

Rupprecht-Gymnasium
München
Studienseminar Mathematik/Informatik 2010/2012

ObjectWriter

Entwicklung und Erprobung eines Programms zur
Einführung der Klassen ZEICHEN und ABSATZ

Schriftliche Hausarbeit zur Zweiten Staatsprüfung
für das Lehramt an Gymnasien in Bayern

im Fachgebiet Informatik

vorgelegt von
Studienreferendar Martin Omasreiter

Abgabetermin 14.08.2011

Erklärung

Hiermit versichere ich, dass ich die Arbeit in allen Teilen selbständig gefertigt und keine anderen als die angegebenen Hilfsmittel verwendet habe. Die schriftliche Hausarbeit wurde nicht schon als Doktor-, Magister- oder Diplomarbeit bei einer Hochschule oder als schriftliche Hausarbeit bei einer anderen Staatsprüfung für ein Lehramt eingereicht.

München, 14.08.2011

StRef Martin Omasreiter

Inhaltsverzeichnis

1. Einleitung	4
2. Der Themenkomplex „Informationsdarstellung mit Textdokumenten“	6
2.1. Einbettung des Themenkomplexes „Informationsdarstellung mit Textdokumenten“ in den bayerischen Gymnasiallehrplan	6
2.2. Nachteile der bisherigen didaktischen Umsetzung	7
3. Das Programm <i>ObjectWriter</i>	10
3.1. Konzeption des Programms <i>ObjectWriter</i>	10
3.2. Erläuterung der einzelnen Programmkomponenten	12
3.3. Erwarteter Beitrag des Programms zum Verständnis der Unterrichtsinhalte	15
4. Didaktische Umsetzung der Unterrichtssequenz	16
4.1. Integration von <i>ObjectWriter</i> in die Unterrichtssequenz	16
4.2. Beobachteter Gewinn und Schwierigkeiten durch den Einsatz von <i>ObjectWriter</i>	20
4.3. Evaluation von <i>ObjectWriter</i> durch die Schüler	21
5. Zusammenfassung und Ausblick	24
A. Anhang	27
A.1. Bezugsmöglichkeiten des Programms <i>ObjectWriter</i>	27
A.2. Arbeitsauftrag – Die Attribute der Klasse ZEICHEN	28
A.3. Arbeitsauftrag – Die Attribute der Klasse ABSATZ	29
A.4. Fragebogen	30

1. Einleitung

Der Informatikunterricht in der 6. Jahrgangsstufe soll die Schüler langsam und auf eher spielerische Weise an das Konzept der objektorientierten Modellierung heranführen. Hierzu werden Programme zur Unterstützung verwendet, welche zu großen Teilen selbsterklärend sind und welche einige Schüler bereits kennen.

Zu Beginn des Schuljahres arbeiten die Schülerinnen und Schüler mit Grafikdokumenten und werden anhand dieser mit Objekten und Klassen vertraut gemacht. Zur Veranschaulichung der Objektorientierung dienen hier häufig die Werkzeuge *ObjectDraw* [13] und *EOS* [12]. Anhand dieser Programme lernen die Schüler einzelne Komponenten von Grafiken als Objekte mit Eigenschaften und Fähigkeiten zu begreifen und das Aussehen und die Position dieser Objekte durch Änderungen ihrer Eigenschaften zu manipulieren. Zudem wird die Punktnotation eingeführt, die als „Sprache“ der Informatik in ähnlicher Form auch in höheren objektorientierten Programmiersprachen verwendet wird.

Anhand der Objekte in Textdokumenten soll in einer weiteren Unterrichtssequenz das bereits erworbene Verständnis für die Objektorientierung vertieft werden. Zudem sollen die Schüler erkennen, dass zwischen verschiedenen Objekten Beziehungen bestehen können.

Wünschenswert wäre es hierbei, wenn die Schüler die bereits erworbenen Kenntnisse im neuen Zusammenhang anwenden könnten. In der Lernpsychologie wird eine derartige Anwendung von bereits Gelerntem auf eine neue Situation gleichen Komplexitätsgrades als lateraler Transfer [6] bezeichnet. Laut dem Psychologen Charles E. Osgood (1949) findet dieser umso leichter statt, je ähnlicher sich die Elemente in der Ausgangs- und der neuen Lernsituation sind [10]. Daher wäre es hilfreich, Klassen in Textdokumenten auf eine ähnliche Weise zu veranschaulichen, wie dies bereits in *ObjectDraw* geschah. Ein Programm, welches dies leistet, existierte bisher noch nicht.

Ziel dieser Arbeit war die Entwicklung und Erprobung eines Programms namens *ObjectWriter*, welches den Schülern hilft, Zeichen und Absätze in Textdokumenten als Objekte zu begreifen und den Zusammenhang zwischen Absätzen und den darin enthaltenen Zeichen zu veranschaulichen. Im Folgenden soll sowohl beschrieben werden, wie das Programm *ObjectWriter* konzipiert wurde, als auch welche Erfahrungen beim Einsatz der Software im Unterricht gemacht wurden.

Kapitel 2 gibt einen kurzen Überblick über den Themenkomplex „Informationsdarstellung mit Textdokumenten“. Dabei wird sowohl auf die Einbindung des Themas im bayerischen Gymnasiallehrplan eingegangen als auch dargestellt, auf welche Weise in der Regel bisher die Klassen ZEICHEN und ABSATZ didaktisch eingeführt wurden.

Im Rahmen dieser Arbeit wurde die Software *ObjectWriter* entwickelt. In Kapitel 3 werden die Überlegungen, die der Programmentwicklung zugrunde lagen, näher ausgeführt, sowie das fertige Programm und dessen einzelne Komponenten vorgestellt. Zudem werden die zu erwartenden Vorteile, welche sich aus der Nutzung des Programms bei der Einführung der Thematik ergeben, dargestellt.

ObjectWriter wurde im vergangenen Schuljahr in zwei Klassen der 6. Jahrgangsstufe des Neu-Ulmer Lessing-Gymnasiums ausführlich getestet. In Kapitel 4 werden die verwendete Unterrichtssequenz und deren didaktische Umsetzung beschrieben. Nach Abschluss der Unterrichtssequenz wurden die Erfahrungen der Schüler mit Hilfe eines Fragebogens evaluiert. Die Ergebnisse dieser Befragung werden in Kapitel 4 ebenso dargestellt wie die Beobachtungen der Lehrkraft während des Unterrichts.

In Kapitel 5 werden schließlich die im Rahmen des Projekts gesammelten Erfahrungen zusammengefasst. Da das Programm in Zukunft noch weiter entwickelt werden soll, wird auf mögliche und bereits umgesetzte Verbesserungen eingegangen.

2. Der Themenkomplex „Informationsdarstellung mit Textdokumenten“

Im Folgenden wird erläutert, wie der Themenkomplex „Informationsdarstellung mit Textdokumenten“ im bayerischen Lehrplan eingebettet ist, und auf welche Art die Inhalte in der Regel bisher vermittelt wurden.

2.1. Einbettung des Themenkomplexes „Informationsdarstellung mit Textdokumenten“ in den bayerischen Gymnasiallehrplan

In der 6. Jahrgangsstufe lernen die Schüler vermehrt Zusammenhänge systematisch zu betrachten, was insbesondere das „Ordnen, Strukturieren und Darstellen“ [16] von Informationen mit einschließt.

Im Schwerpunkt Informatik des Fachs „Natur und Technik“ wird den Schülern vermittelt, wie man Informationen mit Hilfe von Standardsoftware strukturiert darstellen kann. Dabei steht allerdings nicht der Umgang mit den verwendeten Programmen im Vordergrund, sondern die objektorientierte Modellierung. Durch diese Herangehensweise vermeidet man eine reine Produktschulung und vermittelt den Schülern vielmehr gemeinsame Grundprinzipien und Strukturen von Dokumenten. Dieses Wissen können die Schüler dann auch in anderen Fachbereichen anwenden.

Die Schüler haben zu Beginn des Schuljahres erkannt, dass sich Information auf verschiedene Arten darstellen lässt, und anhand von Grafiken bereits eine sehr aussagekräftige Form der Informationsdarstellung kennen gelernt. Bei der Beschäftigung mit den in Grafikdokumenten enthaltenen Objekten lernten sie, dass Objekte mit gleichen Attributen (Eigenschaften) und Methoden (Fähigkeiten) zu Klassen zusammengefasst werden können.

Mit Hilfe der speziell für den Unterricht konzipierten Software-Werkzeuge *Object-Draw* [13] und *EOS* [12] wurden den Schülern die Auswirkungen von Änderungen der Attributwerte durch Methodenaufrufe veranschaulicht. Mit Hilfe dieser Programme lernten sie außerdem die Punkt-Schreibweise, mit welcher Zustände und Methodenaufrufe einfach beschrieben werden konnten.

In einer weiteren Unterrichtssequenz sollen die Schüler nun das bisher Gelernte vertiefen, indem sie es auf Textdokumente anwenden. Sie erkennen, dass sich die objektorientierte Sichtweise auch auf Texte übertragen lässt. Gemeinsam mit dem Lehrer erarbeiten sie Attribute und Methoden der Klassen ZEICHEN und ABSATZ und erkennen darüber hinaus, dass zwischen den Objekten in Textdokumenten Beziehungen bestehen können. Zudem wird den Schülern bewusst, „dass viele alltägliche Zusammenhänge ebenfalls durch Beziehungen zwischen Objekten beschrieben werden können, diese Begriffe also eine allgemeinere Bedeutung haben.“ [16]

Durch praktische Arbeit am Computer werden diese neu gelernten Zusammenhänge vertieft.

2.2. Nachteile der bisherigen didaktischen Umsetzung

Bisher wurden die Klassen ZEICHEN und ABSATZ mit Hilfe von Textverarbeitungssoftware eingeführt. Die behandelten objektorientierten Konzepte wurden dabei durch ausführliche Arbeit mit gebräuchlichen Programmen wie *Microsoft Word* [9] oder *OpenOffice.org Writer* [11] vertieft. [15]

Bei dieser Herangehensweise konnte jedoch die Unterscheidung zwischen Objekten der Klasse ZEICHEN und denen der Klasse ABSATZ nur schwer verdeutlicht werden. So standen Methoden der Klasse ABSATZ – beispielsweise zur Änderung des Zeilenabstands oder der Ausrichtung – auch bei Markierung einzelner Zeichen zur Verfügung, was bei vielen Schülern den Eindruck hinterließ, dass diese Methoden zur Klasse ZEICHEN gehören. Ebenso konnten Methoden der Klasse ZEICHEN bei der Markierung ganzer Absätze aufgerufen werden. Dass es sich hierbei um eine implizite Umsetzung der später zu behandelnden Enthält-Beziehung zwischen den Klassen ZEICHEN und ABSATZ handelt, wurde bei der Verwendung von herkömmlichen Textverarbeitungsprogrammen ohne ausdrücklichen Hinweis des Lehrers nicht deutlich.

Textverarbeitungssoftware bietet die Möglichkeit, die Methoden der Klasse ZEICHEN – beispielsweise zur Änderung von Schriftart und Schriftgröße – auf ganze Wörter, Sätze oder Absätze anzuwenden. Dies ist zwar für den täglichen Gebrauch sinnvoll, sorgt aber bei Schülern nicht selten für Verwirrung, da es den Eindruck vermittelt, es gäbe weitere Objektklassen für Wörter und Sätze. Dass bei der Kursivsetzung eines Wortes die entsprechende Methode `KursivSetzen(wahr)` für jedes einzelne Zeichen des Wortes aufgerufen wird, erkennen die Schüler in der Regel nicht.

Die Enthält-Beziehungen innerhalb eines Textdokuments lassen sich Schülern zwar leicht vermitteln, jedoch wird der Zusammenhang zwischen einem einzelnen Zeichen und dem enthaltenden Absatz in herkömmlichen Textverarbeitungsprogrammen nicht deutlich. Die Möglichkeit, Methoden der Klasse ABSATZ scheinbar auf einzelne Zeichen – und umgekehrt Methoden der Klasse ZEICHEN auf ganze Absätze – anwenden zu können, sorgt eher dafür, dass sich den Schülern der Sinn einer solchen Beziehung nicht erschließt.

Ein entscheidender Nachteil des Einsatzes von Standard-Software bei der Vermittlung der objektorientierten Sichtweise ist die fehlende Anwendbarkeit der Punktnotation. Diese Schreibweise ist grundlegend für den Informatikunterricht in allen folgenden Jahrgangsstufen, kann jedoch bei Verwendung herkömmlicher Textverarbeitungssoftware nur theoretisch wiederholt werden. Motivierender und für einen langfristigen Lernerfolg förderlicher wäre eine praktische Anwendung am Computer.

Aus den oben genannten Gründen entstand daher der Wunsch nach einer Software, welche die Unterscheidung zwischen den Objekten der Klasse ZEICHEN und denen der Klasse ABSATZ deutlicher hervorhebt, sowie eine Anwendung der Punktschreibweise erlaubt, um diese im neuen Zusammenhang zu verwenden und den Schülern den damit verbundenen lateralen Lerntransfer zu ermöglichen. Da die Schüler in Zukunft herkömmliche Textverarbeitungssoftware verwenden werden, sollte das zu entwickelnde Werkzeug einen möglichst leichten Umstieg auf derartige Programme gestatten.

Das im folgenden Kapitel vorgestellte Software-Werkzeug *ObjectWriter* ist das Ergebnis der Bemühung, die oben beschriebenen Überlegungen umzusetzen.

3. Das Programm *ObjectWriter*

Um Lehrern ein Unterrichts-Werkzeug zur Verfügung zu stellen, mit welchem sie Schülern die Klassen ZEICHEN und ABSATZ deutlicher veranschaulichen können, wurde im Rahmen der vorliegenden Arbeit ein Programm namens *ObjectWriter* erstellt. Die Konzeption dieser Software, sowie deren einzelne Komponenten werden im Folgenden beschrieben.

3.1. Konzeption des Programms *ObjectWriter*

Das Programm *ObjectWriter* sollte optisch und funktionell stark an *ObjectDraw* [13] (s. Abb. 1) angelehnt sein. *ObjectDraw* vermittelte den Schülern zu Beginn des Schuljahres einen ersten Eindruck von Objekten in Grafikdokumenten. Um den Schülern den Transfer zu erleichtern, sollte das Programmdesign von *ObjectDraw* größtenteils beibehalten werden, einige Komponenten sollten bei *ObjectWriter* zum besseren Verständnis aber hinzugefügt oder anders angeordnet werden.

Ebenso wie das Programm *ObjectDraw* sollte *ObjectWriter* aus einem Hauptfenster bestehen, in welchem die Objekte angezeigt werden, sowie einem Fenster für Objekt- und Klassenkarten und einem Fenster, welches die bereits getätigten Methodenaufrufe in Punktnotation anzeigt.

Die Option, sich den Objektbaum anzeigen zu lassen, sollte erhalten bleiben. Aufgrund der großen Anzahl an Objekten und der sehr flachen Hierarchie wird der Objektbaum in der Regel sehr unübersichtlich sein, zur Veranschaulichung der Struktur eines Textdokuments kann er jedoch hinzugezogen werden.

Ebenfalls zur Veranschaulichung der Enthält-Beziehung sollte im Objektdiagramm sowohl das ausgewählte Zeichen dargestellt werden, als auch der Absatz, der die-

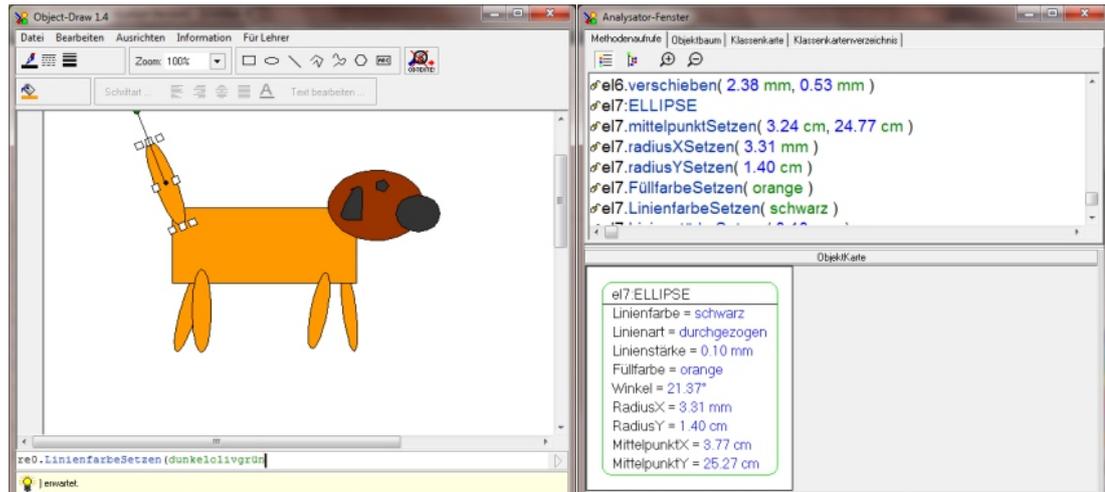


Abbildung 1.: Das Programm *ObjectDraw* zur Veranschaulichung der Objekte in Grafikdokumenten

ses enthält. Die Beziehung zwischen Zeichen und Absatz sollte im Objekt- und im Klassendiagramm grafisch veranschaulicht werden.

Um die Unterscheidung zwischen Objekten der Klasse ZEICHEN und Objekten der Klasse ABSATZ zu erleichtern, sollte ein einfaches Farbkonzept verwendet werden: Zeichen werden blau markiert, Absätze grün. Dies gilt sowohl bei der Markierung im Textdokument als auch im Objekt- und Klassendiagramm.

Methodenaufrufe sollten sowohl mit Hilfe von Buttons als auch durch eine Texteingabe möglich sein. Anders als bei herkömmlichen Textverarbeitungsprogrammen sollten die Buttons für Zeichenmethoden und diejenigen für Absatzmethoden deutlich getrennt werden.

Gerade zu Beginn der Unterrichtssequenz sollen die Schüler mit Hilfe der Texteingabe die Punktnotation wiederholen. Damit die Schüler sich die Arbeit in den Freiarbeitsphasen nicht mit Hilfe der Schaltflächen erleichtern, sollte es eine Möglichkeit geben, diese zu deaktivieren. Wünschenswert wäre eine Umsetzung, welche es nur dem Lehrer erlaubt, die Buttons zu reaktivieren.

Der Großteil der oben beschriebenen Funktionalität wurde bereits in einer ersten Version des Programms *ObjectWriter* umgesetzt. Die einzelnen Komponenten der Software werden im nächsten Abschnitt vorgestellt.

3.2. Erläuterung der einzelnen Programmkomponenten

Das Fenster von *ObjectWriter* ist in fünf Bereiche unterteilt (s. Abb. 2). Um alle Bereiche im Blick haben zu können, wurden die Bereiche – anders als bei *ObjectDraw* – in einem einzigen Fenster fixiert. Im Folgenden soll die Funktionsweise der einzelnen Bereiche erläutert werden:

Das **Textfenster (1)** zeigt den zu bearbeitenden Text an. Das Programm stellt verschiedene Texte zur Verfügung, welche über das Menü geöffnet werden können. Bei einem Methodenaufruf sind dessen Folgen umgehend im Textfenster erkennbar. Ein wesentlicher Unterschied zu herkömmlichen Textverarbeitungsprogrammen liegt in der Markierung von Zeichen und Absatz. In *ObjectWriter* kann immer nur ein einzelnes Zeichen markiert werden. Diese Markierung wird durch ein blaues Kästchen um das Zeichen gekennzeichnet. Zugleich wird der Absatz, welcher dieses Zeichen enthält, durch einen grünen Kasten markiert. Dadurch wird deutlich, welche Bereiche von den entsprechenden Methodenaufrufen betroffen sind.

Um den Benutzern die Identifizierung der Zeichen und Absätze zu verdeutlichen, erscheint ein Hinweis mit Zeichen- und Absatzbezeichner neben jedem Zeichen, sobald man mit dem Mauszeiger darauf zeigt. Dies hat den Vorteil, dass ein Benutzer ein Zeichen oder einen Absatz formatieren kann, ohne das entsprechende Objekt zuvor markieren und dessen Bezeichner im Objektdiagramm ablesen zu müssen.

Die Schüler sind es gewohnt, einen Zeilenumbruch mittels der *Return*-Taste zu erreichen, was stets einen neuen Absatz zur Folge hat. Die Möglichkeit, einen Zeilenumbruch innerhalb eines Absatzes mittels der Tastenkombination *Shift+Return* herbeizuführen, ist den Schülern in der Regel nicht bekannt. Um die Schüler nicht zu verwirren, wurde auch bei den in *ObjectWriter* integrierten Texten bei jedem Zeilenumbruch ein neuer Absatz erzeugt, auch wenn in einzelnen Fällen ein einzelner Absatz mit Zeilenumbrüchen mehr Sinn gemacht hätte.¹

¹Beispiele hierfür wären beispielsweise das Gedicht in „Rumpelstilzchen“ oder die Befragung des Spiegels in „Schneewittchen“.

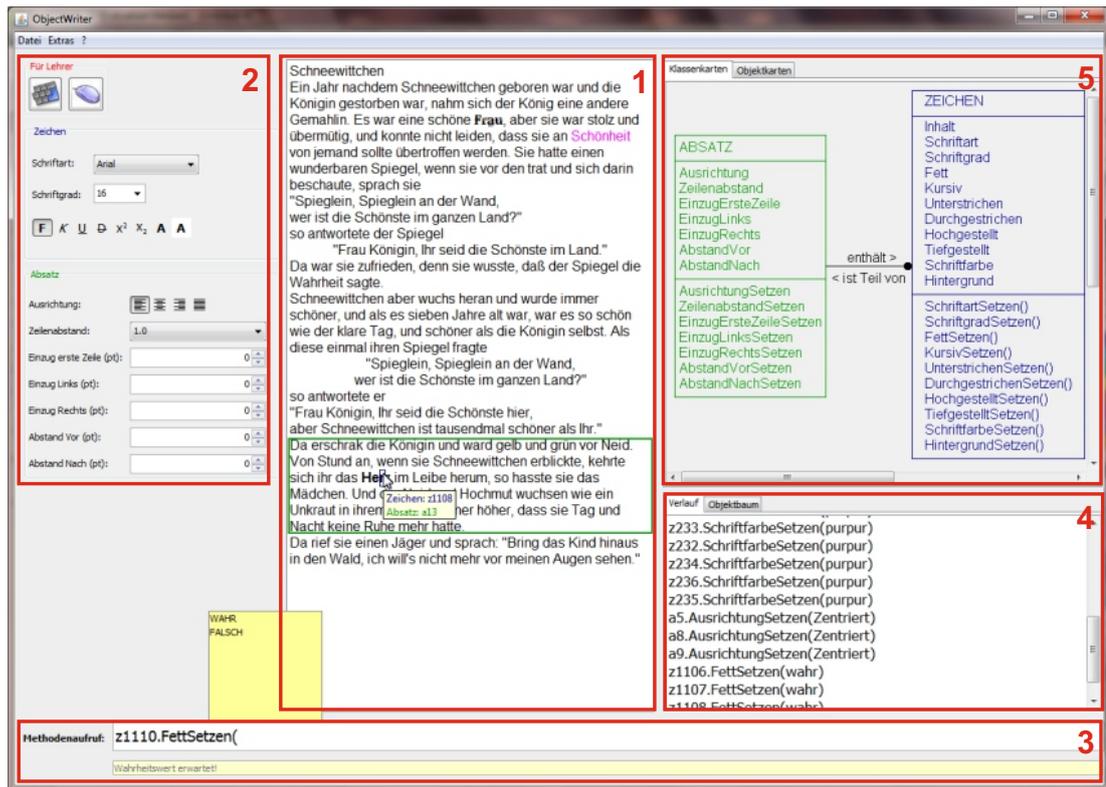


Abbildung 2.: Das Programm *ObjectWriter* zur Veranschaulichung der Objekte in Textdokumenten

Im **Schaltflächen-Fenster (2)** stehen verschiedene Buttons für Methodenauf-rufe zur Verfügung. Anders als bei herkömmlichen Textverarbeitungsprogrammen sind die Methodenbuttons für Absätze und Zeichen deutlich getrennt und durch einen Rahmen den entsprechenden Klassen zugewiesen. Die Symbole auf den Schaltflächen entsprechen denjenigen in herkömmlichen Textverarbeitungsprogrammen, um einen späteren Umstieg auf Standardsoftware zu erleichtern. Um den Zusammenhang zwischen Button und zugehörigem Methodenaufruf noch deutlicher zu machen, erscheinen bei Berühren der Knöpfe mit dem Mauszeiger Tooltips mit dem Bezeichner der entsprechenden Methode.

In *ObjectWriter* können die Schüler Methoden aber nicht nur über einen Button, sondern auch mit Hilfe der Texteingabe aufrufen. So kann der Nutzer Methodenauf-rufe in Punktnotation in das **Methodeneingabefenster (3)** schreiben. Dieses Vorgehen sollte zu Beginn forciert werden, um die Punktschreibweise in einem neuen Zusammenhang zu wiederholen und den Schülern so den Transfer zu ermöglichen. Die Eingabe wird durch eine Zeile mit Hinweisen unterhalb der Eingabezeile unterstützt. Anders als bei *ObjectDraw* werden den Schülern während

der Eingabe jedoch keine möglichen Methoden angeboten, sondern die Schüler sollen die passenden Bezeichner mit Hilfe des Klassendiagramms oder der Tool-tips selbst finden. Bei Eingabe eines gültigen Methodenbezeichners können die Schüler jedoch in der Regel aus verschiedenen Eingabewerten wählen, so dass es nicht nötig ist, beispielsweise die Bezeichnungen aller Schriftarten zu kennen.

Der **Reportbereich (4)** dient der Protokollierung der Methodenaufrufe und der Anzeige des Objektbaums. Da jedes einzelne Zeichen ein Objekt darstellt, werden die Einträge im Objektbaum in der Regel sehr umfangreich sein, was zu einer gewissen Unübersichtlichkeit führen kann. Allerdings kann der Objektbaum den Schülern die Enthält-Beziehung zwischen Objekten der Klassen TEXTDOKUMENT, ABSATZ und ZEICHEN verdeutlichen.

Bei der Arbeit mit *ObjectWriter* werden sämtliche getätigte Methodenaufrufe – egal, ob per Schaltfläche oder per Texteingabe – protokolliert. Dieses Protokoll soll dazu beitragen, den Schülern zu verdeutlichen, welche Methoden bei Betätigung einer Schaltfläche ausgeführt werden. Die Schriftgröße des Protokolls kann der Lehrer – wie im Internetbrowser oder in Office-Anwendungen üblich – durch Drücken der Strg-Taste bei gleichzeitiger Betätigung des Scrollrads der Maus ändern. Dies kann insbesondere für die Präsentation am Beamer hilfreich sein.

Der **Diagrammbereich (5)** zeigt wahlweise das Klassen- oder das Objektdiagramm. Im Objektdiagramm werden die Objektkarten der gerade im Textfenster ausgewählten Objekte dargestellt. Neben der Enthält-Beziehung zwischen dem Zeichen und dem zugehörigen Absatz dient das Objektdiagramm vor allem der Veranschaulichung der Auswirkungen eines Methodenaufrufs auf die Attributwerte der ausgewählten Objekte.

Das Klassendiagramm kann einerseits zur Information über verfügbare Methoden dienen, andererseits veranschaulicht es die Enthält-Beziehung zwischen der Klasse ABSATZ und der Klasse ZEICHEN.

Sowohl beim Klassendiagramm als auch beim Objektdiagramm wurde das im Textfenster verwendete Farbkonzept beibehalten, um die Unterscheidung zwischen ZEICHEN und ABSATZ deutlich zu machen.

Um den Schülern die Möglichkeit zu nehmen, Methoden lediglich mit Hilfe der Buttons aufzurufen, können mit Hilfe eines „Lehrermodus“ (im Menü: „Extras“ → „Lehrermodus“) Maus oder Tastatur deaktiviert werden.

3.3. Erwarteter Beitrag des Programms zum Verständnis der Unterrichtsinhalte

Bisher wurden die Klassen ZEICHEN und ABSATZ mit Hilfe herkömmlicher Textverarbeitungssoftware eingeführt. Den Schülern fiel bei dieser Herangehensweise oft die Unterscheidung der beiden Klassen schwer. Durch eine konsequente Trennung der Methoden der beiden Klassen und durch eine grafische Darstellung sowohl der Zeichen als auch der Absätze, in welchen diese enthalten sind, soll den Schülern der Unterschied der beiden Klassen deutlich gemacht werden. Zudem werden die Auswirkungen von Methodenaufrufen sowohl am Text als auch im Objektdiagramm deutlich gemacht und liefern den Schülern somit eine Rückmeldung über die Folgen ihrer Aktionen.

Dass man in herkömmlicher Textverarbeitungssoftware mehrere Zeichen gleichzeitig markieren kann, trägt zwar in großem Maße zum praktischen Arbeiten bei, verschleiert aber die Tatsache, dass die entsprechenden Methoden jedes einzelnen markierten Zeichens beziehungsweise jedes einzelnen markierten Absatzes aufgerufen werden. Durch *ObjectWriter* wird dies den Schülern nun deutlich gemacht, da nur jeweils ein einzelnes Zeichen und ein einzelner Absatz markiert werden können.

Die Punktschreibweise wurde bei der ausschließlichen Verwendung herkömmlicher Textverarbeitungsprogramme vernachlässigt. Da die textbasierte Eingabe von Methodenaufrufen in Standardsoftware zur Textverarbeitung nicht möglich ist, konnte die Punktschreibweise nur theoretisch wiederholt werden, was bei vielen Schülern zu einer mangelnden Motivation führte. Dass die Buttons in Textverarbeitungssoftware lediglich als Schnittstellen dienen, bei deren Betätigung die Methoden implizit aufgerufen werden, kann den Schülern durch die parallele Darstellung der aufgerufenen Methoden in *ObjectWriter* anschaulich vermittelt werden.

4. Didaktische Umsetzung der Unterrichtssequenz

Nach der Entwicklung von *ObjectWriter* wurde das Programm im Unterricht in zwei Klassen der 6. Jahrgangsstufen getestet. Im nachfolgenden Kapitel werden die dafür erstellte Unterrichtssequenz und die Erfahrungen, die mit dem Einsatz von *ObjectWriter* gemacht wurden, zusammengefasst.

4.1. Integration von *ObjectWriter* in die Unterrichtssequenz

Die Verwendung von *ObjectWriter* kann vor allem bei der Einführung der Klassen ZEICHEN und ABSATZ sowie zur Veranschaulichung der Enthält-Beziehung sinnvoll sein. Daher soll im Nachfolgenden vorrangig auf diese Bereiche eingegangen werden. Abgesehen von den Unterrichtsstunden, in welchen *ObjectWriter* eingesetzt wurde, orientiert sich die Konzeption des Unterrichts stark der ISB-Handreichung [15]. Durch die Einführung einer zusätzlichen Software und das langsamere Arbeiten in *ObjectWriter* war es notwendig, den Umfang der Unterrichtssequenz „Informationsdarstellung mit Textdokumenten“ von 8 auf 9 Unterrichtsstunden zu erhöhen. Die Erfahrung im vergangenen Schuljahr hat aber gezeigt, dass sich bei der Behandlung der Enthält-Beziehung zukünftig sicher Zeit einsparen lässt, da diese in *ObjectWriter* von Beginn der Unterrichtssequenz an deutlich veranschaulicht wird. Dies führte dazu, dass die Schüler den Zusammenhang zwischen den Objekten der Klassen ZEICHEN und ABSATZ schnell erfassten und ähnliche Beziehungen auch leicht zwischen anderen Objekten ihrer Erfahrungswelt erkennen konnten.

Die sich hieraus ergebende Unterrichtssequenz soll hier nur skizziert werden, lediglich Stunden, in denen *ObjectWriter* verwendet wurde, werden detaillierter beschrieben.

1. Stunde: Darstellung von Informationen durch Texte

Die Schüler haben bereits zu Schuljahresbeginn erfahren, dass bei der ausschließlich grafischen Darstellung von Sachverhalten Information verloren gehen kann und dass Fakten durch Texte oft detaillierter ausgedrückt werden können. Anhand einiger verschieden formatierter Textdokumente erkennen die Schüler den Zusammenhang zwischen der Formatierung eines Textes und dessen Informationsgehalt. [15] Mit Hilfe eines herkömmlichen Textverarbeitungsprogramms sollen die Schüler ein eigenes knappes Textdokument entwerfen und formatieren. Einige der dabei entstehenden Textdokumente werden abschließend der Klasse präsentiert, die darin enthaltenden Objekte werden aufgrund einzelner Formatierungsmerkmale identifiziert. Die Schüler sollten einzelne Zeichen als Objekte identifizieren, eventuell kann auch bereits die Klasse ABSATZ angesprochen werden.

2. Stunde: Die Klasse ZEICHEN

In der vorangegangenen Stunde haben die Schüler wahrscheinlich Buchstaben, Zahlen und Satzzeichen als Objekte identifiziert. Um ihnen die Abstraktion, die schließlich zur Klasse ZEICHEN führt, zu erleichtern, sollen die Schüler in Partnerarbeit mögliche Attribute verschiedener Zeichen finden. Dabei sollen sie erkennen, dass die Attribute eines Buchstaben dieselben sind wie diejenigen einer Zahl oder eines Sonderzeichens. Diese Erkenntnis führt zur allgemeineren Klasse ZEICHEN. Die Methoden dieser Klasse dienen hauptsächlich der Manipulation der Attributwerte und sollten daher leicht von den Schülern selbst benannt werden können. Die so erarbeitete Klassenkarte der Klasse ZEICHEN wird schließlich in das Heft übertragen.

Nun erhalten die Schüler eine kurze Einführung in das Programm *ObjectWriter*. Dabei soll insbesondere auf die Übereinstimmung zwischen den markierten Objekten im Textfenster und den Attributwerten im Objektdiagramm hingewiesen werden. Anhand einiger Beispiele wird die Punknotation wiederholt und die Funk-

tionsweise des Methodeneingabefensters vorgeführt. Der Text „Die zwei Freunde und der Bär“¹ [1] dient den Schülern als Übung. Die dazu notwendigen Methoden sollten in diesem frühen Stadium vorzugsweise per Texteingabe aufgerufen werden, um einerseits die Punktschreibweise zu vertiefen, andererseits den Unterschied zwischen Bezeichner (z.B. „z173“) und Inhalt (z.B. „a“) eines Objekts der Klasse ZEICHEN verdeutlichen.

In *ObjectWriter* kann stets nur ein einzelnes Zeichen formatiert werden, was besonders beim Methodenaufruf per Texteingabe sehr aufwändig ist. Daher sollen beim Arbeitsauftrag nur wenige kurze Wörter mit Hilfe der Punktnotation formatiert werden. Um die Ergebnisse zu sichern, sollen die Kinder die Methodenaufrufe auf dem Arbeitsblatt notieren. Schüler, welche die ersten vier Aufgaben schnell gelöst haben, können fakultativ die restlichen Aufgaben auf dem Arbeitsblatt mit Hilfe der Buttons bearbeiten. Am Ende der Stunde sollte in jedem Fall noch Zeit für eine Besprechung und eine eventuelle Verbesserung der ersten vier Aufgaben zur Verfügung stehen.

3. Stunde: Die Klasse ABSATZ

Den Schülern ist vermutlich bewusst, dass mehrere Zeichen zu einem Absatz zusammengefasst werden. Anhand eines Beispieltextes, dessen Absätze verschiedene Attributwerte aufweisen, (z.B. aus dem Lehrbuch oder aus dem gerade behandelten Thema im Fach Biologie) sollen sie erkennen, dass durch Absätze „zusammengehörige Gedanken gruppiert und Texte inhaltlich gegliedert werden“ [15]. Anders als ein Satz oder ein Wort verfügt ein Absatz über Eigenschaften, welche nicht durch die Attribute der enthaltenen Zeichen ausgedrückt werden können. Die Schüler erkennen anhand des Textes einige Attribute der Klasse ABSATZ und übertragen diese gemeinsam mit den zusammengehörigen Methoden in ihr Heft. Als Beispieltext sollte bewusst ein Text gewählt werden, bei welchem in einigen Absätzen alle Zeichen – beispielsweise durch Fettdruck – formatiert wurden, um die Schüler explizit auf die Unterscheidung zwischen Absatz- und Zeichenattributen aufmerksam zu machen.

Die Schüler bearbeiten den Text „Absätze“² [15] in *ObjectWriter*. Dabei sollen sie jeden Absatz so formatieren, dass seine Absatzattribute dem Text entsprechen.

¹s. Anhang A.2: „Arbeitsauftrag – Die Attribute der Klasse ZEICHEN“

²s. Anhang A.3: „Arbeitsauftrag – Die Attribute der Klasse ABSATZ“

4. bis 6. Stunde: Vertiefung der Klassen ZEICHEN und ABSATZ

Die Schüler sollten mittlerweile die Attribute der Klasse ZEICHEN und diejenigen der Klasse ABSATZ unterscheiden können. In drei Schulstunden soll mittels Übungen die Unterscheidung zwischen beiden Klassen weiter verinnerlicht werden. Dabei können die Stunden analog zur ISB-Handreichung [15] konzipiert werden. Da – besonders die textbasierten – Methodenaufrufe in *ObjectWriter* sehr aufwändig sind, würde eine Verwendung dieses Programms für umfangreichere Arbeitsaufträge zu viel Zeit in Anspruch nehmen und die Schüler zudem demotivieren. Es ist daher anzuraten, die weiteren Arbeitsaufträge in einem herkömmlichen Textverarbeitungsprogramm wie *Microsoft Word* [9] oder *OpenOffice.org Writer* [11] zu bearbeiten.

Um den Schülern den Umstieg auf das verwendete Standard-Textverarbeitungsprogramm zu erleichtern, sollte eine kurze Einführung in die verwendete Software gegeben werden. Insbesondere das Formatierungsmenü mit den Untermenüs „Zeichen“ und „Absatz“ sollte hervorgehoben werden, wobei im Rahmen des Unterrichtsgesprächs auf die in herkömmlichen Programmen sehr unscharfe Trennung zwischen ZEICHEN- und ABSATZ-Methoden eingegangen werden kann. Auch die Vorteile einer Markierung mehrerer – auch nicht zusammenhängender – Zeichen sollte angesprochen werden. Dabei sollte man die Schüler jedoch eindringlich darauf hinweisen, dass auch bei der Markierung mehrerer Zeichen oder Absätze die Methoden implizit für jedes einzelne markierte Objekt aufgerufen werden.

Als praktischer Arbeitsauftrag bietet sich der in der ISB besprochenen Text „Katze und Maus“ [15] sowie die Arbeitsblätter „Zeichen- und Absatzattribute“ [15] und „Aufgabe zu Zeichen- und Absatzattributen“ [15] an.

Weder das Markieren mehrerer Zeichen noch das Kopieren und Verschieben von Text sind in *ObjectWriter* vorgesehen. Daher wurden die zugehörigen Methoden auch nicht in den entsprechenden Klassenkarten des Programms notiert. Wenn den Schülern diese Fertigkeiten mit Hilfe eines Standard-Textverarbeitungsprogramms vermittelt werden, sollten diese explizit auf die Existenz der entsprechenden Methoden hingewiesen werden. Es empfiehlt sich, die bereits im Heft angelegten Klassenkarten zu ergänzen, beziehungsweise die Methoden `Markieren()`, `Kopieren()`, `Verschieben()` und `Löschen()` bereits vorgreifend an entsprechender Stelle in das Diagramm mit aufzunehmen.

Als Arbeitsauftrag zum Üben von Kopier- und Verschiebevorgängen eignet sich das in der ISB-Handreichung verwendete Gedicht „fünfter sein“ von Ernst Jandl. [7]

7. bis 9. Stunde: Die Enthält-Beziehung und die Klasse TEXTDOKUMENT

Analog zur ISB-Handreichung [15] können die Enthält-Beziehung und die Klasse TEXTDOKUMENT im Unterricht behandelt werden. Die Enthält-Beziehung wurde den Schülern bereits während der Arbeit mit *ObjectWriter* durch die Markierung von Zeichen und Absatz im Text sowie durch die Klassen- und Objektdiagramme deutlich vor Augen geführt. Daher bietet es sich an, diese Beziehung mit Hilfe von *ObjectWriter* einzuführen. Eine Verwendung von *ObjectWriter* zur Bearbeitung der von der ISB vorgeschlagenen Arbeitsaufträge bringt jedoch keinen deutlichen Mehrwert.

4.2. Beobachteter Gewinn und Schwierigkeiten durch den Einsatz von *ObjectWriter*

Die zuvor beschriebene Unterrichtssequenz kam in zwei Klassen der 6. Jahrgangsstufe zur Anwendung. Dabei zeigten sich einige Nachteile bei der Verwendung von *ObjectWriter*, die im Folgenden zusammengefasst werden.

Da viele Schüler bereits Erfahrungen mit herkömmlichen Textverarbeitungsprogrammen gesammelt hatten, kam einigen die Arbeit mit *ObjectWriter* offensichtlich wie ein Rückschritt vor. Insbesondere die Tatsache, dass nur einzelne Zeichen bzw. Absätze markiert werden konnten, ließ ihnen das Programm unnötig umständlich vorkommen. Da zu Beginn der Unterrichtssequenz großer Wert darauf gelegt wurde, dass die Schüler die Tastatureingabe für Methodenaufrufe verwendeten, stieg der Aufwand für die Markierung selbst eines kurzen Wortes weiter an. Dies führte bei einigen Schülern offensichtlich zu einer abnehmenden Motivation, was sich nicht zuletzt dadurch bemerkbar machte, dass sie heimlich mit Hilfe des „Lehrer-Modus“ von der Tastatureingabe auf die Eingabe mit Hilfe der Buttons umstiegen.

Einige Schüler bemängelten, dass sich der aktuelle Stand der von ihnen bearbeiteten Texte nicht speichern ließ. Da die Arbeitsaufträge in der Regel kurz waren, stellte sich dies zwar als ein nicht allzu großer Nachteil dar, dennoch wäre es für spätere Versionen sinnvoll, ein Speichern der Texte zu ermöglichen. Dies würde den Schülern einerseits nicht das Gefühl geben, dass die von ihnen geleistete Arbeit sofort verworfen würde, andererseits könnten nicht abgeschlossene Arbeitsaufträge in nachfolgenden Schulstunden vervollständigt werden.

Trotz dieser Nachteile trug *ObjectWriter* scheinbar nicht unwesentlich zum Verständnis des zu vermittelnden Stoffs bei.

Der hohe Aufwand beim Aufruf der Methoden schien den Spaß, den die Schüler bei der Arbeit mit *ObjectWriter* hatten, nicht zu mindern. Zudem war ein Großteil der Schüler nach der Unterrichtssequenz in der Lage die Attribute der Klassen ZEICHEN und ABSATZ zu unterscheiden und Methodenaufrufe den jeweiligen Klassen zuzuordnen.

Insbesondere durch die Veranschaulichung der Folgen von Methodenaufrufen in Punktschreibweise gelang den Schülern der Transfer des bereits Gelernten von Grafik- hin zu Textdokumenten. Dass durch den Einsatz von *ObjectWriter* die Punktnotation weiter verinnerlicht wurde, zeigte sich auch im weiteren Verlauf des Schuljahres bei der Behandlung von Multimediadokumenten. Hier waren die Schüler ohne vorherige Wiederholung in der Lage, Methodenaufrufe in Punktnotation korrekt aufzuschreiben.

Auch zum Verständnis der Enthält-Beziehung trug deren grafische Darstellung in *ObjectWriter* offensichtlich bei. Die Schüler erkannten eine derartige Beziehung zwischen Zeichen und Absätzen gleich zu Beginn der Unterrichtssequenz und konnten auch in späteren Unterrichtseinheiten derartige oder ähnliche Beziehungen zwischen Klassen identifizieren.

4.3. Evaluation von *ObjectWriter* durch die Schüler

Um zu evaluieren, inwieweit die Schüler einen Gewinn durch die Verwendung von *ObjectWriter* hatten, wurde am Ende der Unterrichtssequenz ein Fragebogen an die Schüler verteilt (siehe Anhang A.4). Da die bei einem einstündigen Fach

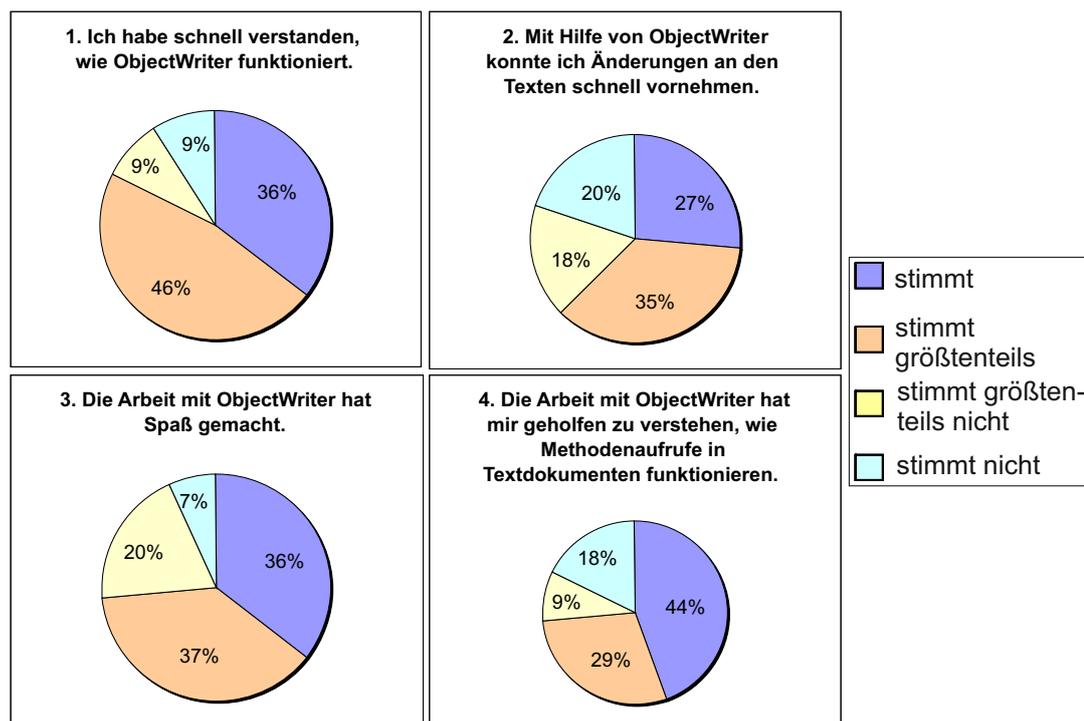


Abbildung 3.: Ergebnisse der Evaluation durch die Schüler

ohnehin knappe Unterrichtszeit durch zahlreiche Stundenausfälle weiter verringert wurde, sollte die Befragung der Schüler möglichst wenig Zeit in Anspruch nehmen. Der Fragebogen wurde aus diesem Grund bewusst knapp gehalten.

Neben grundlegenden Fragen zur Bedienbarkeit des Programms sollten die Schüler auch eine Einschätzung abgeben, in welchem Maße ihnen *ObjectWriter* geholfen hatte, zu verstehen, wie Methodenaufrufe in Textdokumenten funktionieren.

In einem zweiten Teil konnten die Schüler in eigenen Worten angeben, welche Vorteile sie in *ObjectWriter* sahen und was an dem Programm ihrer Meinung nach verbessert werden könnte.

Die Antworten der Schüler decken sich zu großen Teilen mit den Beobachtungen während des Unterrichts. Mehr als ein Drittel der Schüler empfanden das Formatieren jedes einzelnen Zeichens mit Hilfe der Texteingabe auf Dauer als zu mühsam. Dies tat jedoch dem Spaß, den ein Großteil der Schüler bei der Arbeit mit *ObjectWriter* hatte, offenbar keinen Abbruch (vgl. Abb. 3).

Die Möglichkeit, bereits formatierte Texte zu speichern, vermissten einige Schüler. Die Speicheroption war auch diejenige Funktionalität, welche sich die meis-

ten Schüler für eine zukünftige Programmversion wünschten. Vereinzelt wurde bemängelt, dass man sich die Bezeichner der verfügbaren Methoden auswendig merken müsste. In Zukunft sollten daher die Schüler ausdrücklich darauf hingewiesen werden, dass die Bezeichner bei angezeigtem Klassendiagramm permanent präsent sind.

Als Vorteile von *ObjectWriter* nannten die Schüler vor allem die vielen unterschiedlichen Möglichkeiten zur Formatierung der Texte, sowie die Einfachheit der Bedienung. Auch wurde oft angegeben, dass die Methodenaufrufe in *ObjectWriter* anschaulich dargestellt und somit verständlich waren. Dass – im Gegensatz zu *ObjectDraw* – die Objektbezeichner mittels Tooltip angezeigt wurden, war ebenfalls einigen Schülern positiv aufgefallen.

5. Zusammenfassung und Ausblick

Die 50 Schüler der beiden 6. Klassen des Lessing-Gymnasiums arbeiteten über einen Zeitraum von mehreren Wochen mit dem Programm *ObjectWriter*. Sowohl die Beobachtung der Schüler im Unterricht als auch die Auswertung der Fragebögen lieferten wertvolle Informationen, in welchem Maße das Programm sinnvoll eingesetzt werden kann und wie das Programm in Zukunft noch weiter an die Bedürfnisse von Lehrern und Schülern angepasst werden kann.

Zusammenfassend lässt sich sagen, dass die Arbeit mit *ObjectWriter* bei den Schülern erkennbar zum Verständnis des behandelten Unterrichtsstoffes beitrug: Objekte der Klassen ZEICHEN und ABSATZ wurden im weiteren Unterrichtsverlauf sicher identifiziert, Methoden konnten den richtigen Objekten zugeordnet werden. Die Schüler konnten die Punktnotation an ausführlichen und praxisnahen Beispielen wiederholen und auf einen neuen Sachzusammenhang anwenden. Besonders die Tatsache, dass die Effekte von über die Texteingabe eingegebenen Methodenaufrufen sofort im Text und an den Objektkarten sichtbar wurden, trug erheblich zur Motivation und zur Unterscheidung von Zeichen und Absätzen bei. Gerade letztere Unterscheidung ist in herkömmlichen Textverarbeitungsprogrammen nicht klar getrennt, was bei den Schülern nicht selten für Verwirrung sorgt. Die Enthält-Beziehung wurde aufgrund der konsequenten grafischen Darstellung in *ObjectWriter* von Anfang an erkannt und konnte später sicher auf andere Beispiele des alltäglichen Lebens angewendet werden.

Für den beschriebenen Mehrwert sollte die Tatsache, dass die Methodenaufrufe mit Hilfe der Tastatur sehr aufwändig sind, in Kauf genommen werden. Offenbar minderte dies ohnehin bei einem Großteil der Schüler die Freude an der Arbeit mit dem Programm nicht.

Allerdings sollte darauf geachtet werden, dass *ObjectWriter* lediglich für den Einstieg in den Themenkomplex „Informationsdarstellung mit Textdokumenten“ ver-

wendet wird. Die mühsame Formatierung jedes einzelnen Zeichens oder Absatzes kann sonst auf Dauer durchaus demotivierend wirken. Zudem sollen Schüler auch den Transfer auf herkömmliche Textverarbeitungsprogramme leisten und mit diesen umzugehen lernen. Wichtig ist, dass den Schülern durch die Verwendung von *ObjectWriter* bewusst wird, dass auch in Standard-Textverarbeitungssoftware bei der Markierung mehrerer Zeichen die entsprechenden Methoden für jedes einzelne Objekt aufgerufen werden.

Trotz des erfolgreichen Einsatzes von *ObjectWriter* im Unterricht kann das Programm in einigen Punkten noch verbessert werden (vgl. Kapitel 4).

Die in dieser Arbeit beschriebene Programmversion 1.0 wurde bereits an einigen Stellen angepasst. So musste der Großteil der Texte auf die neue deutsche Rechtschreibung umgestellt werden, einige Rechtschreibfehler in den Menüs wurden korrigiert. Die meisten vom Programm zur Verfügung gestellten Texte sind im Rahmen des Projekts Gutenberg [14] frei verfügbar, evtl. bestehende Urheberrechte wurden im Dialog „Über *ObjectWriter*“ (Programm-Menü „?“ → „Über...“) vermerkt.

Die Funktionalität der Schaltflächen zum Setzen von Schriftfarbe und Hintergrundfarbe ließ sich bisher nicht leicht erschließen. Dies lag daran, dass die Schriftfarbe standardmäßig schwarz, die Hintergrundfarbe standardmäßig weiß ist. Da dies der Darstellung aller anderen Schaltflächen entspricht, hatten beide Buttons bei Programmstart beinahe identisches Aussehen mit dem Button für Fettdruck. Um die Funktionalität zu verdeutlichen, wurde beiden Knöpfen eine entsprechende Beschriftung vorangestellt.

Um den Zusammenhang zwischen den Buttons und den damit verbundenen Methodenaufrufen zu verdeutlichen, wurden alle Buttons mit Tooltips versehen, welche die Bezeichner der mit den Schaltflächen verknüpften Methoden anzeigen.

Die oben beschriebenen Änderungen wurden bereits in der aktuellen Version 1.1 des Programms implementiert.

Basierend auf der Version 1.1 sind zukünftig weitere Änderungen geplant:

Wenn Schüler im Rahmen des Unterrichts praktische Arbeit leisten, möchten sie, dass diese auch nach Stundenende erhalten bleibt. Daher soll eine Speicherfunktio-

on implementiert werden, um die Motivation der Schüler zu erhöhen. Dies würde auch ein Anknüpfen an Arbeitsaufträge erleichtern, die in der vergangenen Stunde nicht vollständig bearbeitet werden konnten.

Die erweiterte Punktnotation, wie in der ISB-Handreichung [15] beschrieben, wird von *ObjectWriter* bisher noch nicht unterstützt. Daher sollen in Zukunft die Tool-tips, welche bisher die Bezeichner von Absatz und Zeichen anzeigen, so erweitert werden, dass auch die Position eines Zeichens innerhalb seines Absatzes angezeigt wird.

Der Methodenaufruf mit Hilfe der Texteingabe leistet einen wertvollen Beitrag bei der Wiederholung der Punktnotation. Um zu verhindern, dass die Schüler heimlich die Maussteuerung aktivieren, soll der Lehrermodus zukünftig mittels eines Passworts gesichert werden.

Sollte *ObjectWriter* wie oben beschrieben optimiert werden, kann das Programm in Zukunft noch besser den Informatik-Unterricht im Fach „Natur und Technik“ unterstützen und somit als wertvolles Werkzeug zum Verständnis der Klassen in Textdokumenten sowie der Enthält-Beziehung beitragen.

A. Anhang

A.1. Bezugsmöglichkeiten des Programms *ObjectWriter*

ObjectWriter liegt als lauffähige .jar-Datei vor. Voraussetzung für die Benutzung ist eine auf dem PC installierte Java Runtime Environment (JRE).

Das Programm kann von folgenden Quellen bezogen werden:

- Internet: *www.omasreiter.org*
- beiliegende CD

Hinweis: Auf der beiliegenden CD befinden sich neben dem Programm *ObjectWriter* – inklusive aller Quelltexte – auch der Quelltext dieses Dokuments sowie alle weiteren im Rahmen dieser Arbeit erstellten Dokumente.

A.2. Arbeitsauftrag – Die Attribute der Klasse ZEICHEN

Öffne in *ObjectWriter* (über das Menü „Datei“ → „Text öffnen“) den Text „Die Zwei Freunde und der Bär“. Formatiere danach den Text wie unten dargestellt, indem Du die folgenden Aufgaben löst:

Mit Hilfe der Punktnotation (Notiere jeweils *einen* Methodenaufruf):

1. Schreibe das Wort „fetter“ (Zeile 2) fett.
2. Setze den Schriftgrad des Wortes „großer“ (Zeile 5) auf 20 pt.
3. Schreibe das Wort „Bär“ (Zeile 5) orange.
4. Streiche das falsch geschriebene Wort „tod“ (Zeile 10) durch.

Mit Hilfe der Buttons:

5. Setze die Schriftart des Wortes „Leid“ (Zeile 3) auf Courier New.
6. Hinterlege das Wort „Freud“ (Zeile 3) purpur.
7. Stelle das Wort „Baum“ (Zeile 9) hoch.
8. Stelle das Wort „platt“ (Zeile 10) tief.
9. Unterstreiche das Wort „Warnung“ (Zeile 17).

1	Die zwei Freunde und der Bär
2	Zwei Freunde, ein schlanker und ein fetter Junge, gelobten sich gegensei-
3	tig, sich in allen Fällen treu beizustehen und Freud und Leid miteinander
4	zu teilen. So traten sie ihre Wanderschaft an.
5	Unvermutet kam ihnen auf einem engen Waldwege ein großer Bär
6	entgegen. Vereint hätten sie ihn vielleicht bezwungen. Da aber dem schlan-
7	ken Jungen sein Leben zu lieb war, verließ er, ebenso bald vergessend, was
8	er kurz vorher versprochen hatte, seinen Freund und kletterte auf einen
9	Baum. Als sich der andere nun verlassen sah, hatte er kaum noch Zeit,
10	sich <small>platt</small> auf den Boden zu werfen und sich tot tot zu stellen,
11	weil er gehört hatte, dass der Bär keine Toten verzehre.
12	Der Bär kam nun herbei, beleckte dem Daliegenden die Ohren, warf ihn
13	mit der Schnauze einige Male herum und trabte dann davon, weil er ihn
14	für tot hielt.
15	Sobald die Gefahr vorüber war, stieg jener vom Baume herab und fragte
16	seinen Gefährten voll Neugierde, was ihm der Bär zugeflüstert habe?
17	„Eine vortreffliche <u>Warnung</u> “, antwortete dieser, „nur schade, dass ich sie
18	nicht früher gewusst habe. Er sagte: Man solle sich nicht mit Menschen
19	einlassen, die ihre Freunde in der Not verlassen.“

A.3. Arbeitsauftrag – Die Attribute der Klasse ABSATZ

Öffne in *ObjectWriter* (über „Datei“ → „Text öffnen“) den Text „Absätze“.
Formatiere danach den Text wie unten dargestellt mit Hilfe der Punktnotation.

Absatz	Text
a0	Absätze
a1	Dieser Absatz ist linksbündig ausgerichtet. Die Zeilen sind so angeordnet, dass die Zeilenanfänge links senkrecht untereinander stehen, und die Zeilenenden frei auslaufen, d.h. flattern.
a2	Dieser Absatz ist rechtsbündig ausgerichtet. Die Zeilen sind so angeordnet, dass alle Zeilenenden rechts senkrecht untereinander stehen, während die Zeilenanfänge frei flattern.
a3	Dieser Absatz ist zentriert. Die Zeilen sind mittig zwischen den Rändern ausgerichtet. Die einzelnen Zeilen sind symmetrisch zur Mittelachse angeordnet.
a4	Dieser Absatz ist im Blocksatz geschrieben. Die Zeilen sind so angeordnet, dass alle Zeilenanfänge bzw. -enden (bis auf die letzte Zeile) links bzw. rechts senkrecht untereinander stehen. Dabei werden die Leerzeichen entsprechend verbreitert.
a5	Dieser Absatz hat einen Zeilenabstand von 2. Das bedeutet, dass eine Zeile so viel Platz bekommt, wie normalerweise zwei Zeilen.
a6	Dieser Absatz ist links um 24 pt eingezogen. Das bedeutet, dass der linke Rand um 24 pt vergrößert wurde. Alle Zeilen sind links eingerückt.
a7	Dieser Absatz ist rechts um 30 pt eingezogen. Das bedeutet, dass der rechte Rand um 30 pt vergrößert wurde. Alle Zeilen sind rechts eingerückt.
a8	Bei diesem Absatz ist die erste Zeile links um 54 pt eingezogen. Das bedeutet, dass der Rand der ersten Zeile um 54 pt vergrößert wurde. Die erste Zeile beginnt also weiter rechts als die restlichen Zeilen.
a9	Dieser Absatz ist links um 48 pt eingezogen, die erste Zeile ist links um -48 pt eingezogen. Das bedeutet, dass alle Zeilen bis auf die erste Zeile links eingerückt sind. Die erste Zeile beginnt also weiter links als die restlichen Zeilen. Dies nennt man einen hängenden Einzug.

Quelle:

Staatsinstitut für Schulqualität und Bildungsforschung, München:

Handreichungsentwurf: Informatik im Fach „Natur und Technik“.

URL: <http://www.isb.bayern.de/isb/>

A.4. Fragebogen

Der folgende Fragebogen wurde zur Evaluation des Programms *ObjectWriter* verwendet. Von den 50 Schülern der beiden 6. Klassen nahmen 45 Schüler an der Befragung teil.

Umfrage zum Programm ObjectWriter

1. Ich habe schnell verstanden, wie das Programm ObjectWriter funktioniert.

stimmt stimmt größtenteils stimmt größtenteils nicht stimmt nicht

2. Mit Hilfe des Programms konnte ich Änderungen an den Texten schnell vornehmen.

stimmt stimmt größtenteils stimmt größtenteils nicht stimmt nicht

3. Die Arbeit mit ObjectWriter hat Spaß gemacht.

stimmt stimmt größtenteils stimmt größtenteils nicht stimmt nicht

4. Die Arbeit mit ObjectWriter hat mir geholfen, zu verstehen, wie Methodenaufrufe in Textdokumenten funktionieren.

stimmt stimmt größtenteils stimmt größtenteils nicht stimmt nicht

5. Das hat mir an ObjectWriter gut gefallen:

6. Das hat mir an ObjectWriter nicht gefallen:

7. Das könnte man an ObjectWriter besser machen:

Abbildung 4.: Fragebogen zur Evaluation des Programms *ObjectWriter*.

Literaturverzeichnis

- [1] Aesop: *Die zwei Freunde und der Bär*, in: „Ausgewählte Fabeln von Aesop“
URL: <http://www.udoklinger.de/Deutsch/Fabeln/Aesop.htm>
- [2] Brichzin, P., Freiberger, U., Reinold, K.: *Ikarus. Natur und Technik. Schwerpunkt: Informatik 6/7 – Schülerbuch.*, 1. Auflage. Oldenbourg Schulbuchverlag, München, 2004.
- [3] Akademie für Lehrerfortbildung und Personalführung Dillingen (Hrsg.): *Psychologie für das Studienseminar*, 5. Auflage. Dillingen, 2009.
- [4] Flanagan, D.: *Java in a Nutshell*, 3. Auflage. O'Reilly, Köln, 2000.
- [5] Frey, E., Hubwieser, P., Winhard, F.: *Informatik 1: Objekte, Strukturen, Algorithmen. – Schülerbuch*, 1. Auflage. Klett, München, 2004.
- [6] Gagné, R. M.: *Die Bedingungen des menschlichen Lernens*, 2. Auflage. Schroedel, Hannover, 1970.
- [7] Jandl, E.: *fuenfter sein*, in: Jandl, E: Poetische Werke in 10 Bänden. Luchterhand, München, 1997.
URL: <http://www.kulturundsprache.at/index.php?id=160> [Stand: August 2011]
- [8] *Java*.
URL: www.java.com/de
- [9] *Microsoft Word*, Textverarbeitungsprogramm
URL: <http://office.microsoft.com/de-de/word/> [Stand: August 2011]
- [10] Oerter, R., Montada, L. (Hrsg.): *Entwicklungspsychologie*, 5. Auflage. BeltzPVU, München, 2002.
- [11] *OpenOffice.org Writer*, Textverarbeitungsprogramm
URL: <http://de.openoffice.org/product/writer.html> [Stand: August 2011]

-
- [12] Pabst, M.: *EOS: Einfache objektorientierte Sprache mit Entwicklungsumgebung*.
URL: <http://www.pabst-software.de/doku.php/programme:eos:start> [Stand: August 2011]
- [13] Pabst, M.: *ObjectDraw: ein gläsernes Zeichenprogramm*.
URL: <http://www.pabst-software.de/doku.php/programme:object-draw:start>
[Stand: August 2011]
- [14] *Project Gutenberg*, kostenlose eBooks.
URL: http://www.gutenberg.org/wiki/Main_Page [Stand: März 2011]
- [15] Staatsinstitut für Schulqualität und Bildungsforschung, München: *Handreichungs-entwurf: Informatik im Fach „Natur und Technik“*.
URL: <http://www.isb.bayern.de/isb/index.asp?MNav=6&QNav=12&TNav=1&INav=0&Pub=427> [Stand: August 2011]
- [16] Staatsinstitut für Schulqualität und Bildungsforschung, München: *Lehrplan für das Gymnasium in Bayern – Jahrgangsstufen-Lehrplan der Jahrgangsstufe 6 im Fach „Natur und Technik“*
URL: <http://www.isb-gym8-lehrplan.de/contentserv/3.1.neu/g8.de/index.php?StoryID=26433> [Stand: August 2011]
- [17] *Wikipedia - The Free Encyclopedia*.
URL: www.wikipedia.org [Stand: Juli 2011]