokabeln. Die Menge der Vokabeln mag groß sein, jedoch stehen die Vokabeln in keiner komplexen Verbindung miteinander. Es geht darum, wie unabhängig die Informationen miteinander verbunden sind und wie sie selbständig gelernt werden.

Die **extrinsische kognitive Belastung** hängt von der äußerlichen Darstellung der Informationen ab. Ist die äußerliche Gestaltung umständlich, enthält zum Beispiel eine Informationsseite viele Querverweise und unnötige Bilder, ist die extrinsische Belastung hoch. Einfach ausgedrückt, kann die Gestaltung der Informationen stark von dem eigentlichem Inhalt ablenken. Die **lernbezogene kognitive Belastung** beschreibt, wie viel Lernaufwand der Lernende aufbringen muss, um die Information zu verstehen. Dabei kommt es auf das vorhandenes Wissen, bzw. Schemata an. Der Lernende baut neue Schemata auf seine vorherigen auf.

2.6.3 Cognitive Load Theory Effekte

Bei multimedialen Anwendungen, können Effekte für den Benutzer auftreten, die zunehmend die intrinsische und extrinsische Belastung fördern

oder entlasten.

Zentraler Bestandteil des **Worked Example Effect** ([NDH⁺08] S.47) ist, dass dem Lernenden funktionierende Lösungsbeispiele zu einer Aufgabe gezeigt werden. Der Lernende sieht und erkennt den Lösungsweg zu dieser Aufgabe. Würde er die Lösung nicht kennen, müsste er den Lösungsweg selbst finden und könnte sich damit lange aufhalten und möglicherweise fehlerhafte Lösungswege verfolgen.

Der **Split Attention Effect** entsteht, wenn zusammenhängende Elemente voneinander getrennt sind. Ist der erste Teil einer Informationsfolie auf einer Seite und der andere Teil auf der anderen Seite, sind die Informationen voneinander getrennt. Der Benutzer muss sich die erste Folie merken, wenn er auf der zweiten Seite ist. Dadurch wird der Benutzer überfordert. Diesen Effekt will man vermeiden, um das Arbeitsgedächtnis des Lernenden nicht unnötig zu belasten.

Der **Redunancy Effect** entsteht wenn überflüssige oder doppelte Informationen in der Präsentation vorhanden sind. Beispielsweise gibt es die gleiche Informationen auditiv und als Text.

Der **Expertise Reversal Effect** entsteht, wenn für einen Lernenden, mit stark ausgeprägten Vorwissen in einem bestimmten Gebiet, die Anwendung dieses Wissens nochmals erklärt wird. Für den Lernenden, welcher kein Vorwissen hat, muss die Lernumgebung das Thema didaktisch aufbereiten. Dies bedeutet, dass mehr Informationen angezeigt werden müssen. Dieser Effekt lässt sich vermeiden, in dem man eine eindeutige Zielgruppe definiert.

2.7 Wissen

Es besteht bei Wissen eine Unterscheidung zwischen deklarativem, prozeduralem und kontextuellem Wissen. Bei dem deklarativem (*beschreibend*) Wissen handelt es sich um das Wissen, das einfach sprachlich ausgedrückt werden kann, wie beispielsweise die Aussage „Java ist eine Programmiersprache“. Das prozedurale oder iterative Wissen beschreibt ein Verfahren, das deklaratives Wissen miteinander verknüpft. Das prozedurale Wissen wird verwendet, um einen Handlungsablauf wiederzugeben [Pro17]. Als gutes Beispiel lässt sich das „Schuhebinden“ benennen. Es ist alltäglicher Ablauf eines Menschen, welcher motorische und physikalische Erkenntnisse miteinander verknüpft und somit erfolgreiche Ergebnisse beim Ausführen der Tätigkeit liefert.

Das kontextuelle Wissen ([Ker01] S.168) umfasst Strategien, um ein Problem zu lösen. Es beschreibt wann und wo das prozedurale Wissen angewendet werden soll. Es organisiert und verwaltet den Zugriff auf anderes Wissen. Das kontextuelle Wissen kann einem Lernenden beigebracht werden, indem ihm authentische Situationen, die oft auftreten, erklärt werden und ihm eine Lösung präsentiert wird.

Bei dem metakognitivem Wissen [Kat17] geht es um das selbstreflektierte Wissen einer Person. Die Person ist sich über ihre kognitiven Prozesse und Fähigkeiten bewusst. Dadurch kann sie einschätzen, was ihr an Wissen fehlt und wie sie es sich aneignen kann. Besteht ein Problem oder eine Aufgabe, sucht die Person sich eine erfolgreiche Strategie, das Wissen anzueignen und dann eine entsprechende Lösung zu finden. Auf einen E-Learning Kurs bezogen, spielt der selbst regulative Prozess des Lernenden eine wichtige Rolle ([NDH⁺08] S.73). Die Person kann ihren Lernprozess selbst steuern, indem sie plant, was sie lernt.

2.7.1 Kognitive Prozesse und Lernziele

In der originalen Arbeit von Benjamin Bloom aus dem Jahr 1956 [Blo17b] werden Lernziele in sieben Stufen unterteilt. Von unten nach oben steigt

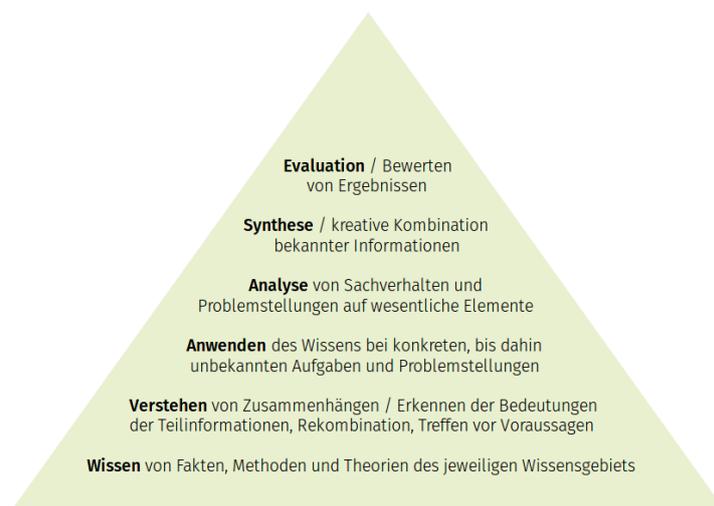


Abbildung 2: Blooms Einteilung [Blo17a]

die lernbezogene kognitive Belastung. Die Person muss sich demnach in höheren Stufen mehr anstrengen. Die erste Stufe beschreibt das Wissen (siehe S.7 2.7). In der zweiten Stufe geht es um das „Verstehen“. Informationen sollen in einem Zusammenhang gebracht werden. Diese Informationen sollen dann in eigenen Worten wiedergegeben werden. In der dritten Stufe geht es um das Anwenden des Wissens im praktischen Bereich. Hier wird das prozedurale Wissen gefordert.

Bei der „Analyse“ wird der Zusammenhang der Elemente der Informationen genau untersucht. In der Synthese findet bei der Person der kreative Prozess statt. Aus vorigem Wissen, wird etwas neues geschaffen. In der Stufe der Evaluation „fließen die vorigen Kompetenzen“ mit ein. Bei der Stufe der „Evaluation“ geht es darum objektive Urteile zu fällen. Auch

wenn die Theorie älter ist, hat sie heute noch einen hohen Stellenwert. In der Neuinterpretation aus dem 2001 von Krathwohl [Kat17] gibt es Änderungen. Der kreative Prozess wird höher angesetzt als der evaluierende Prozess. Die Nomen, die die Stufen der Lernziele beschreiben, werden nach

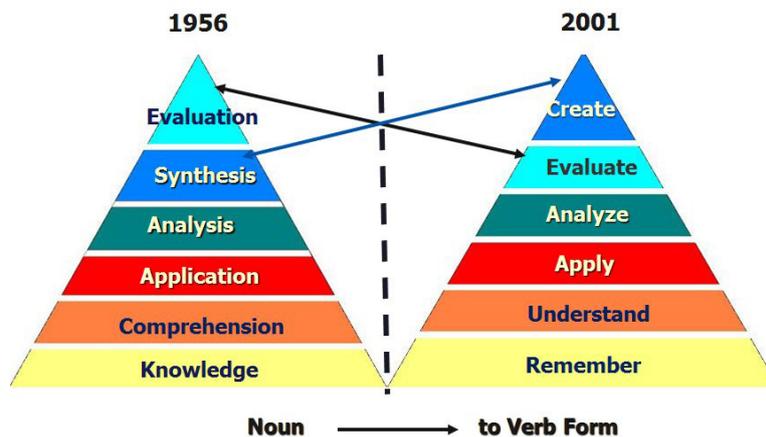


Abbildung 3: Anderson und Krathwohl Einteilung [Kat17]

Krathwohl (siehe Abbildung 3) in Verben umgewandelt.

2.8 Interaktivität

Interaktionen wird im Zusammenhang dem im den 80er Jahren entstandenen Begriff „Mensch Maschine Interaktion“ ([IK02] S. 128), welcher das Nutzungsgeschehen zwischen einem Menschen und einem Menschen bezeichnet, verwendet. Der Begriff der Interaktivität lässt sich als Eigenschaft eines Systems und Software beschreiben, das eine Eingriffs- und Steuermöglichkeit eines Benutzers besitzt. Das Maß der Interaktivität kann unterschieden werden. Kann der Benutzer auf mehr Elemente frei zugreifen, steigt der Grad der Interaktivität.

2.9 Sinn und Ziele von E-Learning

E-Learning soll das selbstständige Lernen des Lernenden fördern. Der Lernende soll letztendlich Einfluss auf seinen eigenen Lernprozess haben. Es wird empfohlen, keinen strikten Lernprozess vorzugeben. Durch die Interaktivität kann der Lernende das Lerngeschehen direkt beeinflussen ([Zum10] S.52). Ein Beispiel dafür wäre, dass der Lernende die Bereiche frei auswählen kann, anstelle einer vorgegebenen linearen Abarbeitung der einzelnen Kapitel. Die Motivation einer Einführung der E-Learning Anwendungen stammt in der Mehrheit von Unternehmen. Sie haben häufig das Bedürfnis

viele Leute einzuarbeiten. Im Gegensatz steht der Aufwand eines Tutoriums mit anwesenden Tutoren. Das kostet verhältnismäßig viele Ressourcen.

Motivation von E-Learning ist die Einsparung von Ressourcen und eine effektive Methode Lernende zu unterrichten.

2.10 Fazit

E-Learning ist ein vielseitiger Begriff. Die einfachste Interpretation ist das die lernende Person elektronische Medien zum lernen nimmt. Die Motivation des Lernenden und die Aufgabe des Lehrenden, das Wissen zu vermitteln, ist dabei von besonderer Bedeutung, weil es die grundlegenden Eigenschaften sind, die für ein erfolgreiches Durchführen der beiden Handlungen erforderlich sind. Die Lerntheorien sind begrifflich eng miteinander verbunden. Die wichtigsten und aktuellen Lerntheorien beziehen den bewusstseinsmäßigen Lernprozess des Lernenden mit ein. Es gibt unterschiedliche Arten von Wissen und die Belastung beim E-Learning spielt eine wichtige Rolle.

3 Verwandte Arbeiten

3.1 Einleitung

Viele Arbeiten beschäftigen sich mit einem Einsteigerkurs für Studien- und Programmieranfänger, der in den meisten Fällen ein ganzes Semester lang dauert. Sie befassen sich im Genaueren mit den Konzepten und Analysen von Erstsemesterkursen für Programmierer. Einige dieser Publikationen werden bei der Konzeptausarbeitung erwähnt, denn deren Fokus liegt in der Analyse der Konzepte.

Dieser Abschnitt soll einen generellen Überblick über wissenschaftliche Arbeiten im Bereich von E-Learning Programmierkursen für Anfänger verschaffen.

3.2 Vergleich von Online Kursen

In der Publikation „Comparing the Effectiveness of Online Learning Approaches on CS1 Learning Outcomes“ [LK15] geht es um einen Vergleich von drei Online-Programmierkursen. Die drei Plattformen sind Codecademy [Cod17a], Gidget und Gidget Puzzle Designer. Im Abstract wird darauf hingewiesen, dass immer mehr Menschen Online-Ressourcen benutzen, um programmieren zu lernen. Die Autoren der Publikationen möchten die Effektivität solcher Programmierkurse überprüfen. Es wird die Fragestellung formuliert, ob die Probanden durch den gängigen „CS1“ Kurs gleiche Kompetenzen erworben haben. Durch einen Vortest wird der jeweilige Kenntnisstand festgestellt und durch einen Test nach der Teilnahme eines Kurses werden ihre zu gewonnenen Kenntnisse überprüft. Die beiden Tests gelten als Kriterien des Vergleiches. Schlussfolgernd wird in der Publikation gezeigt, dass Codecademy und Gidget eine deutliche Steigerung der Kenntnisse erheben. Dies lässt sich anhand der Ergebnisse des Nachtests erkennen.

3.3 E-Learning Unterstützung für Java Programmierung

Der Artikel „Technology-enhanced learning for Java programming“ [IP11] beschreibt, wie zwei unterschiedliche Universitäten ein E-Learning Tool „Moodle“ parallel zu ihrem Einführungskurs der Programmierung benutzen. Ein theoretischer Test in Form von Fragen, Materialien zum Kurs und Übungen werden über diese Plattform zur Verfügung gestellt. Zudem besteht eine Möglichkeit über die Plattform mittels Email, Nachricht oder Foren mit dem Lehrenden zu kommunizieren. Jedoch wurde festgestellt, dass die Lernenden eine direkte und persönliche Kommunikation bevorzugen. Daraus ergab sich, dass OpenSource-Plattformen, wie Moodle, einen Mehrwert in der Zukunft haben.

3.4 Verwendung von Micro Worlds

Einige Publikationen erwähnen die Verwendung von einer Micro World, um den Studenten „problemlösendes Denken“ beizubringen. In dem Artikel „A Comparison of Different Approaches to the Introductory Programming Course“ stellen verschiedene Tutoren eines Einsteigerkurses für Programmierung ihren Ansatz vor. In Bezug auf Java benutzt der Tutor Joe Bergin "Karel J Bot" für seinen Kurs. Er ist davon überzeugt, dass Studenten nach einigen Wochen ein solides Bild vom Programmieren und außerdem gute problemlösende Fähigkeiten besitzen werden. Der Tutor Raymond Lister verwendet die Micro World „PigWorld“, um Studenten im ersten Semester Java zu lehren. In der Puplication "Reducing the Dropout Rate in an Introductory Programming Course" [Yad11] wird erklärt, dass nach ein Semester langen Einsatz "Karel J" abgesetzt wurde, aufgrund des objektorientierten Ansatzes, welcher die Anfänger verwirrt hat. Schließlich soll der prozedurale Ansatz beigebracht werden.

3.5 Online Learning Environment : Javala Experiences

In der Puplication „On the Role of Gamification and Localization in an Open Online Learning Environment: Javala Experiences“ [LAIM15] wird der Gamification Aspekt der Online Lernumgebung „Javala Experiences“ untersucht und bewertet. Die Lernumgebung wurde nur für eine begrenzte Anzahl von Benutzern erstellt. In der Lernumgebung werden Programmieraufgaben gestellt. Der daraus entstandene Code, lässt sich ausführen und wird anschließend evaluiert. Die Umsetzung von Gamification erfolgt durch :

- Punkte für bestandene Aufgaben
- die Sammlung von Abzeichen („badges“)
- die Sammlung der Errungenschaften („achievements“)
- die Liste der Top Benutzer



Abbildung 4: "The layout of a Javala page" [LAIM15]

Aus der Untersuchung wird das Fazit gezogen, dass der „Gamification“ Aspekt einen positiven Effekt auf die Weiterbenutzung der Plattform hat.

3.6 GoJava - A Java Development Tool for Beginners

In der Publikation „GoJava - A Java Development Tool for Beginners“ [Wis06] wird eine Webanwendung, die an der Universität von Gloucestershire angewendet wird, um Studienanfängern Java beizubringen, beschrieben. Die Anwendung wurde aufgrund der Tatsache entwickelt, dass die professionellen Tools zu viele Menüs haben und damit den Benutzer vom eigentlichen Programmieren ablenken. Das Benutzerinterface wird auf ein Minimum reduziert und auf die nötigsten Funktionen beschränkt. Das Hauptziel der Anwendung besteht im einfachen Erstellen des Codes. Die Erfahrungen mit "GoJava" sind bei den Studenten überwiegend positiv.

3.7 Extreme Apprenticeship Method in Teaching Programming for Beginners

In der Publikation „Extreme Apprenticeship Method in Teaching Programming for Beginners“ [VPL11] wird ein Konzept des „Cognitive Apprenticeship“ erläutert. Es wird beschrieben, wie das Programmieren den Studierenden in drei Phasen beigebracht werden kann. In der ersten Phase stellt der Lehrer funktionierende Beispiele vor. Er hält sich nicht mit Programmstrukturen auf. In der zweiten Phase - „scaffolding“ - gibt der erfahrene Lehrer den Studenten Hinweise, um die Aufgabe zu lösen (Denkanstöße). Er gibt keine explizite Lösung vor. In der dritten Phase soll der Student soweit sein, dass er selbst die Aufgabe lösen kann.

4 E-Learningplattformen

Dieses Kapitel soll einen generellen Überblick über E-Learning Plattformen für Programmierer geben. Es folgt eine grobe Klassifikation der verschiedensten Formen. Diese Arbeit konzentriert sich auf interaktive E-Learning Plattformen für Programmierer, deshalb werden diese genauer vorgestellt und analysiert.

4.1 Allgemeine Klassifikation von E-Learning Programmen

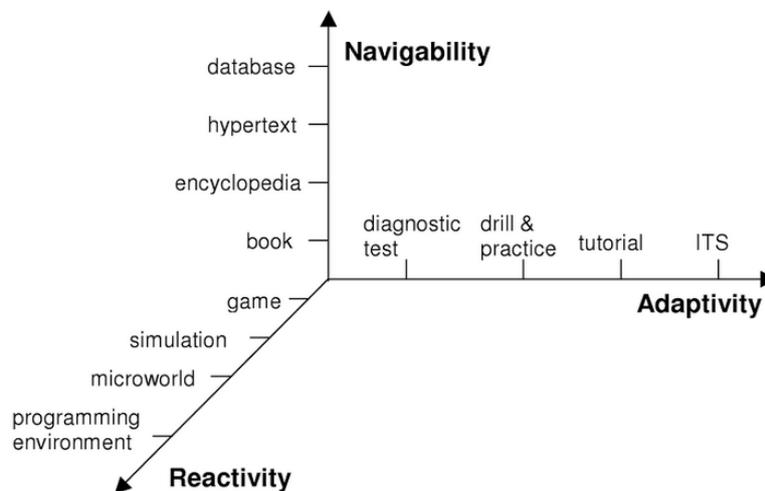


Abbildung 5: Klassifikation von Lernprogrammen nach Midoro, Olimpo et al. (1991, zit n. [Sch02], 2002, S. 47).

Die Abbildung 5 der Klassifikation von Lernprogrammen stammt aus dem Jahr 1991. Allerdings ist sie weiterhin aktuell (2017). Mit dieser Klassifikation lassen sich E-Learning Plattformen aus dem Internet kategorisieren. Die Klassifikation ist in drei Dimensionen eingeteilt. Die Dimension der Navigierbarkeit („Navigability“) beschreibt den Grad der Linearität in der Verteilung der Informationen. Es fängt mit einem einfachen Buch an, bei dem der Text sehr linear dargestellt wird, und geht hinüber zum quer verknüpften Dokument (Hypertext), sowie dem Zugriff auf eine Datenbank. Die Dimension der Anpassungsfähigkeit („Adaptivity“) beschreibt, wie flexibel sich die Aufgaben an das Lernverhalten des Benutzers anpassen. Die Höchste Stufe ist das „Intelligente Tutorielle System“, das sich durch einen intelligenten Algorithmus an das Niveau des Lernenden anpasst. In der Dimension der Reaktivität („Reactivity“) wird die Lernumgebung, in der sich der Lernende befindet, gezeigt. In Folgendem werden Bei-

spiele der Dimension der Reaktivität benannt und kurz beschrieben. Diese Beispiele haben einen starken Bezug zu der Programmiersprache Java [Jav17a].

4.1.1 Programmierumgebungen

Eine Programmierumgebung ist eine Ansammlung von Werkzeugen und Programmen, welche bei der Entwicklung von Programmen helfen. Ausführliche Beispiele zu den Java E-Learning Anwendungen werden in Kapitel 4.4 auf S.18 gezeigt.

4.1.2 Mikrowelten (*en: Micro Worlds*)

Bei Mikrowelten handelt es sich im allgemeinen um Simulationen, die einen kleinen Teil einer Welt darstellen. Bei Mikrowelten wird für die Programmierung nur eine Teilmenge einer Programmiersprache zur Verfügung gestellt. Sie sollen anfänglichen Benutzern einen leichteren Einstieg verschaffen. Durch eine grafische Oberfläche lässt sich ein Objekt durch Befehle steuern. Für Java existiert beispielsweise der Java-Hamster und Kara der Marienkäfer [Kar17]. Beide Mikrowelten müssen als jar-Datei runtergeladen und auf dem Desktop ausgeführt werden.

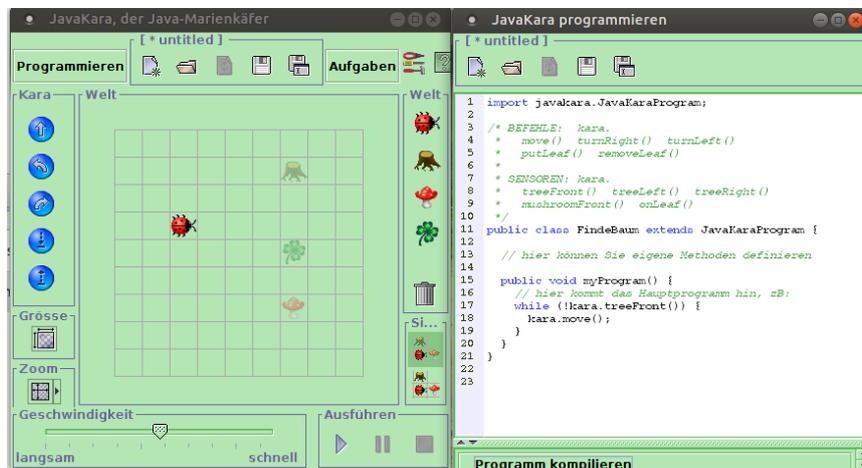


Abbildung 6: Java Kara (Screenshot)

4.1.3 Simulation

Die vorher vorgestellten Mikrowelten gehören zu dem Bereich der Simulationen. Die Tätigkeiten, wie ein Laborversuch, können interaktiv aufbereitet und anschließend in der Simulation ausgeführt werden.

Da die Programmierung direkt am PC durchgeführt wird, und ein Resultat sofort ersichtlich ist, besteht kein Bedarf an einer direkten Simulation.

4.1.4 Spiel

Das Programmieren lässt sich spielerisch lernen. Gamification ist ein Ansatz dafür. Beispielsweise könnte das Programmieren kleiner Algorithmen in einer Mikrowelt spielerisch ausgeweitet werden. Auf der Webseite von Codingame [Cod17b] muss man einen Algorithmus programmieren, der eine bestimmte Aufgabe in einem Spielgeschehen hat. Die Abb. 7 zeigt ein Spiel von Codingame, wo man Code programmieren muss, um ankommende Raumschiffe „abzuschießen“. Es ist ebenso möglich in anderen Spielen gegen andere menschliche Gegner zu spielen. Die Seite ist allerdings besser für erfahrene Programmierer, die sich in einem Wettbewerb messen wollen, geeignet.

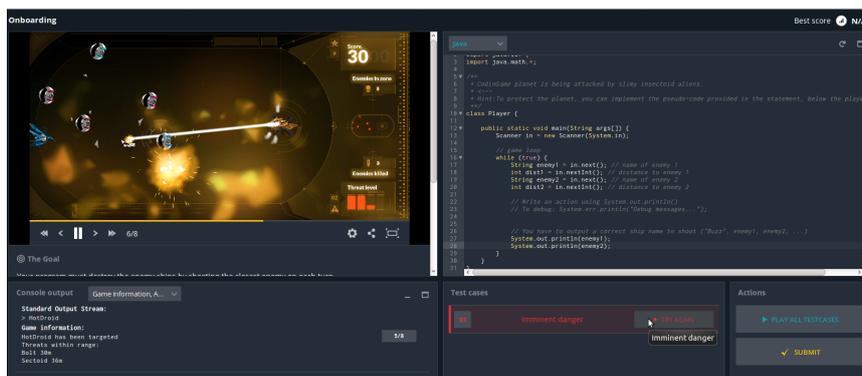


Abbildung 7: Codingame (Screenshot)

Wie an den Beispielen zu sehen ist, kann man nicht immer eine klare Grenze zwischen Mikrowelten, Simulationen und Spielen ziehen.

4.2 E-Learning für Programmier(er)(-anfänger)

Es gibt zahlreiche Onlinematerialien zu Programmiersprachen. In diesem Abschnitt werden die Materialien zum Lernen von Programmiersprachen kategorisiert. Insbesondere wird dabei auf den Lernstoff für Programmieranfänger eingegangen, wodurch die Menge der E-Learning Medien erheblich eingeschränkt wird.

4.2.1 Online-Buch

Ein Online-Buch zu einer Programmiersprache ist als HTML Form im Internet verfügbar. Durch die unproblematische Navigierbarkeit, welche durch

den Hypertext gewährleistet wird, kann schnell auf die Themen zugegriffen werden. Das Buch kann als Nachschlagewerk verwendet werden. Ein sehr bekanntes Online-Buch für Java ist „Java ist eine Insel“ [Jav17b]. Es hat 19 Hauptkapitel und umfasst mit 1300 Seiten ausgedruckte Ausgabe einen großen Teil der Informationen über die Programmiersprache Java.

4.2.2 Tutorials

Ein Tutorial ist eine Anleitung, welche ein Programmierbeispiel Schritt für Schritt durchgeht und es dabei beschreibt. Ein Tutorial, wie es auf Oracle [Ora17] für Java versteht, besitzt kurze Beschreibungen von Begriffen und ein Beispiel zu dem zugehörigen Thema. Daneben gibt es Tutorials in medialen Form eines Videos. Oft wird in solchen Tutorials ein komplettes Programm aufgebaut und begleitend audiovisuell erklärt. Der Benutzer sieht genau, wie der Programmierer in einer Entwicklungsumgebung vorgeht. Diese Art der Lernmedien ist im Vergleich zu anderen, wie beispielsweise Bücher, sehr praxisorientiert.

4.2.3 Zusätzliche Aufgaben

Ein häufiger Bestandteil der Bücher und Tutorials sind Aufgaben zu zugehörigen Themen. Es kann eine Musterlösung vorgegeben werden. Der Lernende muss die Musterlösung anschließend mit seiner Lösung vergleichen.

4.3 Interaktivität bei E-Learninganwendungen

4.3.1 Interaktivität von Videos

Im Prinzip können auch Videos mit einer Vorspul- und Rückwärts-taste als Interaktivität gewertet werden. In diesem Abschnitt werden die Modelle kategorisiert, die eine direkte interaktive Kommunikation mit dem Benutzer ausführen.

4.3.2 Quiz und der Test

In einem Test oder Quiz wird auf den E-Learning Plattformen wird eine Frage gestellt und es werden zu dieser Frage verschiedene Antwortmöglichkeiten vorgegeben. Davon kann meistens nur eine Antwort ausgewählt werden. Eine andere Form der Abfrage ist, dass hier keine Antwortmöglichkeiten gegeben sind. Man muss die Antwort selbst eingeben. Das wird von der Anwendung im Anschluss mit der mustergültigen Antwort verglichen.

Demzufolge unterscheidet man zwischen einem Lückentext, einem Vokabeltest sowie einer Multiplechoice-Frage. Diese Form der Interaktion ist technisch am einfachsten zu programmieren.

4.3.3 Interaktive Aufgabe

Eine komplexere Form eines Frage/ Antwort-Tests ist eine Aufgabe zu stellen, die der Lernende lösen soll. An dieser Stelle bietet sich eine Programmieraufgabe an. Der Benutzer muss bei diesem Aufgabentyp den Code eingeben. Anschließend kann er den Code ausführen. Das Ergebnis der Ausgabe des ausgeführten Codes wird von der Anwendung auf seine Richtigkeit getestet. Der Benutzer bekommt eine Rückmeldung, ob er die Aufgabe bestanden hat.

4.4 Beispiele interaktiver E-Learning Webseiten für Java mit Programmierumgebungen

In diesem Abschnitt werden Beispiele bekannter interaktive E-Learning Plattformen analysiert. Der Blick fokussiert sich dabei auf den Inhalt und den Grad der Interaktivität.

4.4.1 gailer-net.de

Die Internet-Ressource [Gai17] stellt eine einsteigerfreundliche Einführung in Java dar. Es ist also explizit ein Kurs für Anfänger. Der Kurs besteht aus 11 Teilen mit insgesamt 58 Kapiteln. Jedes Kapitel behandelt in kurzer Form (unter 100 Wörter) ein kleines Thema. Zu jedem Kapitel gibt es ein interaktives Quiz oder einen kleinen Test, wodurch sich die Interaktivität des Kurses beschränkt. Die Lernkontrolle erfolgt durch die eigene Kontrolle. Positiv zu bemerken ist die Tatsache, dass die Erklärungen kurz und verständlich sind. Es lässt sich jedoch das veraltete Design der Benutzero-



Abbildung 8: learnjava.org Kapitel Lerneinheit „Variables and Types“ und Aufgabe, Codeeditor und Ausgabefenster

berfläche kritisieren.

4.4.2 learnjavaonline.org

Die Webseite [lea17] bietet acht Basislerneinheiten und zehn erweiterte Lerneinheiten zu der Programmiersprache Java. Der Lernstoff wird teilweise genau erklärt, allerdings sind die Erklärungen an manchen Stellen zu unpräzise. Eine Lerneinheit besteht aus einem Fließtext mit Informationen und einem Programmcode, der ausgeführt werden kann. Am Ende der Lerneinheit erfolgt eine abschließende Aufgabe. Anzumerken ist, dass die Aufgaben für Programmieranfänger nicht geeignet sind.

Die folgende Aufgabe (siehe Textabb. 1) ist aus der Lerneinheit „Variables and Types“ übernommen worden. Man geht explizit auf viele Datentypen ein, was den Programmieranfänger überfordern könnte.

Textfeld 1: Aufgabe von learnjava.org

Exercise Create all of the primitives (except long and double) with different values. Concatenate them into a string and print it to the screen so it will print: H3110 w0r1d 2.0 true

boolean

Every comparison operator in java will return the type `boolean` that not like other languages can only accept two special values: `true` or `false`.

```
boolean b = false;
b = true;

boolean table = false;
b = table || !table;
if (b) {
    System.out.println(table);
}

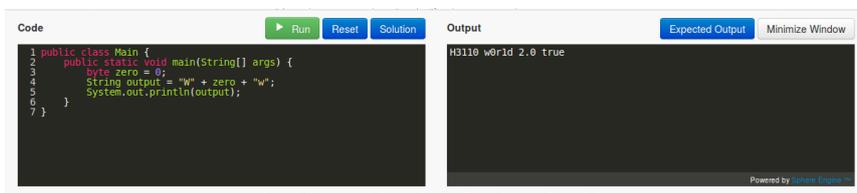
int children = 0;
b = children; // will not work
if (children) { // will not work
    // will not work
}
```

Execute Code

Exercise

Create all of the primitives (except long and double) with different values. Concatenate them into a string and print it to the screen so it will print: H3110 w0r1d 2.0 true

Start Exercise



The screenshot shows a code editor with a 'Code' pane on the left and an 'Output' pane on the right. The code in the 'Code' pane is as follows:

```
1 public class Main {
2     public static void main(String[] args) {
3         byte zero = 0;
4         String output = "H" + zero + "r";
5         System.out.println(output);
6     }
7 }
```

The 'Output' pane shows the result: "H3110 w0r1d 2.0 true". Above the code editor are buttons for 'Run', 'Reset', and 'Solution'. Above the output pane are buttons for 'Expected Output' and 'Minimize Window'. At the bottom of the editor, there is a small text: 'Copyright © 1999-2009 LearnJavaOnline.com. Based on the Terms of Use and Privacy Policy. Powered by @codecademy'.

Abbildung 9: learnjava.org Kapitel Lerneinheit „Variables and Types“ und Aufgabe, Codeeditor und Ausgabefenster

Die Aufgabe wird in dem Editor (siehe Abb. 9) programmiert, wobei der ausgeführte Code in einem Ausgabefenster angezeigt wird. Als Hilfe-

stellung kann die erwartete Ausgabe sowie die Lösung erscheinen. Die Seite wirkt unübersichtlich und die Werbung lenkt häufig ab. Das schränkt die Aufnahmefähigkeit des Benutzers ein. Positiv hervorzuheben ist, dass es einen Code-Editor und ein Ausgabefenster gibt, zudem ist eine Registrierung und Anmeldung nicht erforderlich.

4.4.3 Codecademy

Codecademy [Cod17a] ist eine E-Learningplattform, die viele Kurse für verschiedene Programmiersprachen anbietet. Der kostenlose Teil des Java Programmierkurses besteht aus vier Kapiteln. Die ersten zwei Kapitel „Introduction to Java“ und „Conditionals and Control Flow“ sind für Anfänger gedacht. Ein Kapitel (siehe Abb. 10) besitzt mehrere Lerneinheiten. Die Anzahl der Lerneinheiten beläuft sich auf 10 bis 12. Die jeweilige Lernein-

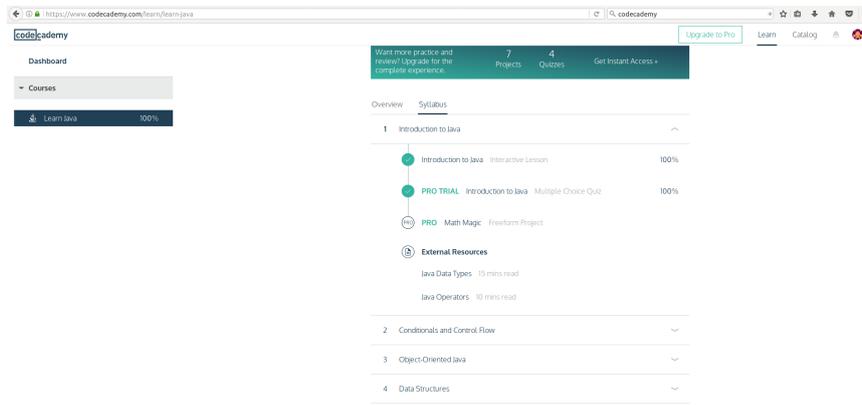


Abbildung 10: Codeacademy Übersicht

heit besitzt eine strikte Abfolge. Die Lernuntereinheit fängt mit einem oft umfangreichen Einleitungstext an. Anschließend folgen die Instruktionen, welche im Code-Editor eingetippt werden müssen. Das Programm ist immer vorgegeben, anschließend muss nur ein Teil des Codes geändert oder ergänzt werden. Dieser Code kann danach ausgeführt werden. Die Eingabe des Benutzers sowie die Ausgabe des Programms sind für das Bestehen der Aufgabe bestimmend. Die Ausgabe und der Code wird von der Webapplikation überprüft und ausgewertet. Ist die Instruktion erfolgreich ausgeführt worden, kann man zur nächsten Instruktion gehen. Wie in Abbildung 11 zu sehen ist, wird die Webseite in drei Hauptelemente aufgeteilt. Die entsprechenden Informationen zu dem Thema und die dazugehörigen Instruktionen sind auf dem linken Teil der Seite vorhanden. Im mittleren Teil befindet sich der Codeeditor und auf der rechten Seite ist das Ausgabefenster.

Codecademy ist eine jüngere Plattform, die ein modernes Design anbietet.

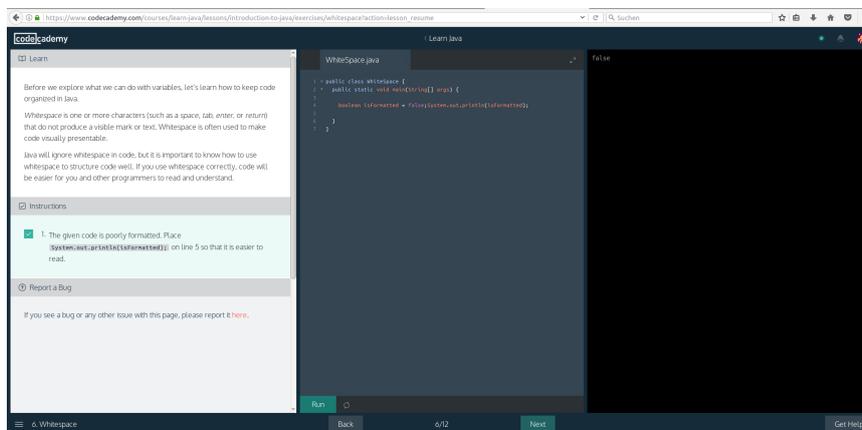


Abbildung 11: Codecademy Ansicht bei der Übung

Der vorgegebene Code erleichtert das Lösen der Aufgaben, aus dem Grund muss der Benutzer nicht viel Zeit in die Erstellung des kompletten Codes investieren. Die Anweisungen in den unteren Lerneinheiten sind kurz, verständlich und nicht schwierig.

Man lernt jedoch keine komplexen Algorithmen in dem Kurs. Die Instruktionen erfordern lediglich einen minimalen Transfer.

Codecademy ist für Anfänger geeignet sowie für Personen, die bereits Erfahrungen in einer Programmiersprache haben und eine neue Programmiersprache erlernen wollen. Die Plattform bietet einen kurzen und guten Einblick in die Programmiersprache Java an. Der kostenpflichtige Teil wird in dieser Arbeit nicht bewertet.

4.4.4 Mobile Applikation

Für den mobilen Bereich gibt es mehrere interaktive Java Lernanwendungen, beispielsweise „Learn Java“ von SoloLearn [Sol17], „Codebox“ und „Javvy“. Diese mobilen Anwendungen lassen sich in die gleichen Kategorien, wie eine Webapplikation, einordnen. Codebox ist ein digitales Buch. In der Applikation kann man sich den Code anhand von Beispielen anschauen. „Learn Java“ enthält viele Quizes (siehe Abbildung 13) und einen *Lückencode* (siehe Abbildung 12), der auszufüllen ist. Javvy verfolgt eine

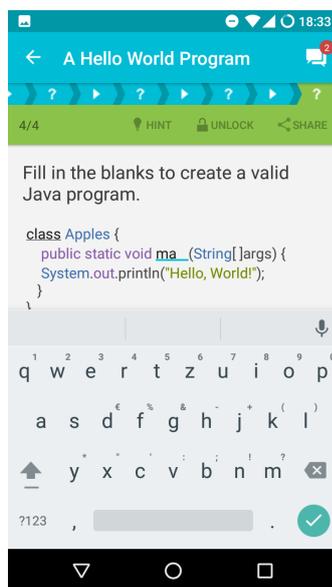


Abbildung 12: Learn Java Solo Fülltext

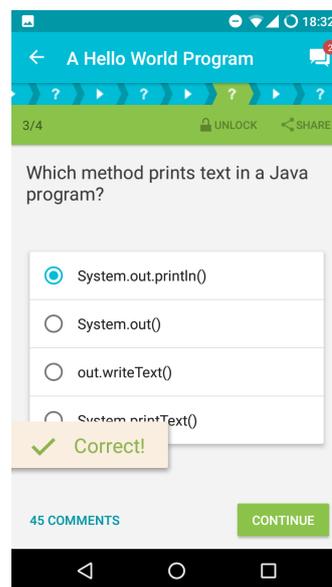


Abbildung 13: Learn Java Solo Quiz

ähnliche Methodik. Es gibt Lücken in dem Code, die ausgefüllt werden sollen. Als Beispiel soll eine String Variable initialisiert werden (siehe Abb. 14 und 15). In einigen Aufgaben muss die Antwort über die Tastatur eingegeben werden (siehe Abbildung 14), in den anderen Aufgaben können mehrere Optionen ausgewählt werden (siehe Abbildung 15). Die Applikationen sind vom Inhalt identisch wie Webanwendungen aufgebaut. Der Unterschied besteht im interaktiven Sinne darin, dass bei der mobilen Applikation versucht wird, das Tippen auf einer Tastatur zu vermeiden.

Es existieren mobile Applikationen, die für das Lernen der Programmiersprache Java geeignet sind. Sie bieten die Aufgaben in der Form eines Quizes an. Im E-Learning geht man davon aus, dass durch die Quizes das Fachwissen beigebracht werden kann. Jedoch wird in diesen Anwendungen nicht das lösungsbasierte Denken gefördert.

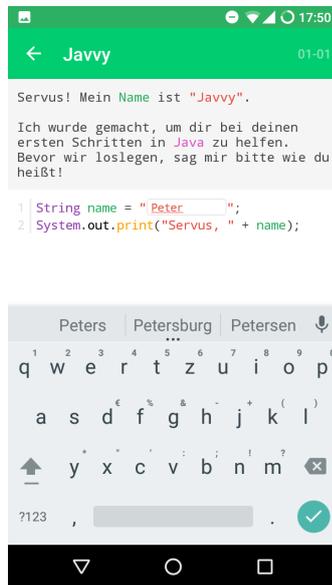


Abbildung 14: Javvy: Eingabe über Tastatur



Abbildung 15: Javvy: Auswahl des Strings

4.5 Fazit

Im Vergleich zu anderen Programmiersprachen gibt es für Java eine geringe Anzahl an E-Learning Umgebungen. Einige dieser Umgebungen sind älter und ihre Form ist meistens simpel und entspricht dem eines Buches. Positiv zu benennen ist die Lernumgebung von Codecademy. Sie gehört zu den neueren Plattformen und bietet ein modernes Design an. Die Interaktivität und die Aufgabenstellung verfolgen das „Learning-by-Doing“ Prinzip. Die Aufgaben sind einfach gestellt und auch für relativ unerfahrene Programmierer lösbar.

5 Konzept

Dieses Kapitel geht auf das allgemeine Konzept des Java Programmier-E-Learningkurses ein. Die Erkenntnisse aus den vorherigen zwei Kapiteln „Verwandte Arbeiten“ und „E-Learningplattformen“ werden erläutert.

5.1 Einleitung

Bei der Erstellung des Konzeptes werden, die Analyse über bestehende E-Learningplattformen, sowie konstruktive Erkenntnisse aus den oben genannten Publikationen (siehe S.11) aufgenommen.

Eine der wichtigsten Erkenntnisse bei den verwandten Arbeiten (siehe S.11) ist das selbstständige Praktizieren des Programmierens. Demzufolge muss der zu entwickelnde Kurs aufgaben-orientiert ("aufgaben-orientierte Didaktik" siehe [Arn13] S.40) konzipiert werden, sodass den Teilnehmern nicht nur Theorie vermittelt wird, sondern auch die Möglichkeit gegeben ist, den Code selbst auszuprobieren.

Für den Aufbau des Konzeptes ist die Orientierung an bewährten Modellen von Bedeutung. Das praxisorientierte und didaktische Modell, wie das "Extreme Apprentice Ship" fließt mit in die Konzeption des E-Learningkurses ein und wird im späteren Abschnitt 5.6 genauer beschrieben.

Aus den bereits dargelegten Informationen der bestehenden E-Learning Plattformen (Kapitel 4 S.14) können viele positive Aspekte für das Konzept übernommen werden. Aus der Analyse lässt sich feststellen, dass die Onlinekurse populär sind und keinen zusätzlichen Aufwand für die Installation auf dem PC benötigen. Der Benutzer kann den Kurs sofort über den Browser starten. Aus diesem Grund fällt die Entscheidung auf einen Online-E-Learning Kurs.

Ein weiterer Aspekt legt den Fokus bei der Aufgabenstellung auf die Formulierung der Informationen (theoretischer Teil) für Programmieranfänger. Die Aufgaben sollen weitestgehend verständlich und einfach gestellt werden. Die Texte und Informationen müssen gut und möglichst einfach formuliert sein. Zudem soll der Text sich auf das Nötigste beschränken, um einen höheren praktischen Anteil anzubieten. Die didaktische Gestaltung von Beispielen, Aufgaben und Informationen soll durch ihre Kombinationen verbessert werden.

Die analysierten E-Learning Plattformen beschäftigen sich wenig mit fehlerbehafteten Code und deren Compilermeldung. In der zu entwickelnden Anwendung soll dieses Defizit behoben werden.

Wie auch interaktive Onlinekurse, wird in der zu entwickelnden Applikation eine Programmierumgebung mit einem Code Editor und einem Ausgabeelement verwendet. Eine zeitaufwändige Registrierung und Anmeldung sollte keine Pflicht sein. Um mehr Multimediaelemente in den Kurs einzubringen soll es zusätzlich eine Ausgabe über Bilder geben.

Viele Kurse, wie beispielsweise Codecademy, sind auf Englisch. Um keine sprachliche Barriere einer Fremdsprache zu haben wird der Kurs in Deutsch entwickelt.

Die Anwendung soll auf eine Weise konzipiert werden, dass der Inhalt andauernd verbessert und geändert werden kann. Ein Argument dafür ist die Verwendung der Lernumgebung im universitären Bereich. Die Zielgruppe (Erstsemester) der Anwendung ist jederzeit verfügbar, sodass ein von ihnen laufendes Feedback erhalten und entsprechend zeitnah zu Verbesserungsmaßnahmen eingesetzt werden kann.

5.2 Entwicklung

Folgende Schritte [Hoh02] werden für die Erstellung einer E-Learning Plattform empfohlen:

- Lernlösung (Vision)
- Inhalt
- Gestaltung (Präsentation des Inhaltes) / Design
- Umsetzung
- Steuerung von Lernprozessen
- Evaluation von Lernprozessen

Die Lernlösung, der Inhalt, die Gestaltung und die Steuerung von Lernprozessen werden in diesem Kapitel besprochen. Die Beschreibung der Umsetzung wird realisiert in den Kapiteln "Entwurf" und "Implementation". Die Besprechung der Evaluation von Lernprozessen folgt im Kapitel "Auswertung".

5.3 Lernlösung (Vision)

Die E-Learninganwendung soll Benutzern interaktiv, einerseits Grundkonzepte von der Programmiersprache Java beibringen und zu dem soll ein Verständnis für das Lösen von Aufgaben vermitteln. Die Zielgruppe des Kurses sind Anfänger und Personen, welche bisher noch keine Erfahrung in Programmieren gesammelt haben.

Die Anwendung soll Studium-interessierten einen Einblick geben, wie der praktische Anteil eines Informatikstudiums aussieht. Es soll ihnen der Kern der Informatik, nämlich das Lösen von Problemen mittels Programmiersprachen, gezeigt werden.

Fortgeschrittene oder programmiererfahrene Studienanfänger können den

Kurs zur Vorbereitung nehmen und gezielt Kapitel und Übungen auswählen, wo sie nach ihrem Gefühl den Lernstoff nachholen müssen. Der Benutzer kann auf jede Übung oder Kapitel zugreifen. Er muss keine vorherige Übung bestanden haben.

5.4 Inhalt

Der Inhalt und Aufbau eines Einsteigerkurses ist stark abhängig von der Programmiersprache, die benutzt werden soll. Es existieren objekt-orientierten Sprachen, wie beispielsweise C++ und Java, sowie funktionelle und deklarative Sprachen, wie Haskell, Perl und Python. Eine Programmiersprache ist lediglich ein Werkzeug um ein Problem zu lösen. Deshalb ist die Wahl einer guten Programmiersprache unabdingbar ([MR05] S.5). Die Lernumgebung, welche für Java Programmieranfänger entwickelt werden soll, muss sich an den Konzepten der Programmiersprache orientieren.

Die fundamentalen Elemente einer prozeduralen Programmiersprache sind:

- Ausgabe
- Variablen und Zuweisungen
- Datentypen
- Kontrollanweisungen
- Schleifen
- Datenstrukturen

Wenn diese fundamentalen Elemente didaktisch beigebracht worden sind, können weitere Aufgaben zu komplexeren Problemen erstellt werden.

Um die Programmiersprache Java anfangs zu lernen, muss erklärt werden, wie eine Klasse und eine Methode (siehe einfache Klasse von Quellcode 1) aufgebaut sind. Eine Erklärung ist notwendig, weil der ausgeführte Code immer in der main-Methode startet. Das stark objekt-orientierte Konzept wird anfangs nicht erklärt, weil es zu viele Informationen wären, die erst später verwendet werden können, und anfangs den Lernenden überfordern. Das Konzept der Objekt-Orientierung kann in einem späteren Kapitel aufgegriffen werden.

Es wird anfangs die Existenz der Klassen und Methoden erwähnt. Das Ziel ist es, so schnell wie möglich einen ausführbaren und gültigen Code zu produzieren.

Quellcode 1: Basisklasse mit main-Methode

```
public class MeineKlasse
{
    public static void main (String[] args)
    {
    }
}
```

In der Lernumgebung gibt es zu jeder Übung Informations- und Aufgabentexte zum jeweiligen Thema. Es wird versucht, den Text möglichst zu reduzieren. Die Form und der Inhalt des Textes sollen kurz und verständlich sein.

5.5 Sprache

Die allgemeine Sprache soll einfach gehalten werden. Es gibt viele englische Begriffe, die in der Programmiersprache verwendet werden. Es wird versucht die Begriffe auf Deutsch zu halten und eine schulische Terminologie zu verwenden.

Tabelle 1: Fachbegriffe und einfache Begriffe

Fachbegriff	Einfacher Begriff(e)
Char	Zeichen (Ein Buchstabe etc)
String	Text
Integer	Ganze Zahl
Float	Kommazahl
Boolean	Wahr- oder Falsch
Array	Liste

5.6 Beispiele

In den Übungen werden viele Beispiele gezeigt. Diese Beispiele besitzen einen Code und können anschließend ausgeführt werden. Sie können geändert werden. Das entspricht einem E-Learning Konzept des "Extreme Apprenticeship" - Models [VPL11] und es wird der "Worked Example Effect" (siehe S. 7 Kapitel 2.6.3) gefördert. Das bedeutet, die Lernenden werden nicht überlastet einen Lösungsweg zu einem Thema, von dem sie bisher nichts wissen, zu finden. Die Lernenden lernen von den erfahrenen Programmierern. Durch viele Beispiele erkennt der Lernende eine Logik und Systematik der Programme.

5.7 Aufgaben

Bevor eine Aufgabe gestellt wird, wird ein Beispiel zu dem entsprechenden Thema gezeigt. Um die Aufgabe zu lösen, muss der Lernende, ausgehend von dem Beispiel und den vorigen Informationen, eine Transferleistung vollbringen. Dabei kann der Schwierigkeitsgrad unterschiedlich sein. In diesem Konzept soll der Schwierigkeitsgrad für die anfänglichen Aufgaben, für die fundamentalen Elemente der Programmiersprache Java, leicht bis leicht anspruchsvoll gehalten werden. Dadurch fällt das Lösen der Aufgaben leichter und gleichzeitig ist die Frustrationsgrenze herunter gesetzt. Die Aufgaben beziehungsweise Lernziele sind in kleinen Schritten unterteilt. Eine Übung besteht aus mehreren kleinen Aufgaben. Kleine Schritte führen dazu, dass der Lernende sich über den Fortschritt des Lernens die ganze Zeit bewusst ist ([VPL11] S.95). Durch mehr Erfolgserlebnisse sind die Lernenden motivierter den Kurs weiter zu machen.

Wenn der Lernende nicht weiter kommt, wird ihm angeboten eine Musterlösung anzeigen zu lassen. Durch die Vorgabe von geschriebenem Code für die Aufgabe, erspart sich der Benutzer die Zeit den Code zu tippen. In dem Fall wird sich auf das wesentliche konzentriert, die Aufgabe zu lösen. Alle Online E-Learning Plattformen enthalten keine spezifischen Aufgaben zu den Fehlern, die beim Schreiben des Codes entstehen können. Für dieses Konzept werden Aufgaben entwickelt, die einen Fehler enthalten und nicht kompilierbar sind. Der Lernende muss den Fehler im Code korrigieren. Das Konzept soll das kontextuelle Wissen (siehe S.7 Kapitel 2.7) bereichern. Das bedeutet, die Situation, bei der man einen Fehler macht, taucht immer wieder auf. Deshalb macht es Sinn, die Prozedur der Fehlererkennung zu erlernen. Eine mögliche Aufgabe ist in dem Textfeld 2 zu sehen.

Textfeld 2: Aufgabe zur Fehlerkorrektur

Irgendetwas fehlt in der Methode. Korrigiere alle Fehler, damit der Code funktioniert.

Quellcode 2: Unvollständiger / Fehlerhafter Code

```
public class MeineKlasse{  
  
    public static void main (String[] args)  
    {  
        System.out.println("Hello");  
        System.out.println("World!");  
    }  
}
```

In diesem Code (siehe Quellcode 2) fehlt ein Semikolon nach der *System.out.println()* Anweisung. Ein expliziter Hinweis und diese Aufgabe sollen Lernende daran erinnern, bei der Fehlersuche, als erstes darauf zu achten, ein Semikolon nach einer Anweisung zu schreiben. Neben dem Semikolon ist auch die richtige Klammerung sehr wichtig. Das folgende Beispiel (siehe Quellcode 3) zeigt einen Code, der eine fehlende Klammer besitzt. Diesen Code sollen die Benutzer korrigieren.

Quellcode 3: Unvollständiger Code

```
public class MeineKlasse{  
  
    public static void main (String[] args)  
    {  
        // Klammer fehlt  
    }  
}
```

Wenn die Lernenden mit diesen Fehlern vertraut werden, führt die Fehlerfindung schneller zum Erfolg.

Der Einsatz von Multimediaelementen bei den Aufgaben soll für mehr Abwechslung sorgen. Gedacht wird dabei an eine Bildausgabe, anstatt der Ausgabe über die Konsole. Einerseits kann ein Bild editiert werden. Eine vorstellbare Aufgabe wäre, die Farbe in einem Bild zu ersetzen. Es lässt sich damit didaktisch und visuell die Kontrollanweisung *if* erklären.

Eine andere Aufgabe mit einer Bildausgabe, wäre beispielsweise eine Funktion ein Bild mit Formen selbst zu erstellen. Es soll eine *putPixel(x,y)* Funktion zu diesem Zweck benutzt werden. Man kann durch mehrere Pixel eine Linie zeichnen und das kann effizient mit einer Schleife gemacht werden. Es wird dadurch didaktisch und visuell die Wirkung und Funktion einer Schleife erklärt.

Die Aufgaben erfolgen nach einer deduktiven und einer induktiven Lernstrategie. Einerseits müssen einige Regeln und Beispiele nach der deduktiven Lernstrategie gezeigt werden. Andererseits sieht die induktive Lernstrategie vor, dass der Lernende sich durch seine eigene Transferleistung

die Regeln eigenständig erschließen kann. Das folgende Beispiel (siehe Textfeld 3) soll diesen Ansatz erläutern. Einige Details über die ganze Zahl

Textfeld 3: Aufgabe mit Integerzahlen in Java

Die ganze Zahl wird mit dem Typ *int* dargestellt. Man spricht auch von dem Datentyp Integer. Die Zuweisung einer Zahl erfolgt immer rechts von der zugehörigen Variable. Die Variable *nummer* wird mit der ganzen Zahl 123 initialisiert : `int nummer = 123;`

werden nicht erläutert, wie beispielsweise der Wertebereich, welcher durch die Größe von 32 Bit definiert ist. Dieser Ansatz soll eine Überflutung von Informationen vermeiden. Er soll dem Pragmatismus des Programmierens folgen.

5.8 Schritt für Schritt in der Übung

Ein klassisches didaktisches Konzept eines Unterrichts besteht darin, dass ein Beispiel immer dem theoretischen Teil folgt. Die Theorie wird somit durch eine Aufgabe erlernt. Dieser Sachverhalt wird in der Abbildung 16 verdeutlicht.



Abbildung 16: Theorie - Beispiel - Aufgabe

Viele E-Learningplattformen, wie beispielsweise Codecademy, setzen dieses Prinzip ein (siehe Abb.17). Es gibt anfangs einen großen Informationsblock und anschließend folgen mehrere Aufgaben beziehungsweise "Anweisungen". Diese Prinzipien sollen in dem Konzept verwendet werden, wenn es angebracht ist.

Das klassische Prinzip kann durch Zwischenschritte erweitert werden. Als Beispiel kann der Informationsblock aufgeteilt werden. Wie die Abbildung 18 zeigt, wird eine Lerneinheit durch eine Reihenfolge von Lernelementen dargestellt. Zuerst gibt es einen Teil der Informationen zu der Aufgabe und im darauf folgenden Schritt kommt ein neuer Teil der Information für



Abbildung 17: Nach der Information folgen die Aufgaben.

die zweite Aufgabe. Auf diese Weise wird Praxis und Theorie, Schritt-für-Schritt, miteinander kombiniert. Das Modell der Übungsgestaltung in der



Abbildung 18: Info - Aufgabe - Info - Aufgabe

Abbildung 19 erhöht den Erkenntnisgewinn (einen "Aha"-Moment). Beginnend mit einem Einleitungstext folgt im Anschluss die Aufgabe. Folgende

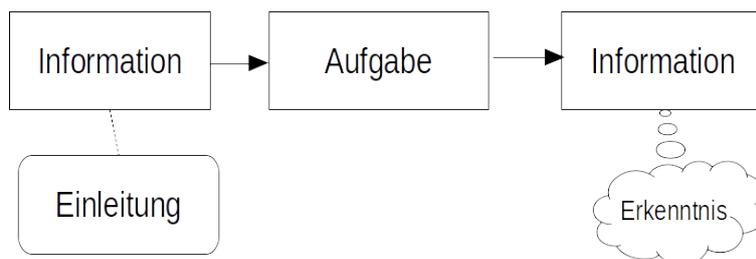


Abbildung 19: Info - Aufgabe - Information (Erkenntnis)

Aufgabe (Textfeld 4) und der dazugehörige Code (siehe Quellcode 4) soll dieses Prinzip verdeutlichen. Und Erkenntniseffekt kommt durch die neue Information (siehe Textfeld 5)

Textfeld 4: Aufgabe mit String

Ersetze das + durch -. Führe den Code aus. Fällt dir etwas an den Ergebnissen auf ?

Quellcode 4: Unvollständiger Code

```
public class MeineKlasse
{
    public static void main (String[] args)
    {
        String a = "Hallo ";
        String b = "Peter";
        String s = a + b;
        System.out.println(s);
    }
}
```

Textfeld 5: Aufgabe mit Integerzahlen in Java

Minus funktioniert nicht bei Strings. Denn der Plus-Operator führt keine richtige Addition aus, sondern lediglich das Zusammenfügen von Variablen.

5.9 Design von Entwicklungsumgebung

Es gibt viele Publikationen, wie [HSSC05] und [Wis06], die sich mit der geeigneten Entwicklungsumgebung für Programmieranfänger beschäftigen. In den Publikationen wird empfohlen, eine leichtgewichtige Entwicklungsumgebung für Programmieranfänger zu benutzen. Es gibt in diesem Segment z.B. BlueJ [Blu17] und DrJava [Dr 17]. Im Gegensatz zu professionellen Tools (wie Eclipse [Ecl17]) sind die Menüs und die Benutzerschnittstelle reduziert.

Die Tendenz der Publikationen und der Tutoren von Universitäten geht dahin, dass die Lernumgebung für Programmieranfänger minimalistisch sein soll. Aus diesem Grund wird für das Konzept eine minimalistische Programmierumgebung entwickelt. Die folgenden Punkte sollte die Entwicklungsumgebung besitzen:

1. Code Editor
2. Ausgabeelement

Hinzu kommt das Element der Anzeige der Informationen und der Aufgaben. Die Einteilung dieser Elemente soll auf einer Seite erfolgen. Dadurch wird der „Split-Attention-Effekt“ (siehe S.7 Kapitel 2.6.3) vermieden. Die

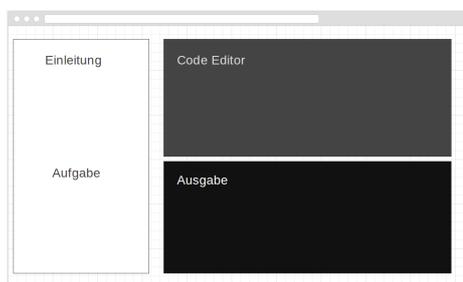


Abbildung 20: Mockup der Webpage

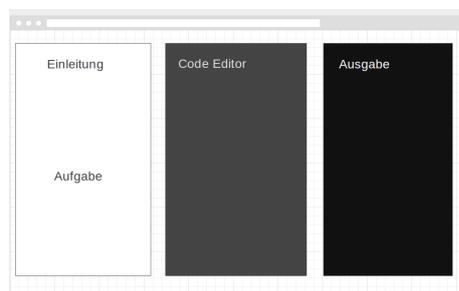


Abbildung 21: Zweites Mockup der Webpage

Einteilung der Fenster des ersten Mockups (siehe Abb. 20) entspricht der gängiger Editoren von Entwicklungsumgebungen wie Eclipse, Netbeans oder IntelliJ [Int17]. Analog zu ihnen erscheint links das Navigationsmenü der Editoren und rechts ist der Codeeditor und die Ausgabekonsolle in zwei Hälften unterteilt. Die Einteilung der Fenster des zweiten Mockups (siehe Abb.21) entspricht dem Layout von Codecademy (siehe S.20).

Die Einteilung hat den Vorteil, dass der Benutzer drei lineare gleich große Fenster hat. Das führt zu einer besseren Übersichtlichkeit. Die folgenden Arbeitsschritte sind dadurch definiert: Der Benutzer liest zunächst links den Aufgaben- und den Einleitungstext. Anschließend geht er weiter nach rechts zum Codeeditor und editiert oder erstellt den Code. Die Ausgabe

wird rechts, nach der Ausführung des Programms, angezeigt. Dies ist eine einfache und logische Schrittfolge von links nach rechts, die unterbewusst wahrgenommen wird.

5.10 Förderung der Motivation der Benutzer

Rückmeldung an dem Lernenden über seinen Fortschritt soll die Kursteilnehmer anreizen ([Ker01] S.204) und die Motivation auf einem hohen Niveau halten. Die Lernumgebung soll eine sofortige Rückmeldung darüber geben, ob die Aufgabe bestanden ist oder nicht. Dadurch erkennt der Lernende seinen eigenen Fortschritt, der in einer Prozentangabe angezeigt wird.

5.11 Lernziele

Im Grundlagenkapitel dieser Arbeit wurde das Modell von Herrn Bloom (siehe S.8 Abschnitt 2.7.1) vorgestellt. Die höchste Ebene, welche man in der Programmierung erreichen will, ist die Stufe der Kreativität. Diese Stufe könnte man realisieren, wenn man den Teilnehmern ein Projekt anbietet und ihnen ihren kreativen Freiraum lässt.

Der Kurs soll für Anfänger geeignet sein. Bei der Erklärung von fundamentalen Inhalten der Java Entwicklung bleibt relativ weniger kreativer Freiraum übrig. Da die Erstellung und die Überprüfung eines freien komplexen Projekts über eine Maschine erschwert wird. Die Teilnehmer des Kurses sollen das Lernziel nach Bloom, des "Anwendens" und der "Analyse", erreichen. Nach dem Kurs können die Teilnehmer programmieren (*anwenden*) und kleine Probleme lösen (*analyse*).

6 Entwurf

Dieses Kapitel geht auf den allgemeinen Entwurf der Anwendung ein. Durch verschiedene Diagramme wird der Zusammenhang und das Zusammenspiel der einzelnen Komponenten erklärt.

6.1 Anwendungsfalldiagramm

Mit dem UseCase Diagramm (siehe Abb. 22) lassen sich die Funktionen aller Stakeholder erklären. Der Kursteilnehmer ist der zentrale Benutzer in diesem Projekt.



Abbildung 22: Anwendungsfalldiagramm

Der Benutzer kann sich registrieren und sich anschließend einloggen. Er kann die theoretischen Informationen und die Teilaufgaben in der Übung lesen. Der Code kann vom Benutzer editiert und auf dem System "Java Lernumgebung" ausgeführt werden. Zudem kann er gespeichert werden, wenn die Login- oder die anonyme Session erstellt wurde. Der Administrator kann den Inhalt des Kurses ändern. Er kann neue Kapitel und Aufgaben erstellen, sowie bestehende Inhalte löschen oder ändern.

6.2 Anforderungen

Aus dem Konzept und der Vision (siehe 25 Abschnitt 5.3) lassen sich folgende Anforderungen erstellen:

6.2.1 Administrator

1. Ein Administrator kann einen neuen Inhalt erstellen.
2. Ein Administrator kann den Inhalt editieren und speichern.
3. Ein Administrator kann den Inhalt löschen.

6.2.2 Navigation und Übersicht

1. Der Benutzer kann in der Anwendung navigieren.
2. Der Benutzer kann jederzeit seinen Lernstatus sehen.

6.2.3 Gestaltung

1. Es gibt einen Bereich mit einem Einleitungstext.
2. Es gibt einen Bereich für den Aufgabentext.
3. Die Aufgabe kann in mehrere kleine Abschnitte unterteilt sein.
4. Es gibt einen Editor.
5. Es gibt ein Ausgabefenster.
6. Einleitung, Aufgabe, Codeeditor und Ausgabe befinden sich auf einer Webseite.
7. Der Einleitungstext ist ein- und ausklappbar.

6.2.4 Benutzer

1. Der Benutzer muss sich nicht anmelden.
2. Der Benutzer kann sich mit einem Benutzernamen und einem Passwort registrieren.
3. Der Benutzer kann seine persönlichen Daten ändern.
4. Der Benutzer kann die "Passwort vergessen"-Funktionen nutzen.
5. Der Benutzer kann sich mit einem Benutzernamen und einem Passwort anmelden.

6.2.5 Benutzer und Aufgabe

1. Der Benutzer bekommt eine Aufgabe gestellt.
2. Der Benutzer gibt seine Lösung in den Codeeditor ein.
3. Der Code kann geändert werden.
4. Der Benutzer muss den Code ausführen können.
5. Die Ausgabe muss angezeigt werden können.
6. Der Benutzer kann auf jedes Kapitel oder Unterkapitel frei zugreifen.
7. Der Benutzer muss keine Aufgabe gelöst haben, um auf eine andere Aufgabe zuzugreifen.
8. Die Teilaufgaben müssen nacheinander gelöst werden, wenn sie innerhalb einer Aufgabe in einem logischen Zusammenhang stehen.

6.2.6 Nicht Funktionale Anforderungen

1. Die Webanwendung läuft in gängigen Browsern.
2. Die Webanwendung wird auf einem Server installiert.

6.3 Architektur

6.3.1 Server und Client

Eine Webapplikation kann server-seitig einen Teil der Informationen berechnen. Das kostet Ressourcen des Servers. Heutzutage ist es modern einen Teil der Webapplikation, welcher Rechenleistung benötigt, auf der Client-seite im Webbrowser auszulagern. Es wird dabei vom einem "Big Client" gesprochen. Ein großer Teil der Daten wird auf dem Client-System gespeichert. Der Server stellt eine REST-Schnittstelle zur Verfügung, worüber der Client die Informationen aufrufen kann. Die Daten sind direkt auf dem Client verfügbar, sodass ein sofortiger Zugriff möglich ist. Als Folge dessen, reagiert die grafische Benutzerschnittstelle (GUI) schneller. Dadurch lässt sich eine schnellere Interaktion ausführen, was für eine bessere Benutzbarkeit der Anwendung förderlich ist. Zudem bietet die REST-Schnittstelle in Aussicht die Erstellung einer mobile Applikationen. In dem Diagramm 23

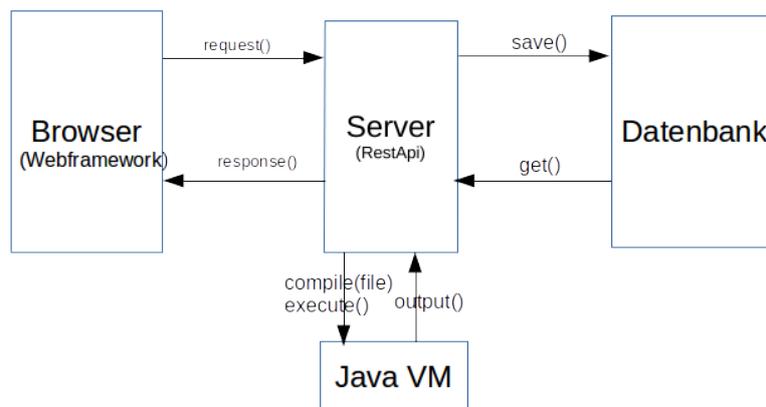


Abbildung 23: ER Diagramm Serverseitig

wird zu der gängigen 3-Schichten Architektur [3-S17] der physikalische Teil der abgeschlossenen virtuellen Maschine von Java genommen. In der klassischen 3-Schichten Architektur sind die Schichten :

- Präsentationsschicht (Browser)
- Logik- / Anwendungsschicht (hier im Webframework und im Server)

- Datenerhaltungsschicht (hier Datenbank)

Die Präsentationsschicht ist physisch im Client vorhanden. Hierzu soll ein Webframework benutzt werden, das die Benutzerschnittstelle im Browser erstellt. Die Anwendungsschicht ist zu einem Teil in der Browseranwendung und zum anderen auf der Serveranwendung vorhanden. Anschließend ist der Server für die Persistenz der Daten in der Datenbank verantwortlich.

6.3.2 Ablauf: Server und Client

Im Diagramm (Abb. 24) wird der reguläre Programmablauf beschrieben. Der Benutzer ruft die Website auf und erstellt oder editiert den Code. Im Anschluss kann er sich das Ergebnis der Ausführung anschauen. Dies beschreibt die Kernfunktion der Anwendung, die der Server ausführt.

Im ersten Schritt (1) stellt der Client eine HTTP-Anfrage an den Server. Der Server ruft die Benutzer- sowie Kursdaten von der Datenbank im zweiten Schritt (2) auf. Dann werden die Daten an den Browserclient geschickt (Schritt 3). In Schritt (4) gibt der Benutzer den Code über den Browser ein. Der Code wird anschließend an die Serverapplikation geschickt und in der Datenbank gespeichert. Der Server kompiliert (Schritt (6)) den Java Code und führt ihn in Schritt (7) in der „Java Virtual Machine“ aus. Der ausgeführte Java Code gibt eine Rückmeldung in Form von einer Nachricht über die Ausgabe zurück. Das Ergebnis und der Code werden in der Datenbank gespeichert (Schritt (9)). Im letzten Schritt (10) wird die Konsolenausgabe von Java angezeigt, außerdem wird der Status über das Bestehen der Aufgabe dem Benutzer mitgeteilt.

6.3.3 Entity-Relationship Diagramm server-seitig

Das ER-Diagramm auf der Serverseite definiert die in der Datenbank zu speichernden Daten und deren Beziehungen. Aus dem Grund, dass der Teil der Daten clientseitig berechnet wird, ist es nicht erforderlich das Datenmodell kompliziert zu halten.

Wie in der Abbildung 25 zu sehen ist, existiert ein Kurs, der mehrere Kapitel beinhaltet. Ein Kapitel (*Chapter*) kann mehrere Sektionen (*Section*) enthalten. Diese Sektion kann mehrere Übungen (*Exercise*) besitzen. Eine Übung kann mehrere Inhalte enthalten (*Exercisestep*). Ein Inhalt kann ein Beispiel (*Example*), eine Aufgabe (*Subexercise*) oder ein Informationstext sein. Zu einer Aufgabe oder einem Beispiel gehört ein Text und ein vorbereiteter Code. Für eine Aufgabe gibt es zusätzlich eine Beispiellösung und einen Text mit der erwarteten Ausgabe. Eine Aufgabe kann auch Regeln beinhalten, welche der eingegebene Code erfüllen soll. Die Aufgabe ist erst bestanden, wenn diese Regeln erfüllt sind. Die Regeln sind in einer *CodeRule*-Klasse vorhanden. Die Informationsklasse kann einen Text und ein Bild enthalten.

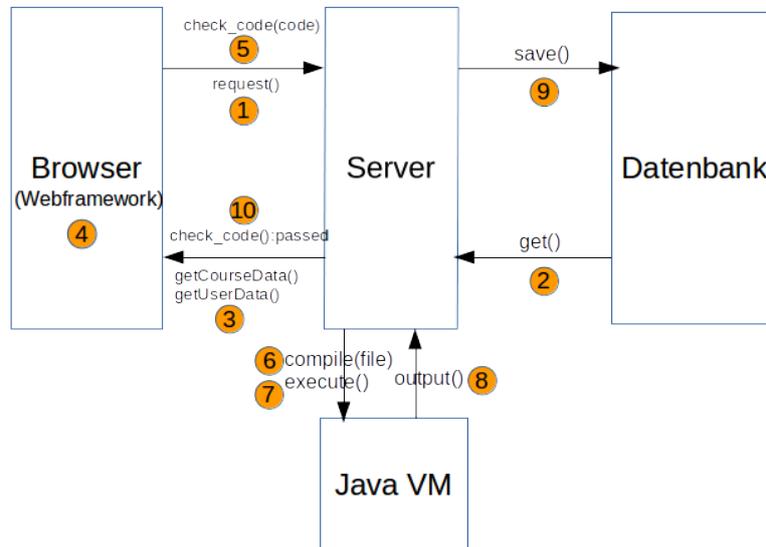


Abbildung 24: Ablauf Client- und Serverseitig

Die Datenstruktur des Kurses ist hierarchisch aufgebaut. Die Besonderheit des Datenmodells besteht darin, dass der Benutzer erzeugte Daten in dem Fall speichern kann, wenn er angemeldet ist. In der Datenbank werden die Lösungen des Benutzers und sein Status zu der Aufgabe ("bestanden" oder "nicht bestanden") gespeichert. Dies geschieht in der Klasse *UserExercise*, die eine eins zu eins Beziehung zu der *SubExercise* Klasse hat.

6.3.4 Klassendiagramm client-seitig

Ein wichtiger Bestandteil jeder interaktiven Webapplikation ist die Applikationslogik auf der Seite des Clients. Das folgende Diagramm beschreibt die Klassen auf der Clientseite.

Im Gegensatz zu dem ER-Diagramm (siehe 6.3.3) werden hier *UserExercise* und *SubExercise* zusammengefügt. Der Aufbau entspricht dem des vorher gezeigten ER-Diagrammes. Ein Kapitel kann mehrere Sektionen haben. Es gibt eine Abstrakte Klasse *AbstractLecture* die das Attribut "*passed:boolean*" enthält. Das Attribut sagt aus, ob der Benutzer das jeweilige Kapitel, die Sektion oder eine Aufgabe bestanden hat. In der Datenbank wird lediglich der Status gespeichert, ob der Benutzer eine Teilaufgabe bestanden hat. Das Bestehen eines Kapitels, einer Sektion oder der gesamten Übung wird im Client durch die Methode *calculateStatus()* berechnet. Um auf die nächste Übung zu navigieren, wird die Methode *nextExercise()* aufgerufen.

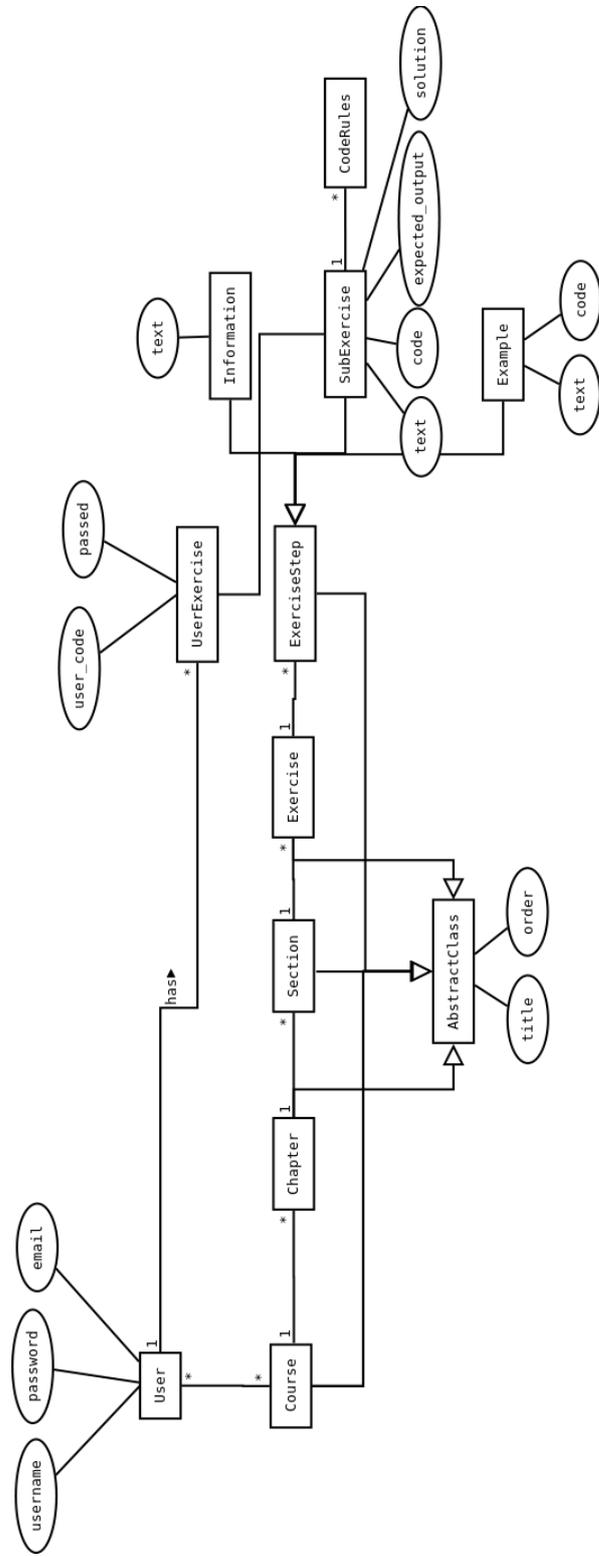


Abbildung 25: ER Diagramm Serverseitig

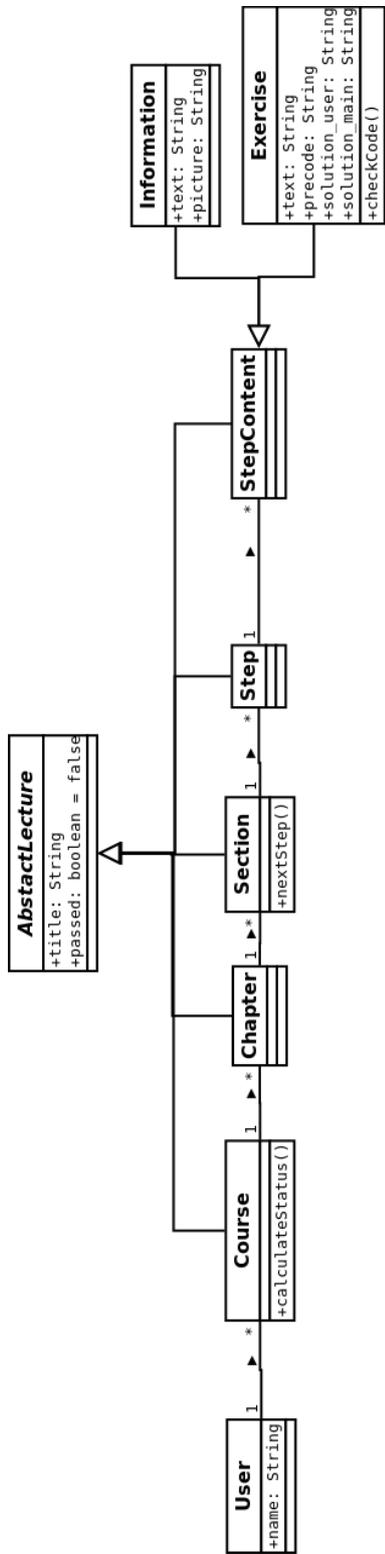


Abbildung 26: Klassen Diagramm Clientseitig

7 Implementation

7.1 Technik

Die Anwendung wird als Webanwendung entwickelt. Sie soll in den gängigen Browsern laufen. Das folgende Diagramm zeigt den Technologie Stack von der physischen Ebenen der Anwendung. Die Webanwendung wird serverseitig mit dem in der Programmiersprache Python aufgebauten Framework Django [Dja17] entwickelt. In dieser Arbeit wird Django modular durch die Library "Django Rest Framework" erweitert. Das Framework ist für die Datenkommunikation zwischen dem Client und dem Server verantwortlich. Auf der Clientseite wird HTML 5, Javascript, JQuery [JQu17], CSS und das Javascript Framework Vue-js [Vue17] eingesetzt. Auf der Datenbankschicht wurde während der Entwicklung eine SQLite Datenbank [SQL17] verwendet. Aufgrund der Tatsache, dass für eine SQLite-Datenbank keine Optimierungen vorgenommen werden können, empfiehlt sich für den produktiven Betrieb eine PostgreSQL [Pos17] Datenbank.

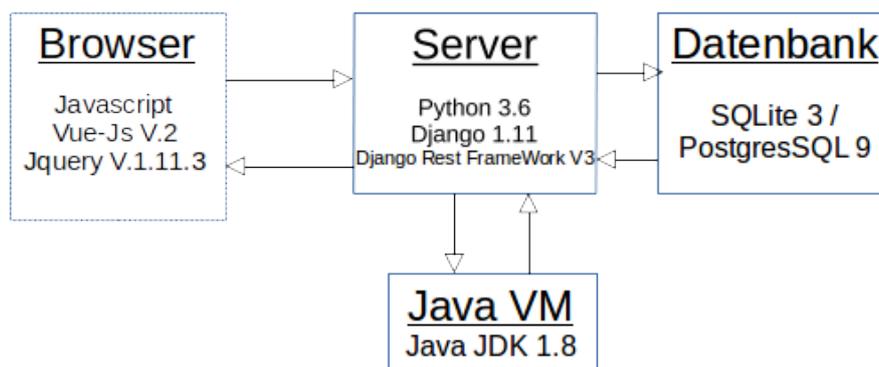


Abbildung 27: Technologie Stack

7.1.1 Django

Django ist ein, in der Programmiersprache Python geschriebenes und weit verbreitetes Webframework. Mit Django lässt sich eine prototypische und später eine produktionsfertige Anwendung angenehm entwickeln.

7.1.2 Javascript Frameworks: Vue-js vs JQuery

Die Anzahl an Javascript Frameworks ist in den letzten Jahren deutlich gestiegen. Zu den bekanntesten gehören AngularJS[Ang17], React[Rea17]

und Vue.js [Vue17].

Als Entscheidungskriterien für die Wahl eines Javascript Frameworks wurden die Anwendbarkeit und simple Erlernbarkeit des Frameworks herangezogen. Aus diesen Grund wurde für die Frontendentwicklung das Framework Vue.js ausgewählt. Vue.js ist ein schlankes und schnelles Javascript Framework, das 2014 veröffentlicht wurde. Sein Vorteil gegenüber der nativen Javascript Sprache und JQuery (ein anderes bekanntes Javascript Framework) liegt in der Datenbindung. Unter Datenbindung (eng. Databinding) versteht man das Weiterleiten der Daten zwischen den Objekten. Konkret bedeutet dies eine Möglichkeit die Daten in einem HTML-Dokument direkt in ein Objekt hinein zuschreiben. Wird das Objekt geändert, ändert sich seine Präsentation auf der HTML-Seite. Der Einsatz von Vue.js bietet im Vergleich mit JQuery den Vorteil, dass HTML Elemente nicht außerhalb der HTML Dokumente programmiert werden müssen. Demzufolge wird der Code deutlich übersichtlicher.

7.2 Server

Die Anwendung wird nach den Django Framework gegebenen Namenskonventionen für Ordner und Dateien entwickelt. Die wichtigsten Dateien und Methoden der Anwendung werden in diesem Abschnitt erläutert. In einem Django-Projekt werden die Entities in einer *models.py*-Datei definiert. Die Objekte der Klassen werden mittels eines "Object-Relation-Mapper" in der Datenbank persistent.

In der *forms.py* werden die Formulare, wie beispielsweise die Login-Form oder die Passwort-Reset-Form, eingetragen.

Die *views_rest.py*-Datei verwaltet die REST-Anfragen. Die REST-Methode *getCourse()* stellt die allgemeinen Kursdaten bereit. Sie ist über *"/rest/course/"* aufrufbar. Eine weitere Methode *getUserExercises()* liefert die benutzerspezifischen Daten zu dem Kurs. Sie ist über *"/rest/user_exercises/"* erreichbar. In der Funktion *check_code()* führt den Code des Benutzers und überprüft ihn.

In dem "templates" Ordner befinden sich html-Dokumente, die in der Templattersprache Jinja [Jin17] geschrieben sind. In der *views.py* werden diese Html Dateien mit Werten für Variablen gefüllt. Zusätzlich sind in dieser Datei, übliche Funktionen für das Registrieren des Benutzers, Einloggen (*login()*) sowie das Resetten (*password_reset()*) des Passworts und Änderung der eigenen Benutzerdaten (*change_user_data()*), enthalten.

In der *settings.py* werden alle wichtigen Einstellungen gespeichert. In der Datei *koja_java.py* sind die Funktionen, welche für das Kompilieren (*compile()*) und Ausführen (*execute()*) des Java Codes verantwortlich sind, implementiert. Die Methode *check_rules()* überprüft, ob der Code bestimmte Inhalte enthält. Durch die Auslagerung, dieser Funktion ist es auch möglich, innerhalb kurzer Zeit eine andere Programmiersprache in diese Plattform

einzubinden.

7.3 Client

In diesem Abschnitt wird der Teil der Anwendung beschrieben, der auf dem Client System, also in einem Browser, ausgeführt wird.

Die Datenstruktur des Kurses ist, wie im Klassendiagramm (siehe Abb. 26) beschrieben, hierarchisch aufgebaut. In der *koja_data.js* werden alle Objekte gespeichert.

Die *koja_course.js* enthält ein Teil der Anwendungslogik, der in den folgenden Methoden beschrieben wird. Die Methode *getCourse()* ruft die allgemeinen Kursdaten über Ajax an die REST-API des Servers auf. In der Methode *getUserExercises()* werden die benutzerspezifischen Daten, wie der gespeicherte Code und der Status ("bestanden" oder "nicht bestanden"), zum Kurs aufgerufen.

Aus den Daten berechnet die Methode *calculateSuccessRate()* die prozentuale Erfolgsrate des Benutzers. Während einer Übung, wird der nächste Schritt erst angezeigt, wenn die vorherige Teilaufgabe bestanden ist. Die Methode *computePassedSteps()* überprüft, ob diese Teilaufgabe bestanden ist und setzt den Wert für den nächsten Schritt auf "wahr", um ihn anzeigen zu lassen.

Die *koja_exercise.js* enthält die GUI-Funktionen für die Übung. Die Methode *checkCode()* schickt den eingegebenen Code des Benutzers über Ajax an den Server. Der Server kompiliert den Code und führt ihn aus. Es kommt eine Nachricht zurück, ob der Benutzer die Teilaufgabe bestanden hat oder nicht. Über die Methode *showSolution()* kann man sich die Lösung der Teilaufgabe anzeigen lassen. Über die Methode *resetCode()* wird die Teilaufgabe auf den Anfangszustand gesetzt. Die Methode *showNextStep()* zeigt die nächste Teilaufgabe der Übung an. Hat der Benutzer die Information gelesen muss die Serverdatenbank synchronisiert werden, dass er diesen Schritt der Übung bestanden hat. Dies geschieht über die Methode *updateExerciseStep()*. Der Code wird automatisch über diese Methode *updateSubExercise()* auf der Server Datenbank aktualisiert.

7.4 Überblick der wichtigsten Funktionen

In der Abbildung 28 werden die wichtigsten Funktionen vom Server und dem Client gezeigt. Die wichtigsten Elemente der Benutzerschnittstelle werden dargestellt. Dazu gehören die Buttons *Reset*, *Lösung* und *Run*. In den vorherigen zwei Abschnitten wurden die Funktionen des Servers und des Clients ausführlich erläutert. Abbildung 28 soll den zentralen Kern der Webanwendung verständlich darstellen.

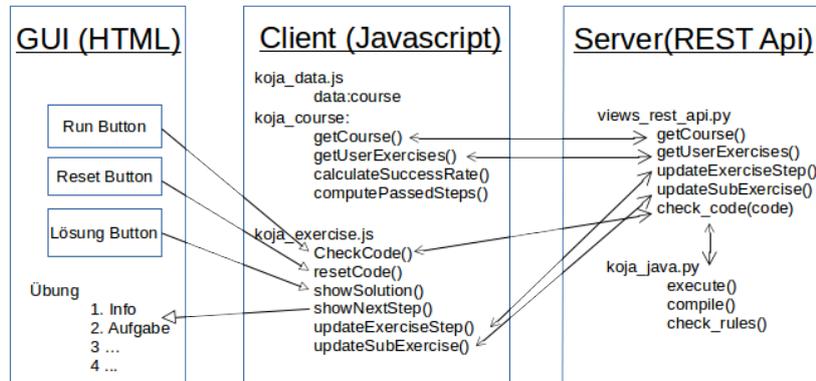


Abbildung 28: Funktionen von Server und Client

Kompilierung und Ausführung des Java Codes

Eine einzelne Klasse kann folgendermaßen kompiliert werden:

```
javac MeineKlasse.java
```

Ihre Ausführung folgt analog:

```
java MeineKlasse
```

Werden in einer Klasse andere Klassen oder Objekte referenziert, sollen sie über den Classpath bei der Kompilierung hinzugefügt werden.

```
javac javac -classpath /pfad/andere/klasse MeineKlasse.java
```

Erfolgt die Ausführung nicht in dem Ordner, wo sich die Java Datei befindet muss auch der der Pfad über dem Classpath angegeben werden.

```
java -classpath /pfad/meine/klasse MeineKlasse
```

7.4.1 Bilder erstellen

In der Aufgabe, bei der ein Bild erstellt werden soll, wird eine Funktion benötigt, die einen einzigen Pixel im Bild zeichnen kann. Zu diesen Zweck wird die Graphics2D [Gra17] Klasse in Java benutzt. Die Klasse kann einfache 2-dimensionale Zeichnungen in ein Bild oder in einer Desktopanwendung erstellen. Die Klasse stellt jedoch keine Pixelfunktion direkt zur Verfügung. Indirekt kann man einen Pixel über die Methode beschreiben.

```
g.drawLine(x, y, x, y);
```

Man zeichnet einfach eine Linie von einem Punkt zu einem gleichen Punkt. Aus didaktischen Gründen muss eine *drawPixel(x,y)* Funktion dem Benutzer zur Verfügung gestellt werden. Die Klasse Graphics2D wurde deshalb um die *drawPixel(y,x)* erweitert (siehe Quellcode 5)

Quellcode 5: Erweiterung der drawPixel-Funktion

```
public class Graphics2DE extends Graphics2D {  
  
    Graphics2D g;  
  
    public Graphics2DE(Graphics2D g) {this.g = g;}  
  
    public void drawPixel(int x, int y)  
    {  
        this.drawLine(x, y, x, y);  
    }  
    ...  
}
```

Die Pixelfunktion wird in einer Aufgabe für while-Schleifen verwendet. Der Kursteilnehmer soll in der Aufgabe eine Linie zeichnen. Die while-Schleife lässt sich damit visuell und interessanter erklären.

7.5 Sicherheit

Um eine Ausführung vom schädlichen Code auf dem Server zu vermeiden, wurden zwei Maßnahmen für den Prototyp implementiert.

Der erstellte Betriebssystem-Benutzer, der den Django Server startet, hat keine Root-Rechte und ist damit ein unprivilegierter Benutzer. Dieser Benutzer hat viele Einschränkungen und kann nicht aktiv in das Serversystem eingreifen.

Die zweite Maßnahme ist die Beschränkung in der Ausführung von bestimmten Methoden in Java. Der normale Aufruf des Java Programms ist:

```
java MeineKlasse
```

Für die Einschränkungen seitens Java, werden bei der Ausführung sogenannte Policies verwendet. Eine Policy-Datei beinhaltet die Zulassungsberechtigungen, auf welche Klassen und Methoden das Programm zugreifen darf. Das nächste minimale Beispiel zeigt die Zugriffserlaubnis auf das Dateisystem mit dem Ordner "tmp".

```
grant {  
permission java.io.FilePermission "/tmp/*", "read,write";  
};
```

Da die Anwendung keine speziellen Berechtigungen braucht, ist die Policy-Datei leer:

```
grant {};
```

Konkret bedeutet das, dass die Ausführung des Java Programms auf die Ein- und Ausgabe beschränkt ist. Die Ausführung des Java Programms mit der Policy erfolgt über die Konsole mit dem folgendem Befehl, wobei die *base.policy* die oben genannte Datei ist:

```
java -Djava.security.manager -Djava.security.policy==base.policy MeineKlasse
```

Bilder und Sicherheit

Es stellt sich die Frage, wie das Java Programm Bilder erstellen und aufrufen kann. Die Antwort ist recht einfach. Die Bilder werden als base64-codierten String übergeben und ausgegeben. Wenn man ein Java Programm ausführt, kann man der *main(String[] args)* mehrere String-Argumente übergeben.

```
java MeineKlasse test test2
```

Es wird ein Bild als String übergeben.

```
java MeineKlasse iVBORw0KGgoAAAANSUhEUg...
```

Das exemplarische Bild aus dem Kurs hat eine 7528 Zeichen. Es hat eine Auflösung von 800 mal 600 Pixeln. Die zusätzliche base64-Kodierung des

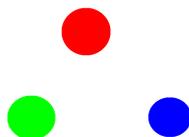


Abbildung 29: Bild mit den drei Grundfarben

Bildes kostet mehr Ressourcen, jedoch wird die Erfüllung der Sicherheitsanforderung dadurch gewährleistet. Das Java Programm darf nicht auf die Festplatte zugreifen oder einen anderen Code ausführen der schädlich sein könnte.

7.6 Content Erstellung während Entwicklung

Die elementare Basis bei der Erstellung des Contents wurde durch die Verwendung einer Yaml Datei realisiert. Das Format ist menschenlesbar. Der Vorteil dieses Dateiformats ist die schnelle Erstellung und die Sicherung von Aufgaben. Wenn die Daten auf der Datenbank gelöscht werden, lassen sie sich innerhalb weniger Sekunden wieder neu aufspielen. Der Administrator und der Entwickler sollten diesen Aspekt auf jeden Fall beachten. Ein Ausschnitt aus besagter Datei befindet sich im Anhang (siehe S.63).

7.7 Installation der Anwendung

Eine ausführliche Anleitung zur der Installationen auf einem Linux-Server befindet sich Anhang (siehe S.64).

8 Anwendung und Umsetzung

Der E-Learning Kurs soll ergonomisch selbsterklärend sein. Dieses Kapitel soll den Aufbau aus der Sicht des Benutzers und aus der Sicht des Administrators zeigen.

8.1 Benutzer

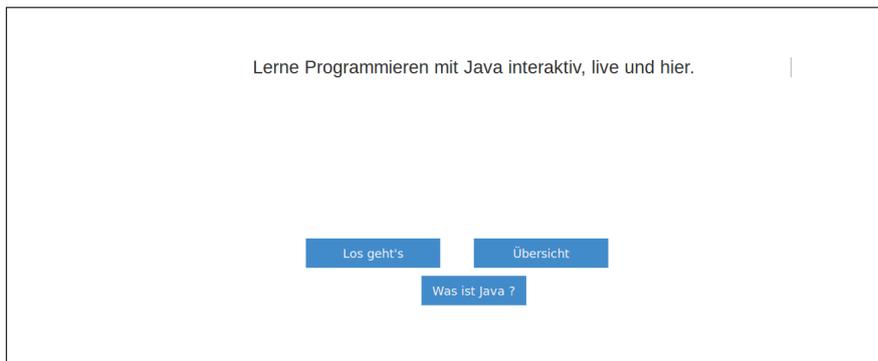


Abbildung 30: Startseite der Lernanwendung

Auf der Startseite der Lernumgebung kann man zur ersten Übung über den "Los geht's" Button gelangen. Möchte man den ganzen Kurs überblicken, erreicht man dies über den "Übersicht"-Button. Die Struktur des Kur-



Abbildung 31: Übersichtsseite der Lernanwendung

ses ist hierarchisch aufgebaut. Durch das klickbare Menü (siehe Abb. 31) lässt sich durch diese Struktur navigieren. Die Übersichtsseite stellt eine Navigationsleiste dar, wobei die Hauptelemente die Kapitel sind. Auf der

untere Ebenen befinden sich die Unterkapitel, gefolgt von den Übungen. In der ersten Spalte links sind die Informationen, Beispiele und Aufgaben-

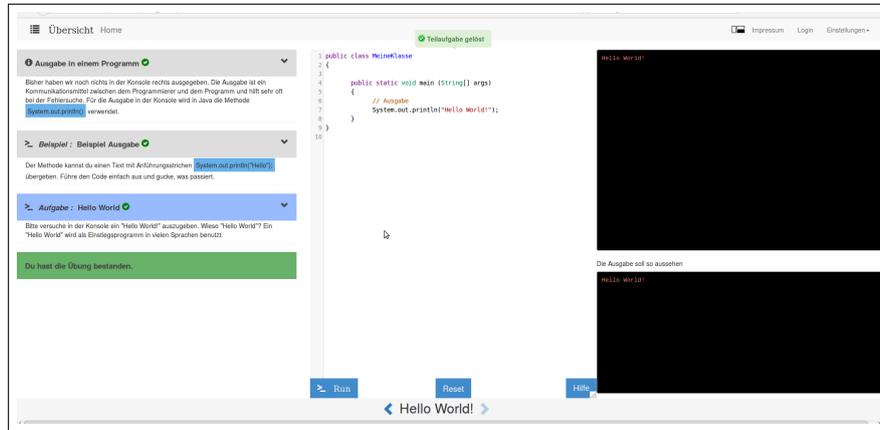


Abbildung 32: Aufgabe, Code und Ausgabe der Lernumgebung

texte. Sie lassen sich ein- und ausklappen. Wenn die vorherigen Aufgaben einklappt sind, wird es übersichtlicher für den Benutzer. Seine Konzentration wird auf die aktuelle Aufgabe gelenkt. In der Mitte ist der Code Editor und unter ihm befinden sich der *Run*-, der *Reset*- und der *Hilfe*-Button. Rechts daneben befindet sich das Ausgabe Element. Um deutlicher zu machen, was für eine Lösung erwartet wird, wird die erwartete Ausgabe der Aufgabe drunter angezeigt. Die Anordnung der des Code-Editors und der Ausgabe erscheint vertikal (siehe 32) oder horizontal (siehe 33). Je nach der Länge und Breite des Codes ist die vertikale oder horizontale Anordnung übersichtlicher.

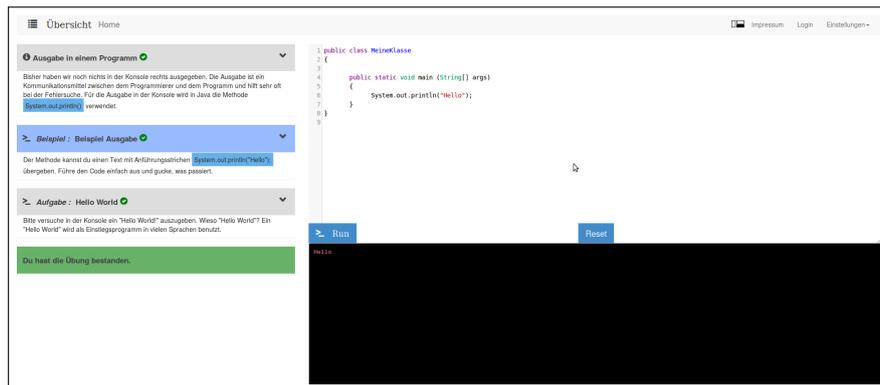


Abbildung 33: Änderung der Anordnung zum klassischen Editor

8.2 Admin und Content Manger

Die Verwaltung der Webapplikation übernimmt ein Administrator. Wie in der Abbildung 34 und Abbildung 35 zu sehen ist, können neue Lerninhalte, wie Kapitel und Übungen, von ihm angelegt und bestehende Lerninhalte geändert werden.

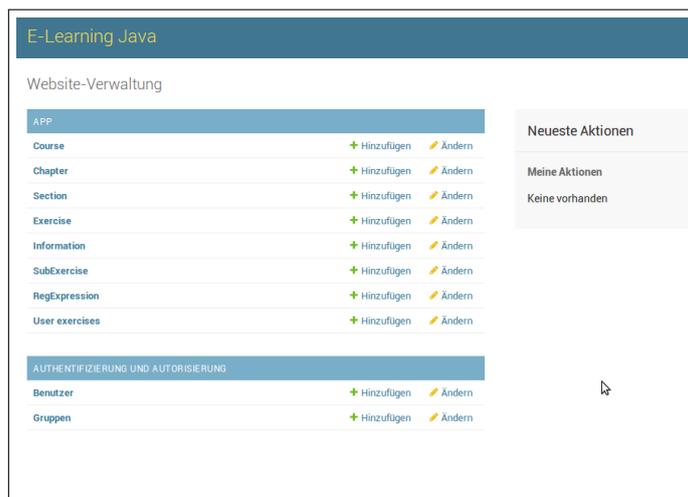


Abbildung 34: Übersicht des Admins Menü

Bei der Erstellung neuer Aufgaben soll die Anwendung eine Funktionalität zur Validierung des vom Benutzer eingegeben Codes bieten. Diese Funktionalität wurde durch den Einsatz von regulären Ausdrücken realisiert. Die Abbildung 36 zeigt in dem Eingabefeld (*Expression*) den regulären Ausdruck, der auf den entsprechenden Code im unterem Eingabefeld (*Code*) zutreffen muss. Der reguläre Ausdruck kann getestet und gespeichert werden.

sub exercise ändern

Id:

Titel:

Order:

Exercise:

Text:
 Die `<div class="highlight">System.out.println()</div>` gibt alles in einer Zeile aus.
 Die Tabelle soll richtig angezeigt werden. Das heißt, jede Zeile soll einzeln ausgegeben werden.
 Baue ein zusätzliches `<div class="highlight">System.out.println()</div>` in den Code ein.

Image: Keine Datei ausgewählt.

Url:

Type:

Subtype:

Abbildung 35: Die Teilübung kann hier geändert werden

Expression

System.out.println^(42)

Code

```
public class MeineKlasse
{
    public static void main (String[] args)
    {
        System.out.println();
    }
}
```

Matched String : No match
 Queried Variable : -

Reguläre Ausdrücke zu dieser Übung

- System.out.println^(42)

Abbildung 36: Test und Speicherung von regulären Ausdrücken

9 Auswertung

Dieses Kapitel beschäftigt sich mit der Auswertung des vorhandenen Fragebogens zu der Anwendung. Das Ziel der Umfrage besteht darin, aufgrund der erhobenen Ergebnisse zu prüfen, ob der Kurs für Programmieranfänger geeignet ist.

9.1 Fragebogen

Die Umfrage besteht aus 19 Fragen und wurde anonym, ohne Rückschlüsse auf die Teilnehmer nehmen zu können, durchgeführt. Der vollständige Umfragebogen befindet sich im Anhang auf S.66.

Der Fragebogen fängt mit drei Fragen zu der Person an (Frage 1-3): eine Frage zum Alter sowie zur Erfahrung des Benutzers. Außerdem eine Frage über die zukünftigen beruflichen Interessen des Benutzers. Anschließend folgen sechs Testfragen (Frage 4-9) zur Programmiersprache Java. Die Testfragen haben vier vorgegebene Antwortmöglichkeiten, wobei nur eine Antwort ausgewählt werden kann. Der Zweck dieser Fragen ist die Überprüfung des neuen (im Kurs gewonnen) Wissensstandes des Benutzers.

Die Antwortmöglichkeiten der Fragen 10 bis 17 sind nach der Likert-Skala (siehe [BD07] S. 224) konzipiert. Es gibt fünf mögliche Werte. Die Skala fängt links mit dem Wert 1 - "Trifft überhaupt nicht zu" - an und schließt rechts mit dem Wert 5 - "Trifft voll und ganz zu" - ab. Der Proband wählt die Tendenz

Die Aufgaben waren für mich lösbar.

	1	2	3	4	5	
trifft überhaupt nicht zu	<input type="radio"/>	trifft voll und ganz zu				

Abbildung 37: Frage aus der Umfrage nach der Likert Skala

seiner Einstellung zu einer Aussage aus.

Frage 10 und 11 sollen ein Aspekt der Benutzbarkeit testen. Die folgenden Fragen (Fragen 12 bis 14) sollen herausfinden, wie gut die Aufgaben und Beispiele von der Formulierung und der Lösbarkeit sind. Die nächsten Fragen (16 und 17) sollen feststellen, ob die Benutzer weiterhin motiviert sind am Kurs teilzunehmen. Die letzten Fragen (18 und 19) erfassen, wie die Probanden die Anwendung aus ihrer eigenen Sicht beschreiben. Die Antworten zu den Fragen sollen frei formuliert werden. Frage 18 fragt, was der Benutzer an dem Kurs gut findet. In der Frage 19 soll der Proband konstruktive Vorschläge für die Verbesserung von Anwendung benennen. Die Antworten liefern zusätzliche und wichtige Informationen über die Wahrnehmung des Kurses von den Benutzern.

9.2 Auswertung

Die Ergebnisse der Auswertung befinden sich in tabellarischer Form im Anhang auf der S.70.

Das durchschnittliche Alter der Teilnehmern der Umfrage, die ihr Alter angegeben haben, liegt bei 15 Jahren. Voraussichtlich gehen diese Personen noch zur Schule. Die genaue Verteilung in absoluten Zahlen ist auf Abbildung 38 zu sehen.



Abbildung 38: Frage 1

Frage (2) Hast du schonmal programmiert ?

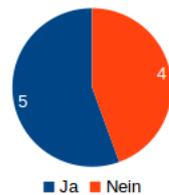


Abbildung 39: Frage 2

Über die Hälfte der Benutzer hat bereits Vorerfahrungen mit einer Programmiersprache gesammelt. Dieser Sachverhalt wird in Abbildung 39 mit einem Kreisdiagramm verdeutlicht.

66,67% der Probanden bejahen ein technisches Interesse an einem Hobby oder Beruf (Frage 3). Eine Person hat diese Frage verneint. 22,22% der Befragten haben eine Angabe in Sonstiges gemacht (siehe Abbildung 40).

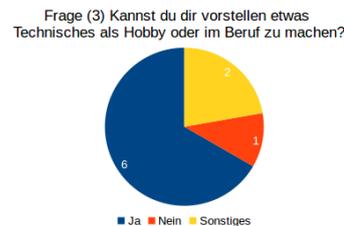


Abbildung 40: Frage 3

Zwei Fragen (8 und 9) von den Testfragen wurden von jedem Probanden richtig beantwortet. Die wenigsten richtigen Antworten mit 66,67% ergaben die Fragen 5 und 6, welche sich auf Methoden und Klassen in Java beziehen. Der Durchschnitt aller richtigen Antworten in Fragenblock zu Java (Fragen 4 bis 9) liegt bei 85,19%. Die Auswertungen der Fragen 4 bis 9 wird in der Abbildung 41 zusammengefasst.

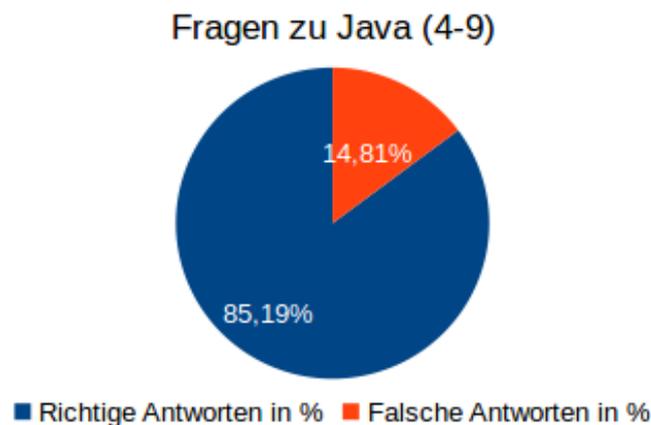


Abbildung 41: Fragen zu Java - 4 bis 9

Die Anzahl der richtigen Antworten lässt darauf schließen, dass die Benutzer die fundamentalen Inhalte des Kurses verstanden haben. Bei den Fragen, die am wenigsten richtig beantwortet wurden, kann in der Anwendung eine zusätzliche Aufgabe zu dem Thema gestellt werden.

Als nächstes werden die Bewertungen der Aussagen zu der Benutzbarkeit der Applikation (Aussage 10 - 11) dargestellt.

Die Aussage 10 "Ich konnte den Code ausführen" weist einen Mittelwert von 3,89 und eine Standardabweichung von 0,57 auf. Dies bedeutet, dass die meisten Benutzer den Code ohne große Schwierigkeiten ausführen konnten.

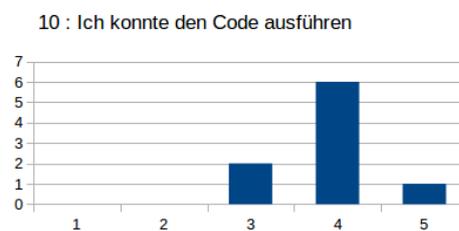


Abbildung 42: Aussage 10

Die Aussage 11 "Ich konnte leicht zu der nächsten Übung gehen" zeigt einen Mittelwert von 4,22 sowie eine Standardabweichung von 0,69 auf. Wie in der Abbildung 43 zu sehen ist, geht die Tendenz zur vollen Zustimmung der Funktion.



Abbildung 43: Aussage 11

Die Aussage 12 "Ich habe die Beispiele und die Aufgaben verstanden" ergibt einen Mittelwert von 3,78 auf. Die Standardabweichung besteht bei 0,79. Der Großteil der Benutzer neigt zu der neutralen Bewertung der Aussage.

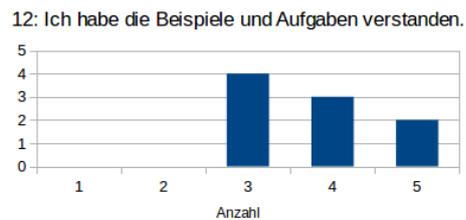


Abbildung 44: Aussage 12

Die Aussage 13 "Die Aufgaben waren für mich lösbar" weist einen Mittelwert von 3,89 und eine Standardabweichung von 0,99 auf. Aus der Auswertung ist ersichtlich, dass der durchschnittliche Benutzer die Aufgaben lösen kann. Wie in der Abbildung 45 zu sehen ist, existiert eine Lücke zwischen den Kursteilnehmern, die die Aufgabe voll verstanden haben und den Teilnehmern, welche Schwierigkeiten bei dem Lösen der Aufgabe hatten.

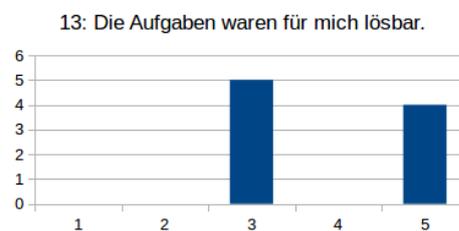


Abbildung 45: Aussage 13

Die Aussage 14 "Die Beispiele haben mir geholfen die folgenden Aufgaben zu lösen" ergibt einen Mittelwert von 3,56 und eine Standardabweichung von 0,96. Aus der Auswertung folgt, dass die Beispiele den Benutzern durchaus geholfen haben, die Aufgaben zu lösen. In der Abbildung 46 sieht man, dass die meisten Kursteilnehmer diese Aussage neutral bewertet haben. Die Verwendung der Beispiele im Kurs muss weiterhin erforscht werden.

14: Die Beispiele haben mir geholfen die folgenden Aufgaben zu lösen.

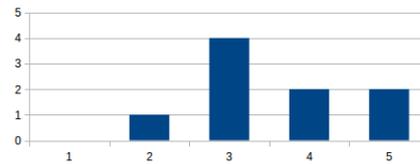


Abbildung 46: Aussage 14

Im folgendem werden die Aussage zu der Motivation und dem Interesse an dem Kurs (Aussage 15 - 16) analysiert.

Die Aussage 15 "Ich fand den Kurs interessant" ergibt einen Mittelwert von 4,44 und eine Standardabweichung von 1,07. Die Auswertung und die Abbildung 47 zeigen, dass das Interesse für den Kurs überdurchschnittlich groß ist.

15: Ich fand dem Kurs interessant.

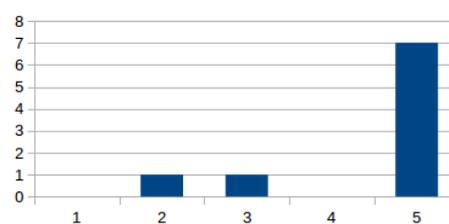


Abbildung 47: Aussage 15

Der Mittelwert für die Aussage 16 "Ich bin motiviert den Kurs weiter zu machen" liegt bei 4,22. Die Standardabweichung besteht bei 1,23. Die Auswertung verdeutlicht, dass die Motivation den Kurs wieder zu besuchen groß ist. Wie in der Abbildung 48 dargestellt wird, gab es einen Wert, der nicht in die erwartete Messreihe passt.

16: Ich bin motiviert den Kurs weiter zu machen.

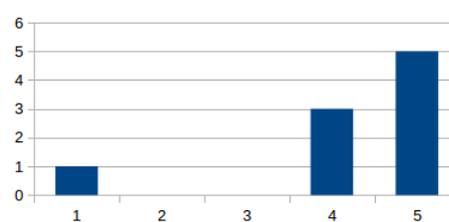


Abbildung 48: Aussage 16

Die Abbildung 49 zeigt die Notenvergabe von dem Kurs. Wie es zu sehen ist, geht die Tendenz zu einer guten Note wie das auch der Durchschnitt mit 2,11 zeigt.

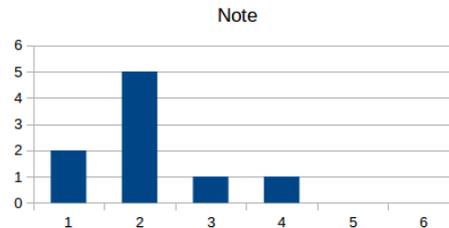


Abbildung 49: Note

Nachstehend werden die Antworten zu der Frage 18 "Was war gut (du kannst einfach was schreiben)?" und der Frage 19 "Was war schlecht und was könnte man besser machen (du kannst einfach was schreiben)?" des Fragebogens analysiert.

Aus den Antworten zu Frage 18 ist ersichtlich, dass der Aufbau des Kurses im allgemeinen („Der Aufbau meistens gut, verständlich erklärt“) als übersichtlich und gut bewertet wurde. Die Beispiele und Aufgaben wurden als praxisnahe („man so etwas Praxisbezug hatte“) bezeichnet. Aus den Kommentaren lässt sich schließen, dass der Ansatz der praktischen Umsetzung der Java Programmierung mit Beispielen und Aufgaben bei den Teilnehmern gut ankommt. Der Fokus auf einen praxisorientierten Kurs der Programmiersprache Java sollte weiter beibehalten werden.

Interessante Erkenntnisse können aus den Antworten zu der Frage 19 gewonnen werden. Die Teilnehmer haben die Tippfehler und teilweise unklaren Formulierungen der Aufgaben bemängelt. Zudem wurde es kritisiert, dass manche Elemente der Applikation (zum Beispiel das Ausführen des Codes für eine Aufgabe) nicht ausführbar waren. Ein Befragter wünscht sich mehr Aufgaben ohne Code-Vorgaben.

Im Zusammenhang auf die vorgestellten Ergebnisse für die Frage 19 wären weitere Beobachtungen und Inhalt- beziehungsweise Formulierungsanalysen lohnenswert. Ferner bietet sich das Testen der Applikation in unterschiedlichen Browsern an. Darüber hinaus ergibt sich ein weiterer Forschungsbedarf hinsichtlich der Frage, wie detailliert die theoretischen Informationen für einen Programmieranfängerkurs sein müssen ("Erklärung hätten teilweise detaillierter sein können").

9.3 Beobachtung

Bei der Umfrage wurden folgende Auffälligkeiten beobachtet.

Die Probanden brauchten zunächst Zeit, um sich an die Benutzeroberfläche mit den drei Spalten zu gewöhnen. Die Benutzer verbrachten eine gewisse Zeit damit, wie eine Aufgabe ausgewählt werden kann und wie der Code zu der Aufgabe angezeigt werden kann.

Die Benutzer hatten bei der folgenden Aufgabe Schwierigkeiten :

Als erstes wird dem Benutzer Beispiel des Quellcodes 6 gezeigt.

Quellcode 6: Mehrzeiliger Kommentar in Java

```
public class MeineKlasse
{
    public static void main (String[] args)
    {
        System.out.println("Ein mehrzeiliger Kommentar");
        /*
        Dies
        ist
        ein
        mehrzeiliger
        Kommentar
        */
    }
}
```

Als nächstes folgt die Aufgabe, die darin besteht, die Zeile 6 bis 9 vom Code(siehe Quellcode 7) auszukommentieren :

Quellcode 7: Aufgabe um einen mehrzeiligen Kommentar zu erstellen

```
public class MeineKlasse
{
    public static void main (String[] args)
    {
        System.out.println("Hallo");
        System.out.println("Kommentiere");
        System.out.println("mich");
        System.out.println("aus");
        System.out.println("bitte");
    }
}
```

Die meisten Benutzer haben aus dem Beispiel nicht verstanden, dass nicht nur der Text, sondern auch der Code auskommentiert werden kann. Aus

diesem Grund wurde das Beispiel überarbeitet. Im Code wird nun explizit gezeigt, dass der Code auskommentiert werden kann (siehe Quellcode 8). Es wird weniger Transferleistung von den Teilnehmern verlangt.

Quellcode 8: Mehrzeiliger Kommentar von Code

```
public class MeineKlasse
{
    public static void main (String[] args)
    {
        System.out.println("Ein mehrzeiliger Kommentar");
        /* System.out.println("Hallo");
        System.out.println("Auskommentiert");
        System.out.println("Auskommentiert");
        System.out.println("Auskommentiert");
        */
    }
}
```

9.4 Schlussfolgerung

Die Umfrage und die Beobachtung bei dem Test lassen viele Erkenntnisse zu.

Die entwickelte E-Learning Anwendung wurde zum großen Teil positiv aufgenommen. Der Java Test in der Umfrage bestätigt, dass die Teilnehmer den Inhalt der ersten Lerneinheiten verstanden haben.

Einerseits lässt sich feststellen, dass die Funktionalität der Plattform ständig verbessert werden muss um eine gute Benutzbarkeit zu erreichen. Der Inhalt, die Formulierungen der Aufgaben und Beispiele müssen ständig analysiert und verbessert werden, denn besonders bei Programmieranfängern ist jede Formulierung wichtig. Für die Zukunft macht es Sinn, kleine Benutzertests zu machen und die Benutzer beim Lösen der Aufgaben in der Lernumgebung zu beobachten.

Das Gleiche gilt für die Beispiele. Einerseits wurde die Frage „Die Beispiele haben mir beim Lösen der Aufgaben geholfen“ überwiegend neutral bewertet. Jedoch wurden die Beispiele in den Kommentaren von Frage 18 positiv gesehen. Es besteht ein weiterer Forschungsbedarf in der Frage, ob man die Beispiele in einem Programmierkurs für Anfänger verwenden soll.

10 Fazit und Ausblick

E-Learning Anwendungen sind heutzutage von großem Interesse, denn im Vordergrund steht ihr didaktisches Konzept.

In dieser Arbeit wurden zunächst aktuelle E-Learningplattformen zur Programmiersprache Java analysiert. Aus den gewonnenen Erkenntnissen wurde ein erweitertes Konzept erstellt. Das didaktische Konzept diente anschließend als Fundament für den technischen Entwurf.

Die Anwendung wurde erfolgreich als Prototyp entwickelt. Zudem wurde ein umfangreicher Inhalt für die Anwendung produziert. Viele unterschiedliche Disziplinen mussten bei der Entwicklung eines solchen Kurses beachtet werden: die Analyse der gegenwärtigen Plattformen, das didaktische Konzept, der technische Entwurf, die Entscheidung für die Technologien, die Implementierung, das Design und die Erstellung des Contents.

Die Lernumgebung wurde modular und flexibel implementiert, sodass eine Erweiterung des Kurses, beispielsweise mit der Programmiersprache C++, ohne großen Aufwand möglich ist.

Es wird fremder Code durch Webapplikation auf dem Server ausgeführt, die Mindestanforderungen an Sicherheit wurden erfüllt. Doch es bedarf eine weitere Diskussion um die Erhöhung der Sicherheit der Webapplikation.

Der Test mit den Probanden und die Umfrage haben wichtige und konstruktive Erkenntnisse gebracht. Der E-Learning Kurs wurde überwiegend positiv angenommen. Die Tests haben gezeigt, dass die meisten Befragten den Inhalt des Kurses verstanden haben. Bei der Beobachtung der Teilnehmern wurde festgestellt, dass Verbesserungsbedarf in manchen Funktionalitäten sowie in inhaltlichen Elementen besteht.

Insbesondere empfiehlt sich eine andauernde Verbesserung des Contents. Dazu ist es sinnvoll, einzelne Aufgaben und Texte von kleinen Testgruppen evaluieren zu lassen. Haben mehrere Benutzer bei einer Aufgabe Schwierigkeiten macht es Sinn die Aufgabe zu optimieren.

Aus der Beobachtung der Testgruppe kam es zu der Erkenntnis, dass die Kooperation der Teilnehmer untereinander eine fördernde Wirkung auf den Lernprozess hat. In dieser Frage besteht Forschungsbedarf.

Eine Integration in die Olat-Plattform [Ola17] über die LTI-Schnittstelle ist wünschenswert und bedarf weiterer Untersuchung. Die Vorteile der Integration sind, dass alle E-Learninganwendungen in der Plattform vereint sind und somit relativ unkompliziert verwaltet werden können. Der Benutzer kann über die Universitätsdaten anmelden. Olat würde dabei der "LTI Consumer" sein. Die E-Learninganwendung würde die Informationen an die Plattform weiterleiten, beispielsweise darüber ob der Benutzer den Kurs bestanden hat oder nicht.

A Anhang

A.1 Beispiel von der HelloWorld Aufgabe

```
- model: app.exercisestep
  pk: aufgabe_hello_world
  fields:
    title: Hello World
    exercise: hello_world
    text : |
      Bitte versuche in der Konsole ein "Hello World!"
      auszugeben. Wieso "Hello World"?
      Ein "Hello World" wird als Einstiegsprogramm
      in vielen Sprachen benutzt.
    order: 3
    type: subexercise
    subtype: console

- model: app.subexercise
  pk: aufgabe_hello_world
  fields:
    expected_output: Hello World!
    order: 3
    code : |
      public class MeineKlasse
      {

          public static void main (String[] args)
          {
              // Ausgabe
              System.out.println("World!");
          }
      }
    solution: |
      public class MeineKlasse
      {

          public static void main (String[] args)
          {
              // Ausgabe
              System.out.println("Hello World!");
          }
      }
```

A.2 Installation der Webanwendung

Dieses Kapitel beschreibt die Vorgehensweise der Installation der Serveranwendung. Dabei wird nicht auf jede Bibliothek oder zusätzlicher Anwendung, die installiert werden muss, eingegangen. Das Betriebssystem des Servers, das in dieser Arbeit beispielhaft verwendet wird, ist Ubuntu 16.04 Server. Zuerst wird ein unprivilegierter Benutzer erstellt, mit entsprechendem Passwort.

```
>sudo adduser server
```

Alle folgenden Installationsschritte erfolgten auf dem genannten Betriebssystem. Die Installation erfolgt durch zwei Paketmanager. Einmal handelt es sich das Advance-Package-Tool (apt-get), der generell Pakete unter Debian Linux verwaltet. Der andere Paketmanager PIP verwaltet die Python Pakete. Installiere PIP :

```
>sudo apt-get install python3-pip
```

Java muss installiert werden um den Code auf Server zu installieren und auszuführen.

```
>sudo apt-get install openjdk-8-jdk
```

Code auschecken oder per Zip

```
>sudo apt-get install subversion
```

Installiere VirtualEnv über PIP :

```
>sudo pip3 install virtualenv
```

Jetzt müsste man sich als der server user einloggen

```
>su server
```

Ordner im Heimatverzeichnis von server für den Checkout wird erstellt.

```
>mkdir koja
```

Befehl für SVN Download (username muss angepasst werden)

```
>svn co --username=yourUserName https://svn.uni-koblenz.de/kasjen/diplom/code/ koja
```

Erstelle VirtualEnv (am besten im 'Home Directory' des Benutzers 'server'):

```
>virtualenv koja_env
```

Aktiviere VirtualEnv :

```
>source koja_env/bin/activate
```

Durch die Aktivierung kann man in einer abgeschlossenen Umgebung arbeiten. Der Vorteil ist das nur bestimmte Pakete für diese Umgebung installiert werden. Folgende Pakete werden nun für die Webapplikation erstellt. Die Pakete mit ihrer Versionierung kann man in eine requirements.txt (koja/deployment) einfügen. Am einfachsten ist die Installation mit dem Befehl :

```
(koja_env)>pip3 install -r koja/deployment/requirements.txt
```

Wenn der Build von pillow nicht geht dann muss eine jpeg library installiert werden aber root Benutzer.

```
>sudo apt-get install libjpeg-dev zlib1g-dev
```

Es folgt eine einfache Einrichtung der Datenbank. Die Datenbank wird anhand von der models.py erstellt.

```
(koja_env)>python3 manage.py makemigrations app
(koja_env)>python3 manage.py migrate
```

Beispieldaten sind in der Yaml Datei vorhanden. Diese wird durch den folgenden Befehl in die Datenbank geladen.

```
(koja_env)>python3 manage.py loaddata initial_data.yaml
```

Installation des NGINX Servers und der UWSGI (Web Server Gateway Interface) Schnittstelle

```
>sudo apt-get install uwsgi uwsgi-plugin-python3
>sudo apt-get install nginx
```

Alle statischen Dateien werden in einem Ordner gesammelt. Dies macht python über den Befehl collectstatic.

```
(koja_env)>python3 manage.py collectstatic
```

In den Konfigurationsdatei müssen evtl. die statischen Pfade und die Userid (in Moment ist alles auf /home/server/) geändert werden. Dann werden die bereitgestellten (im Projekt Ornder config) Konfigurationsdateien in das entspreche Konfigurationsverzeichnis kopiert.

```
>sudo cp ~/path/to/your/myproj/config/mysite_nginx.conf /etc/nginx/sites-enabled/
>sudo cp ~/path/to/your/myproj/config/uwsgi.ini /etc/uwsgi/apps-enabled/
```

Der Server kann neu gestartet werden und kann auf der ip :8080 angeschaut werden

A.3 Umfragebogen

Java Kurs Umfrage

1. Wie alt bist du?

2. Hast du vorher schon mal programmiert?

Markieren Sie nur ein Oval.

- Ja
- Nein
- Sonstiges: _____

3. Kannst du dir vorstellen etwas Technisches als Hobby oder im Beruf zu machen?

Markieren Sie nur ein Oval.

- Ja
- Nein
- Sonstiges: _____

Bitte beantworte einige Fragen zu Java:

4. Welche Klasse ist in Java gültig?

Markieren Sie nur ein Oval.

- public klasse MeineKlasse{ der Code kommt hier }
- public class MeineKlasse{ der Code kommt hier }
- public class MeineKlasse[der Code kommt hier]
- default MeineKlasse{ der Code kommt hier }

5. Wie schreibt man eine Methode in Java richtig?

Markieren Sie nur ein Oval.

- public void class meineMethode() { der Code kommt hier }
- public void variable meineMethode() { der Code kommt hier }
- public void meineMethode() { der Code kommt hier }
- public function meineMethode() { der Code kommt hier }

6. Wo befinden sich in Java immer die Methoden?

Markieren Sie nur ein Oval.

- innerhalb einer Klasse
- außerhalb der Klasse
- nach dem Klassenrumpf
- vor dem Klassenrumpf

7. Wie heißt die Methode, die als Erstes im Javaprogramm ausgeführt wird?

Markieren Sie nur ein Oval.

- public static void start(String [] args)
- public static void main(String [] args)
- public static void starte(String [] args)
- public static void function(String [] args)

8. Wie kann das Wort "Hello" in Java in der Konsole ausgegeben werden?

Markieren Sie nur ein Oval.

- System.Println.Out("Hello")
- System.get.out("Hello")
- System.out.println("Hello")
- System.out.log("Hello")

9. Wie kann man in Java nur eine Zeile Code auskommentieren?

Markieren Sie nur ein Oval.

- // Das ist ein Kommentar
- # Das ist ein Kommentar
- <!--das ist ein Kommentar-->
- (das ist ein Kommentar)

Bitte bewerte auf einer Skala von 1 (trifft überhaupt nicht zu) bis 5 (trifft voll und ganz zu) folgende Aussagen:

10. Ich konnte den Code ausführen und konnte immer das Ergebnis der Ausführung sehen.

Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft überhaupt nicht zu	<input type="radio"/>	trifft voll und ganz zu				

11. Ich konnte leicht zu nächsten Übung gehen.

Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft überhaupt nicht zu	<input type="radio"/>	trifft voll und ganz zu				

12. Ich habe die Beispiele und Aufgaben verstanden.

Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft überhaupt nicht zu	<input type="radio"/>	trifft voll und ganz zu				

13. Die Aufgaben waren für mich lösbar.

Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft überhaupt nicht zu	<input type="radio"/>	trifft voll und ganz zu				

14. Die Beispiele haben mir geholfen die folgenden Aufgaben zu lösen.

Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft überhaupt nicht zu	<input type="radio"/>	trifft voll und ganz zu				

15. Ich fand dem Kurs interessant.

Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft überhaupt nicht zu	<input type="radio"/>	trifft voll und ganz zu				

16. Ich bin motiviert den Kurs weiter zu machen.

Markieren Sie nur ein Oval.

	1	2	3	4	5	
trifft überhaupt nicht zu	<input type="radio"/>	trifft voll und ganz zu				

Bitte bewerte den Kurs insgesamt:

Java Kurs Umfrage

17. Ich gebe dem Kurs die folgende Note:

Benutze hierzu das Schulnotensystem
von 6 (ungenügend) bis 1 (sehr gut).

18. Was war gut (du kannst einfach was schreiben)?

19. Was war schlecht und was könnte man besser machen (du kannst einfach was schreiben)?

Bereitgestellt von



Teilnehmer an der Umfrage : 9

A4 Ergebnisse der Umfrage

Frage (1) Wie alt bist du ?		
Alter	Häufigkeit	In %
14	5	55,56%
15	1	11,11%
17	1	11,11%
19	1	11,11%
Keine Angabe	1	11,11%

Frage (2) Hast du schon mal programmiert ?		
Antwort	Häufigkeit	In %
Ja	5	55,56 %
Nein	4	44,44 %

Frage (3) Kannst du dir vorstellen ... ?		
Antwort	Häufigkeit	Prozent
Ja	6	66,67 %
Nein	1	11,11 %
Sonstiges	2	22,22 %

Die Zelle bzw. die Nr mit der richtigen Antwort ist grün umrandet

Antwort	Frage (4)	Frage (5)	Frage (6)	Frage (7)	Frage (8)	Frage (9)
1	1	3	6	1	0	9
2	8	0	2	8	0	0
3	0	6	1	0	9	0
4	0	0	0	0	0	0

Auswertung nach richtige Antworten

Frage Nr.	4	5	6	7	8	9	Summe
Richtige Antworten	8	6	6	8	9	9	46
Falsche Antworten	1	3	3	1	0	0	8
Richtige Antworten in %	88,89%	66,67%	66,67%	88,89%	100,00%	100,00%	85,19%
Falsche Antworten in %	11,11%	33,33%	33,33%	11,11%	0,00%	0,00%	14,81%
	0	0	0	0	0	0	

Trittft (nicht) zu	Frage (10)	Frage (11)	Frage (12)	Frage (13)	Frage (14)	Frage (15)	Frage (16)
1	0	0	0	0	0	0	1
2	0	0	0	0	1	1	0
3	2	1	4	5	4	1	0
4	6	5	3	0	2	0	3
5	1	3	2	4	2	7	5
Mittelwert	3,89	4,22	3,78	3,89	3,56	4,44	4,22
Standardabweichung	0,57	0,63	0,79	0,99	0,96	1,07	1,23

Aussage (1-5)	Frage (10)	Frage (11)	Frage (12)	Frage (13)	Frage (14)	Frage (15)	Frage (16)
1	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	11,11%
2	0,00%	0,00%	0,00%	0,00%	11,11%	11,11%	0,00%
3	22,22%	11,11%	44,44%	55,56%	44,44%	11,11%	0,00%
4	66,67%	55,56%	33,33%	0,00%	22,22%	0,00%	33,33%
5	11,11%	33,33%	22,22%	44,44%	22,22%	77,78%	55,56%

Kurs (Note)	Frage (17)
1	2
2	5
3	1
4	1
5	0
6	0

Kurs (Note)	In %
1	22,22%
2	55,56%
3	11,11%
4	11,11%
5	0,00%
6	0,00%

Note Ø
2,11111111

Abweichung
0,87488976

Frage (18) Was war gut ?

ziemlich alles

ich fand es gut das Beispiele und antworten die versteckt sind hier waren

Aufbau mit allg. meinen Info, Beispiel, Aufbau,

Die Beispiele, die einen zeigen konnten wie du in der Aufgabe vorzugehen hast

Der Aufbau meistens gut, verständlich erklärt

Übersichtlich gemacht

leicht und schnell erklärt, Beispiele, schön dargestellt

Durch die Beispiele und guten Erklärungen, wurde einem direkt deutlich, wie etwas geht

Die Idee einen Vorkurs für Java einzurichten, um Studienanfänger einen start ins Programmieren zu

erleichtern. Es war auch schön dass man den Code selbst eingeben konnte und man so etwas

Praxisbezug hatte

Frage (19) Was war schlecht (Was kann man verbessern) ?

einige kleine Fehler

Manche Aufgaben so formulieren, das man keine vorgegebene Lösung benutzt, mehrere Hilfestellungen formulieren

Es gab ein paar Tippfehler und manche Aufgaben haben nicht funktioniert. Aufgaben klarer formulieren.

Richtigkeit / Funktionstüchtigkeit der Aufgaben (sehr selten)

manche Aufgaben eindeutig erklären

man musste manche Fragen überspringen , da sie nicht funktionieren

Fehler beheben, einige Sachen z.B., boolean ein wenig mehr und ausführlicher erklären

ein paar Funktionsfehler

Erklärung hätten teilweise detaillierter sein können - Zu viele Rechtschreibfehler - Lösungen teilweise

inkorrekt - Nicht alle Lösungsmöglichkeiten für eine Aufgabe wurden erfasst – Form hätte besser sein

können

Literatur

- [3-S17] Wie funktioniert die 3-schichten-architektur, 27. Sep 2017. URL: <http://www.mrknowing.com/2013/11/08/wie-funktioniert-die-3-schichten-architekt>.
- [Ang17] One framework.mobile and desktop., 5. June 2017. URL: <https://angular.io/>.
- [Arn13] Lars Kilian und Anne Zimmer Gerhard M. Thillosen Arnold, Patricia. *Handbuch E-Learning*. Bertelsmann in Bielefeld, 2013. URL: <https://www.wbv.de/landingpages/handbuch-e-learning.html>.
- [BD07] J. Bortz and N. Döring. *Forschungsmethoden und Evaluation für Human- und Sozialwissenschaftler: Limitierte Sonderausgabe*. Springer-Lehrbuch. Springer Berlin Heidelberg, 2007. URL: <https://books.google.de/books?id=13GbPUYAUHsC>.
- [Blo17a] Blooms taxonomy, 20. Sep 2017. URL: <https://dbs-lin.ruhr-uni-bochum.de/lehreladen/planung-durchfuehrung-kompetenzorientierter-lehre/lehr-und-lernziele/typen-und-stufen/>.
- [Blo17b] Benjamin bloom wikipedia eintrag, 20. Sep 2017. URL: https://de.wikipedia.org/wiki/Benjamin_Bloom.
- [Blu17] Entwicklungsumgebung bluej, 15. Jul 2017. URL: <https://www.bluej.org/>.
- [Cod17a] Codecademy, 19. Jun 2017. URL: <https://www.codecademy.com/>.
- [Cod17b] Codingame, 20. Sep 2017. URL: <https://www.codingame.com/>.
- [Dja17] Django framework, 19. Jul 2017. URL: <https://www.djangoproject.com/>.
- [Dr 17] Drjava, 10. Sep 2017. URL: <http://drjava.sourceforge.net/>.
- [Dri17a] Behaviorismus - reiz-reaktions-schema, 20. Sep 2017. URL: <http://www.lernpsychologie.net/lerntheorien/behaviorismus>.
- [Dri17b] Drill-and-practice programm, 15. Sep 2017. URL: <https://www.e-teaching.org/materialien/glossar/drill-and-practice>.

- [Ecl17] Entwicklungsumgebung eclipse, 15. Jul 2017. URL: <https://www.eclipse.org/>.
- [Gai17] Gailer-net.de : Zum java tutorial interaktiv — programmieren lernen mit java, 12. Sep 2017. URL: <http://www.gailer-net.de/>.
- [Gra17] This graphics2d class extends the graphics class to provide more sophisticated control over geometry, coordinate transformations, color management, and text layout. this is the fundamental class for rendering 2-dimensional shapes, text and images on the java(tm) platform., 12. Aug 2017. URL: <https://docs.oracle.com/javase/8/docs/api/java/awt/Graphics2D.html>.
- [Hoh02] A. Hohenstein. *Handbuch E-Learning: Expertenwissen aus Wissenschaft und Praxis*. Number Bd. 1 in Handbuch Personalentwicklung. Dt. Wirtschaftsdienst, 2002. URL: <https://books.google.de/books?id=CuSloAEACAAJ>.
- [Hol95] K. Holzkamp. *Lernen: subjektwissenschaftliche Grundlegung*. Campus-Verlag, 1995. URL: <https://books.google.de/books?id=5pXIPQAACAAJ>.
- [HSSC05] James I. Hsia, Elspeth Simpson, Daniel Smith, and Robert Cartwright. Taming java for the classroom. *SIGCSE Bull.*, 37(1):327–331, February 2005. URL: <http://doi.acm.org/10.1145/1047124.1047459>, doi:10.1145/1047124.1047459.
- [IK02] Ludwig J. Issing and Paul Klimsa. *Information und Lernen mit Multimedia und Internet*. BeltzPVU, 2002.
- [Int17] Entwicklungsumgebung idea intellij, 15. Jul 2017. URL: <https://www.jetbrains.com/idea/>.
- [IP11] Mirjana Ivanović and Tomáš PitnerI. Technology-enhanced learning for java programming: Duo cum faciunt idem, non est idem. *ACM Inroads*, 2(1):55–63, February 2011. URL: <http://doi.acm.org/10.1145/1929887.1929906>, doi:10.1145/1929887.1929906.
- [Jav17a] Java ist der allgemeine begriff, der zur benennung der software und der dazugehörigen komponenten wie "java runtime environment"(jre), "java virtual machine"(jvm) sowie "plugin"verwendet wird. fehlermeldungen, die die wörter jre, jvm und plug-in enthalten, werden von uns gespeichert., 19. Jun 2017. URL: <https://www.java.com/>.

- [Jav17b] Java ist eine insel, 10. Sep 2017. URL: <http://openbook.rheinwerk-verlag.de/javainsel/>.
- [Jin17] Jinja, 20. Sep 2017. URL: <http://jinja.pocoo.org/>.
- [JQu17] JQuery, 19. Jul 2017. URL: <https://jquery.com/>.
- [Kar17] Kara der roboter, 15. Jun 2017. URL: <http://www.swisseduc.ch/informatik/karatojava/kara>.
- [Kat17] Blooms taxonomy revised, 2017. URL: <http://thesecondprinciple.com/teaching-essentials/beyond-bloom-cognitive-taxonomy-revised/>.
- [Ker01] Michael Kerres. *Multimediale und telemediale Lernumgebungen. Konzeption und Entwicklung*. Oldenbourg, 2001.
- [LAIM15] Timo Lehtonen, Timo Aho, Essi Isohanni, and Tommi Mikkonen. On the role of gamification and localization in an open online learning environment: Javala experiences. In *Proceedings of the 15th Koli Calling Conference on Computing Education Research, Koli Calling '15*, pages 50–59, New York, NY, USA, 2015. ACM. URL: <http://doi.acm.org/10.1145/2828959.2828973>, doi:10.1145/2828959.2828973.
- [lea17] Learnjavaonline, 10. Sep 2017. URL: <http://www.learnjavaonline.org/>.
- [LK15] Michael J. Lee and Andrew J. Ko. Comparing the effectiveness of online learning approaches on cs1 learning outcomes. In *Proceedings of the Eleventh Annual International Conference on International Computing Education Research, ICER '15*, pages 237–246, New York, NY, USA, 2015. ACM. URL: <http://doi.acm.org/10.1145/2787622.2787709>, doi:10.1145/2787622.2787709.
- [MR05] B.N. Miller and D.L. Ranum. *Problem Solving with Algorithms and Data Structures Using Python*. Franklin Beedle Series. Franklin, Beedle & Associates, 2005. URL: <https://books.google.de/books?id=GtODAAAACAAJ>.
- [NDH⁺08] Helmut M. Niegemann, Steffi Domagk, Silvia Hessel, Alexandra Hein, Matthias Hupfer, and Annett Zobel. *Kompendium multimediales Lernen (X.media.press) (German Edition)*. Springer, 2008. URL: <http://www.springer.com/de/book/9783540372257>.

- [Ola17] Olat virtueller campus rheinland-pfalz, 26. Sep 2017. URL: <https://olat.vcrp.de>.
- [Ora17] Oracle, 2017. URL: <http://www.oracle.com/technetwork/tutorials/index.html>.
- [Pos17] PostgreSQL is a powerful, open source object-relational database system. it has more than 15 years of active development and a proven architecture that has earned it a strong reputation for reliability, data integrity, and correctness., 19. Aug 2017. URL: <https://www.postgresql.org/>.
- [Pro17] Prozedurales wissen, 2017. URL: <http://www.spektrum.de/lexikon/psychologie/prozedurales-wissen/11947>.
- [Rea17] A javascript library for building user interfaces, 5. June 2017. URL: <https://facebook.github.io/react/>.
- [Sch02] Rolf Schulmeister. *Grundlagen hypermedialer Lernsysteme. Theorie - Didaktik - Design*. Oldenbourg, 2002.
- [Sol17] Sololearn, 10. Aug 2017. URL: <https://www.sololearn.com/>.
- [SQL17] Sqlite is an in-process library that implements a self-contained, serverless, zero-configuration, transactional sql database engine. the code for sqlite is in the public domain and is thus free for use for any purpose, commercial or private. sqlite is the most widely deployed database in the world with more applications than we can count, including several high-profile projects., 12. Aug 2017. URL: <https://www.sqlite.org/>.
- [VPL11] Arto Vihavainen, Matti Paksula, and Matti Luukkainen. Extreme apprenticeship method in teaching programming for beginners. In *Proceedings of the 42Nd ACM Technical Symposium on Computer Science Education, SIGCSE '11*, pages 93–98, New York, NY, USA, 2011. ACM. URL: <http://doi.acm.org/10.1145/1953163.1953196>, doi:10.1145/1953163.1953196.
- [Vue17] Javascript framework vue-js, 19. Jul 2017. URL: <https://vuejs.org/>.
- [Wis06] Jon Wise. Gojava: A java development tool for beginners. *SIGCSE Bull.*, 38(3):359–359, June 2006. URL: <http://doi.acm.org/10.1145/1140123.1140269>, doi:10.1145/1140123.1140269.

- [Yad11] Aharon Yadin. Reducing the dropout rate in an introductory programming course. *ACM Inroads*, 2(4):71–76, December 2011. URL: <http://doi.acm.org/10.1145/2038876.2038894>, doi:10.1145/2038876.2038894.
- [Zum10] Jörg Zumbach. *Lernen mit neuen Medien: Instruktionspsychologische Grundlagen (Kohlhammer Standards Psychologie) (German Edition)*. Kohlhammer Verlag, 2010. URL: https://www.kohlhammer.de/wms/instances/KOB/appDE/nav_product.php?product=978-3-17-016833-6.