Entwurf eines digitalen Lernszenariums

Fachseminar

Didaktik der Informatik

10-204-2007

Thema:

Zeitkomplexität

Name:	Charlotte Gerlitz
Matrikelnummer:	3741044
E-Mail-Adresse:	cg30towu@studserv.uni-leipzig.de

Inhaltsverzeichnis

1 Kurzübersicht zur Unterrichtseinheit

1.1 Rahmeninformationen

<table>
<tr><td>Unterrichtsfach</td><td>Informatik</td></tr>
<tr><td>Thema</td><td>Zeitkomplexität</td></tr>
<tr><td>Klassenstufe</td><td>Jahrgangsstufe 11</td></tr>
<tr><td>Lernbereich</td><td>Lernbereich 5: Algorithmen</td></tr>
<tr><td>Positionierung innerhalb des Lernbereiches</td><td>Mitte von „Beurteilen von Algorithmen bezüglich ihrer Effizienz“, nach Einführung der Begrifflichkeiten oder zum Festigen des Bestimmens von Zeitkomplexität</td></tr>
<tr><td>Kompetenzniveaus nach DQR</td><td>
<table>
<tr><th>Niveaus</th><th>1 bis 2</th><th>3 bis 4</th><th>5 bis 6</th><th>7 bis 8</th></tr>
<tr><td>Fachkompetenz</td><td></td><td>x</td><td></td><td></td></tr>
<tr><td>Methodenkompetenz</td><td></td><td>x</td><td></td><td></td></tr>
<tr><td>Selbstkompetenz</td><td></td><td></td><td>x</td><td></td></tr>
<tr><td>Sozialkompetenz</td><td>x</td><td></td><td></td><td></td></tr>
</table>
</td></tr>
</table>

1.2 Voraussetzungen zur Durchführung

Technische Voraussetzungen:

Computer mit Internetzugang, Browser für das Internet (Nutzung von OPAL)

Inhaltliche Voraussetzungen:

Die SuS müssen mit Programmcode, mindestens Pseudo-Code vertraut sein und diesen wenigstens auf ihrer Strukturebene lesen können. Sie müssen verstanden haben, was ein abstrahierter Rechenschritt ist und wie Algorithmenstrukturen arbeiten.

Der Umgang mit Termen und die Beschreibung von Sachverhalten in dieser abstrakten Art ist den SuS vertraut. Die SuS sollten ein Grundverständnis von Grenzwerten besitzen.

Ein Grundverständnis vom Bearbeiten von Tests in OPAL ist von Vorteil, da die SuS dann wissen, wann ihnen Rückmeldungen angezeigt werden (beim Abgeben einzelner Aufgaben).

2 Geförderte Kompetenzen

Fachkompetenzen:

Kompetenzniveau nach DQR

1 bis 2	3 bis 4	5 bis 6	7 bis 8
		x	

Die SuS können einzelnen Algorithmenstrukturen benötigte Zeiteinheiten zuordnen.

Durch die elektronische Übungseinheit werden die SuS darin trainiert, in einem vorgegebenen Algorithmus in Pseudo-Code zu annotieren, an welcher Stelle die Länge der Eingabe durchlaufen wird.

Kompetenzniveau nach DQR

1 bis 2	3 bis 4	5 bis 6	7 bis 8
	x		

Die SuS haben die Fähigkeit, die grundlegenden Eigenschaften und Vorteile einer Zeitkomplexitätsanalyse zu benennen und zu beschreiben.

Den SuS wird die Möglichkeit gegeben, anhand der Pseudo-Codezeilen den praktischen Nutzen zu üben und innerhalb von zusammenfassenden Texten den übergeordneten Zweck nachzuvollziehen.

Methodenkompetenzen:

Kompetenzniveau nach DQR

1 bis 2	3 bis 4	5 bis 6	7 bis 8
x			

Die SuS sind in der Lage, Selektionsköpfe in Pseudo-Code so zu vervollständigen, dass eine gewünschte Anzahl an Bearbeitungsschritten benötigt wird.

Zu vorgergebenen if-Köpfen sollen die SuS eine korrekte Auswahl an gegebenen Fortführungen finden, sodass die korrekte Anzahl an Bedingungen gegeben ist.

Kompetenzniveau nach DQR

1 bis 2	3 bis 4	5 bis 6	7 bis 8
		x	

Die SuS sind in der Lage, Algorithmen in Pseudo-Code anhand ihrer Laufzeit zu sortieren.

Von der Fähigkeit, einzelne Strukturen zu annotieren, werden die SuS aufgefordert, zu abstrahieren, um intuitiv einzelne Sequenzlaufzeiten zu addieren und anschließend die Codestücke richtig anzuordnen.

Selbstkompetenz oder Sozialkompetenz:

Kompetenzniveau nach DQR

1 bis 2	3 bis 4	5 bis 6	7 bis 8
		X	

Die SuS können ihren Wissenszuwachs einschätzen.

Die SuS können anhand der Übersichtsgestaltung selbst einen Einstieg wählen und von dort aus weiter arbeiten, sie bekommen individuelles Feedback über ihren Leistungsstand.

3 Sachanalyse

Die Zeitanalyse eines Programmes wird anhand der Länge *n* der Eingabe (meist als Zeichenanzahl des Eingabewortes aufgefasst) und anhand der angenommenen Bearbeitungsschritte zum Ausführen eines Befehls auf dieser Eingabelänge berechnet. Für einzelne Schritte, wie beispielsweise das Einlesen eines Zeichens oder das Schreiben eines Zeichens, werden feste, konstante Zeiten angenommen. Für Schritte, in denen verschiedene Dauern benötigt werden, beispielsweise bei Suchen, wird vom schlimmsten Fall, dem sogenannten Worst Case Szenarium ausgegangen. Damit wird eine Abschätzung des Zeitbedarfs eines Algorithmus nach oben hin erreicht. In diesem Sinne funktioniert die O-Notation, die sich mit der Komplexitätsklasse beschäftigt, in der sich ein Algorithmus befindet. Dabei werden mathematische Funktionsterme ebenfalls nach oben hin abgeschätzt und zwar im Verhalten im Unendlichen. So spielen Summanden, die beispielsweise eine niedrigere Potenz von n haben, eine untergeordnete Rolle und werden zum Zwecke der Abstraktion fallen gelassen. Konstante Faktoren spielen in der Grenzwertbetrachtung ebenfalls kaum einen Einfluss und werden aus diesem Grunde ebenfalls abstrahiert. So gelangt man zu einem vergleichsweise einfachen Term zur Beschreibung des Zeitaufwandes. Dieser lässt sich mit anderen vergleichen, indem die Komplexitätsklassen folgendermaßen von ‚wenig Zeitverbrauch' zu ‚viel Zeitverbrauch' angeordnet sind: $const < \log(n) < \sqrt{n} < n^a < a^n < n! < n^n$. Dabei sind *a* und *const* Konstanten aus den reellen Zahlen.

4 Didaktische Analyse

Die SuS werden digital durch das Lernszenarium geführt. Dabei werden Ihnen Multiple-Choice-Aufgaben zur Wissenszuwachsprüfung und Zuordnungsaufgaben an die Hand gegeben, um ihren eigenen Lernfortschritt zu überprüfen. Anhand der Ergebnisse werden die SuS gebeten, in verschiedenen Lernaufgaben weiter fortzufahren oder sich einzelne Informationsseiten noch einmal durchzulesen. Sie haben die Möglichkeit, die Übungsaufgaben jederzeit zu unterbrechen, um im Hefter oder im Internet nach Informationen zu suchen und dann weiter fort zu fahren. Um den Lernerfolg zu unterstützen, wird bei jeder falschen Antwort eine kurze Analyse gegeben, die postulierte Denkmuster der SuS hinterlegt, welche sie dazu verleitet haben könnten, diese entsprechende Antwort gewählt zu haben. Desweiteren können die SuS die Übungsaufgaben und den Eingangstest beliebig oft wiederholen, damit sie nach einem Wissenszuwachs auch diesen noch einmal prüfen können.

5 Methodische Analyse

Den SuS werden Folien präsentiert, auf denen Beschreibungen des Themas und Arbeitsanweisungen stehen. Sie sollen sich die gegebenen Materialien gewissenhaft anschauen und im Folgenden einige Quiz-Elemente in Form von Multiple-Choice- und Zuordnungsaufgaben beantworten. Dabei dient ihnen ein Selbst-Test als Einstiegsportal, der ihnen im Weiteren Übungsaufgaben oder sogar schon den Abschlusstest freischaltet.

Die Zuordnungsaufgaben befinden sich über einem Bild eines Programmcodes, auf dem die SuS die entsprechenden Laufzeiten anordnen sollen. Variiert werden diese beiden Aufgabentypen mit Korrekturaufgaben, in denen die SuS falsch annotierte Codezeilen identifizieren müssen und Aufgaben, bei denen sie Codezeilen in aufsteigender Zeitkomplexität anordnen müssen. Zur Unterstützung bereits vorhandener Strukturen im Verständnis der SuS sind die Aufgaben nach Algorithmenstrukturen gegliedert. Es wird davon ausgegangen, dass eine Sequenz einfacher zu annotieren ist als eine Selektion, welche aufgrund des Zählens der Bedingungen wieder einfacher zu annotieren ist als eine Schleife, bei der die Durchgänge oftmals von der Veränderung im Schleifenkörper abhängt.

Um vorzubeugen, dass SuS aufgrund einer Fehleinschätzung der eigenen Leistungen wenig Punkte im Abschlusstest erzielen, wird dieser erst freigeschaltet, nachdem eine Mindestanzahl in der Summe der einzelnen Übungsaufgaben oder die volle Punktzahl im Selbst-Test erreicht ist.

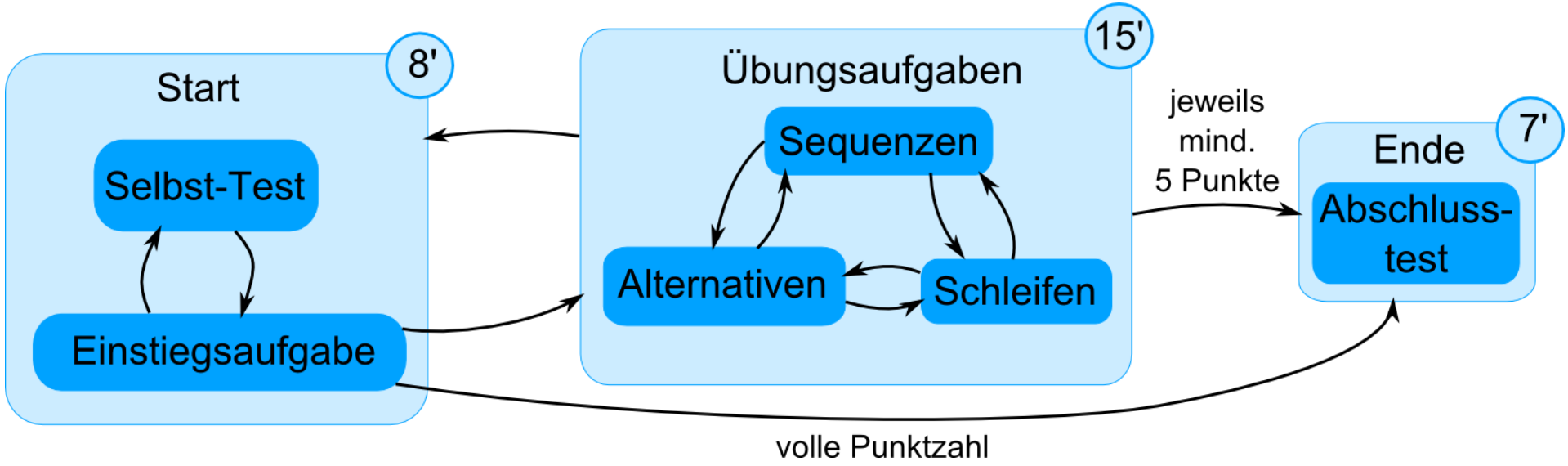
6 Erfolgsevaluation

Durch automatische Rückmeldungen der digitalen Tests können die SuS ihren Lernstand eigenständig überprüfen. Dabei ist auf dem Selbst-Test und den Übungsaufgaben der SuS eine individuelle Rückmeldung für verschiedene Antwortmöglichkeiten definiert, die den SuS Hinweise liefert, wie sie ihr Wissen ergänzen und vervollständigen können. Die Lehrkraft kann den Punktestand der einzelnen SuS im OPAL-Kurs einsehen. In einer Auswertung am Ende der Stunde werden die SuS aufgefordert, ihre Stärken und Schwächen zu reflektieren sowie in Fragen und Rückmeldung mitzuteilen, inwieweit sie mit den Aufgaben zurecht gekommen sind und wo noch Übungsbedarf besteht.

7 Unterrichtsentwurf

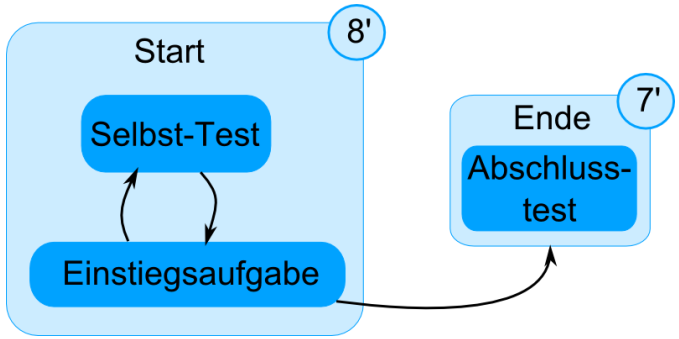
Zeit	Unterrichtsschritt	Lehrer-/ Schülertätigkeit	Methode	Sozialform	Lernmittel, Bemerkungen
1‘	Begrüßung	Vorbereiten auf den kommenden Unterricht	Lehrervortrag	Frontalunterricht	-
5‘	Erarbeitung	L: Gibt eine kurze Einführung in das Thema S: Hören zu	Lehrervortrag	Frontalunterricht	Beamer, Folien
30‘	Erarbeitung	Bearbeitung des e-Learning-Materials Annotieren von Pseudo-Code	Selbstständige SuS-Tätigkeit	Einzelarbeit	Leitmedium: Interaktiver OPALkurs
8‘	Ergebnissicherung	Feedback zu Schwierigkeiten	Lehrgespräch	Frontalunterricht	Notizen aus Heftern, Reflexion
1‘	Verabschiedung	Stunde beenden	Lehrervortrag	Frontalunterricht	-

Schema des e-Learning-Materials der Erarbeitung: Annotieren von Programmcode



Unterrichtsentwurf zum Wiederholen:

Zeit	Unterrichtsschritt	Lehrer-/ Schülertätigkeit	Methode	Sozialform	Lernmittel, Bemerkungen
1‘	Begrüßung	Vorbereiten auf den kommenden Unterricht	Lehrervortrag	Frontalunterricht	-
5‘	Erarbeitung	Ein S: Gibt Wiederholung zum Annotieren	Schülervortrag	Frontalunterricht	Beamer, Folien, ggf. Tafel & Kärtchen (SuS-Materialien)
15‘	Erarbeitung	Bearbeitung des e-Learning-Materials Annotieren von Pseudo-Code	Selbstständige SuS-Tätigkeit	Einzelarbeit	Leitmedium: Interaktiver OPALkurs
5‘	Erarbeitung	Ein S: Gibt Wiederholung zum Zusammenfassen und Abstrahieren	Schülervortrag	Frontalunterricht	Beamer, Folien, ggf. Tafel & Kärtchen (SuS-Materialien)
15‘	Erarbeitung	Bearbeitung des e-Learning-Materials Landau-Symbolik und Hierarchie	Selbstständige SuS-Tätigkeit	Einzelarbeit	Leitmedium: Interaktiver OPALkurs
3‘	Puffer	Reflexion / Klärung abschließender Fragen	Selbstst. / Klassen-gespräch	Einzelarbeit/ Frontalunterricht	Evtl. Tafel, Beamer, OPALkurs
1‘	Verabschiedung	Stunde beenden	Lehrervortrag	Frontalunterricht	-



Anhang

Einstiegsannotation

Wählen Sie die richtigen Bearbeitungszeiten aus.

Diese Aufgabe dient der Einschätzung. Sie wird genutzt, um Ihnen im Folgenden die Aufgaben zum Üben zuzuordnen.

	0	1	2
def einsteigen(char buchstabe)	☐	☐	☐
string wort	☐	☐	☐
int zahl	☐	☐	☐
wort = "Hallo"	☐	☐	☐
zahl = 1	☐	☐	☐
if buchstabe == 'a' then	☐	☐	☐
wort += " Anton!"	☐	☐	☐
else	☐	☐	☐
wort = "Wer bist du?"	☐	☐	☐
end	☐	☐	☐
for i = 1 .. 2	☐	☐	☐
wort += "!"	☐	☐	☐
end	☐	☐	☐
while zahl < 3	☐	☐	☐
print wort	☐	☐	☐

Literaturverzeichnis

(Literaturverzeichnis; nach APA-Standard, gerne als automatisches Verzeichnis.)

J. Hopcroft, R.Motwani, J. Ullman: Einführung in die Automatentheorie, Formale Sprachen und Komplexitätstheorie, Pearson 2002