

FOM – Fachhochschule für Oekonomie & Management Essen

Angestrebter Abschluss
Diplom-Informatiker (FH)

7. Semester

Diplomarbeit zum Thema

Paket zur Erstellung
wissenschaftlicher Arbeiten
in \LaTeX

Betreuer: Professor Dr.-Ing. Torsten Finke

Autor: \LaTeX Kessler

Kaarst, den 28. Februar 2008

Veröffentlichungshinweis

Die vorliegende Diplomarbeit kann ohne Einschränkung der Öffentlichkeit zugänglich gemacht werden. Es wird sogar ausdrücklich darum gebeten. Jedoch ist jedwede kommerzielle Nutzung mit oder durch diese Abhandlung oder auch nur durch Teile dieser Abhandlung ausgeschlossen!

Inhalt

| | |
|--|-------------|
| Vereinbarungen | III |
| Abkürzungen | IV |
| Abbildungen | V |
| Quelltextbeispiele | VII |
| Tabellen | VIII |
| 1 Einleitung | 1 |
| 1.1 Status quo | 1 |
| 1.2 Abgrenzung | 3 |
| 1.3 Motivation und Herausforderung | 5 |
| 2 Dokumenterstellung | 7 |
| 2.1 Allgemeine Grundsätze | 7 |
| 2.2 Quellenzitate | 12 |
| 2.3 Objektbeschriftungen | 17 |
| 2.4 Dokumentstruktur | 19 |
| 2.4.1 Dokumentbeginn | 21 |
| 2.4.2 Textteil | 23 |
| 2.4.3 Dokumentende | 26 |
| 2.5 Seitennummerierung | 29 |
| 3 Implementierung | 32 |
| 3.1 Paketstruktur | 35 |
| 3.2 Paketelemente | 37 |
| 3.2.1 Dokumentenklasse | 37 |
| 3.2.2 Dokumentenbeginn | 52 |
| 3.2.3 Dokumentenende | 57 |

| | | |
|----------|--|------------|
| 3.2.4 | Absatzdefinition | 63 |
| 3.2.5 | Gleitobjektdefinition | 70 |
| 3.2.6 | Vermischtes | 80 |
| 3.2.7 | Indexkonfiguration | 83 |
| 3.3 | Benutzerdokumentation | 86 |
| 3.4 | Installationsmethode | 87 |
| 3.5 | Verträglichkeit | 89 |
| 3.6 | Kritikpunkte | 93 |
| 4 | Schlussbetrachtung | 97 |
| A | Quellcode des Pakets fomsdt | 101 |
| A.1 | Dokumentenklasse (Quelldatei <code>fomsdt.cls</code>) | 101 |
| A.2 | Dokumentenbeginn (Quelldatei <code>fomdss.sty</code>) | 101 |
| A.3 | Dokumentenende (Quelldatei <code>fomdes.sty</code>) | 101 |
| A.4 | Absatzdefinition (Quelldatei <code>fompar.sty</code>) | 101 |
| A.5 | Gleitobjektdefinition (Quelldatei <code>fomfod.sty</code>) | 101 |
| A.6 | Vermischtes (Quelldatei <code>fomisc.sty</code>) | 101 |
| A.7 | Indexkonfiguration (Quelldatei <code>fomidx.ist</code>) | 101 |
| A.8 | Installationsanweisung (Quelldatei <code>readme.txt</code>) | 101 |
| A.9 | Benutzerdokumentation (Quelldatei <code>fomman.tex</code>) | 101 |
| | Glossar | 102 |
| | Literatur | 105 |
| | Index | 109 |

Vereinbarungen

| | |
|--------------------------------------|---|
| <code>package</code> | Paketbezeichner verwenden zur Darstellung die Schriftart Sans Serif. |
| <code>file</code> | Datei- und Programmnamen verwenden zur Darstellung die Schriftart Typewriter . |
| <code>\macro</code> | Makros, respektive Kommandos, werden mit einem umgekehrten Schrägstrich (backslash) eingeleitet und verwenden zur Darstellung die Schriftart Typewriter . |
| <code>environment</code> | Umgebungen verwenden zur Darstellung die Schriftart Typewriter . |
| <code>option</code> | Optionen verwenden zur Darstellung die Schriftart Typewriter . |
| <code>defval</code> | Standardwerte verwenden zur Darstellung die Schriftart <i>Italics</i> . |
| <code><meta></code> | Metadaten, also Platzhalter für Informationen jeglicher Art, verwenden zur Darstellung die Schriftart <i>Italics</i> und sind zusätzlich noch in spitzen Klammern eingeschlossen. |
| <code>{<mustarg>}</code> | Pflichtparameter werden in geschweiften Klammern eingeschlossen und verwenden zur Formatierung der Daten die Darstellung <code><meta></code> . |
| <code>[<optarg>]</code> | Optionale Parameter werden in eckigen Klammern eingeschlossen und verwenden zur Formatierung der Daten die Darstellung <code><meta></code> . |

Abkürzungen

| | |
|---------------------|--|
| ASCII | <u>A</u> merican <u>S</u> tandard <u>C</u> ode for <u>I</u> nformation <u>I</u> nterchange |
| DIN | <u>D</u> eutsches <u>I</u> nstitut für <u>N</u> ormierung |
| EN | <u>E</u> uropäische <u>N</u> ormen |
| HTML | <u>H</u> ypertext <u>M</u> arkup <u>L</u> anguage |
| ISO | <u>I</u> nternationale <u>O</u> rganisation für <u>N</u> ormung |
| PDF | <u>P</u> ortable <u>D</u> ocument <u>F</u> ormat |
| WYSIWYG | <u>W</u> hat <u>y</u> ou <u>s</u> ee <u>i</u> s <u>w</u> hat <u>y</u> ou <u>g</u> et |

Abbildungen

| | | |
|----|--|----|
| 1 | Erstellprozess eines L ^A T _E X-Dokuments | 4 |
| 2 | Beispiel für das „Lesen in Sprüngen“ | 7 |
| 3 | Varianten gängiger Zitierweisen im Überblick | 13 |
| 4 | Zitatquellenangabe mit Vollbeleg | 14 |
| 5 | Zitatquellenangabe mit Kurzbeleg | 14 |
| 6 | Zitatquellenangabe in Harvard-Notation | 15 |
| 7 | Zitatquellenangabe in abgewandelter Harvard-Notation | 15 |
| 8 | Beispielanhang einer Abhandlung | 27 |
| 9 | Funktionsweise des neu definierten Makros <code>\maketitle</code> | 40 |
| 10 | Beispiele zur Variation des Titelblattes im Paket <code>fomstdt</code> | 43 |
| 11 | Ausgabe des Standardsperrvermerks ohne Signatur | 46 |
| 12 | Ausgabe des Sperrvermerks der FOM mit Signatur | 47 |
| 13 | Ausgabe der ehrenwörtlichen Erklärung mit Signatur | 50 |
| 14 | Ausgabe des Abkürzungsverzeichnisses | 56 |
| 15 | Abkürzungsverzeichnis mit geändertem Layout | 57 |
| 16 | Ergebnis der Makroverwendung <code>\SeeAlso</code> | 63 |
| 17 | Ergebnis der Verwendung der Umgebung <code>Citation</code> | 64 |
| 18 | Ergebnis der Makroverwendung <code>\Remark</code> | 66 |
| 19 | Ergebnis der Makroverwendung <code>\Item</code> | 67 |
| 20 | Ergebnis der globalen Anpassung für Makro <code>\Remark</code> | 68 |
| 21 | Mögliche Layouts des Makros <code>\Description</code> | 71 |
| 22 | Ergebnis der Verwendung des Makros <code>\SinglePicture</code> | 73 |
| 23 | Ergebnis der Verwendung der Umgebung <code>PlainFigure</code> | 75 |
| 24 | Ergebnis der Verwendung des Makros <code>\DoublePicture</code> | 77 |
| 25 | Ergebnis der Verwendung der Umgebung <code>Table</code> | 79 |
| 26 | Ergebnisse der Verwendung von <code>CodeListing</code> und <code>\CodeListingFile</code> | 80 |
| 27 | Ergebnis der Verwendung des Makros <code>\Raise</code> | 82 |

| | | |
|----|--|----|
| 28 | Anpassungen und Änderungen des Layouts des Indexes | 85 |
| 29 | Erstellprozess der Paketdateien zur Auslieferung | 88 |

Quelltextbeispiele

| | | |
|----|---|----|
| 1 | Makro mit zwei optionalen Parametern (Beispiel) | 33 |
| 2 | Umgebung mit zwei optionalen Parametern (Beispiel) | 34 |
| 3 | Konfiguration und Erzeugung der Titelseite | 42 |
| 4 | Erzeugung des Standardsperrvermerks ohne Signatur | 45 |
| 5 | Erzeugung des Sperrvermerks der FOM mit Signatur | 46 |
| 6 | Erzeugung der ehrenwörtlichen Erklärung mit Signatur | 49 |
| 7 | Erzeugung des Abkürzungsverzeichnisses | 55 |
| 8 | Layoutanpassungen im Abkürzungsverzeichnis | 56 |
| 9 | Verwendung des Makros <code>\SeeAlso</code> | 62 |
| 10 | Verwendung der Umgebung <code>Citation</code> | 63 |
| 11 | Verwendung des Makros <code>\Remark</code> | 66 |
| 12 | Verwendung des Makros <code>\Item</code> | 66 |
| 13 | Globale Anpassung des Einzugs für Makro <code>\Remark</code> | 68 |
| 14 | Beispiele zur Verwendung des Makros <code>\Description</code> | 70 |
| 15 | Beispiel zur Verwendung des Makros <code>\SinglePicture</code> | 73 |
| 16 | Beispiel zur Verwendung der Umgebung <code>PlainFigure</code> | 74 |
| 17 | Beispiel zur Verwendung des Makros <code>\DoublePicture</code> | 76 |
| 18 | Beispiel zur Verwendung der Umgebung <code>Table</code> | 78 |
| 19 | Beispiel zur Verwendung der Umgebung <code>CodeListing</code> | 80 |
| 20 | Beispiel zur Verwendung des Makros <code>\CodeListingFile</code> | 80 |
| 21 | Beispiele zur Verwendung des Makros <code>\Raise</code> | 81 |
| 22 | Konflikt mit der Option <code>nopagebreak</code> | 92 |
| 23 | Umgehung des Konflikts der Option <code>nopagebreak</code> | 92 |
| 24 | Beispiel zur Demonstration der Einfachheit der Einbeziehung mathematischer Formeln in ein L ^A T _E X-Dokument | 95 |

Tabellen

| | | |
|---|--|----|
| 1 | Seitennummerierung des Dokumentbeginns | 30 |
| 2 | Seitennummerierung des restlichen Dokuments | 31 |
| 3 | Mögliche Verwendungsformen des Makros <code>\TodayShort</code> | 84 |
| 4 | Übersicht der getesteten Pakete | 91 |
| 5 | Empfohlene Skalierungsuntergrenzen für Abbildungen | 91 |

1 Einleitung

Ziel dieser Diplomarbeit ist es, allen Studierenden ein Werkzeug bereitzustellen, mit dessen Hilfe die Anfertigung der im Rahmen des Studiums zu absolvierenden wissenschaftlichen Abhandlungen erleichtert werden kann. Nur sollten nicht ausschließlich Studenten von den Ergebnissen der hier vorliegenden Arbeit profitieren. Deshalb richtet sich diese Diplomarbeit an all jene, die ihren Lesern die „Früchte ihrer Arbeit“ in einer ansprechenden Form darbieten möchten.

Sicherlich sind zur Verbesserung des Verständnisses der im Verlauf dieser Abhandlung dargebotenen Inhalte einige Grundkenntnisse im Umgang mit dem Textsatzsystem \LaTeX ¹ empfehlenswert und auch hilfreich, doch nicht unbedingt erforderlich. Dennoch soll diese Arbeit nicht als „Einführung in \LaTeX “ missverstanden werden, da sie im Kern die speziellen Anforderungen der Erstellung wissenschaftlicher Abhandlungen fokussiert.

Nachdem zuvor noch einige grundlegende Gesichtspunkte behandelt werden, wird sich Kapitel 2 mit der grundsätzlichen Betrachtung wichtiger Elemente in wissenschaftlichen Abhandlungen befassen. Im daran anschließenden Kapitel 3 sollen dann die Ergebnisse der Umsetzung des *Pakets zur Erstellung wissenschaftlicher Arbeiten* näher untersucht werden. Hieran anschließend wird das Resümee in Kapitel 4 alle Betrachtungen dieser Abhandlung zusammenfassend reflektieren und die bis dahin vernachlässigten Bereiche kurz diskutieren.

Bevor jedoch der Einstieg in die Materie der Erstellung wissenschaftlicher Arbeiten erfolgen kann, noch ein Hinweis zu der in diesem Schriftstück verwendeten Anrede der Leserinnen und Leser. Zur Vermeidung eines unnötig komplizierten Satzbaus, bedingt durch eine geschlechtsspezifische Differenzierung, soll innerhalb dieser Abhandlung auf die explizite Unterscheidung in der Anrede der Leserinnen und Leser verzichtet werden. Stattdessen wird stellvertretend eine maskuline Anrede bevorzugt. Daher wird an dieser Stelle bei allen Leserinnen um Pardon für diese Entscheidung gebeten.

1.1 Status quo

Welcher Grund könnte vorliegen, warum sich jemand mit der Thematik der Erstellung wissenschaftlicher Abhandlungen auseinander setzen sollte? Diese Frage kann sicherlich von niemanden allumfassend und vor allem richtig beantwortet werden. Aber warum sollte sich dennoch jemand mit diesem Thema befassen? Weil nahezu jede lehrende Einrichtung den eigenen Vorstellungen

¹ sprich *Lah-Tech*

in Bezug auf Aufbau und Aussehen von wissenschaftlichen Abhandlungen den Vorzug einräumen, ohne dabei zu berücksichtigen, dass die Leser der publizierten Schriftsätze vor zum Teil erhebliche Herausforderungen gestellt werden!

Kurz gesagt existieren nahezu an jeder Hochschule und jeder Universität teilweise gegensätzlich lautende Leitfäden und Richtlinien zu Erstellung schriftlicher Arbeiten. Einige ausgewählte Beispiele sollen helfen, diese These zu untermauern.

Gegensätzlich lautende Richtlinien

Im Leitfaden der Christian-Albrechts-Universität zu Kiel steht zu lesen, dass Tabellen durch Überschriften zu kennzeichnen sind [27, Seite 9]. Hingegen fordert beispielsweise der Leitfaden der Fachhochschule für Oekonomie & Management, dass Tabellen mit Unterschriften zu betiteln sind [13, Seite 14].

Ein anderes Beispiel findet sich im Umgang mit so genannten Trivialabkürzungen. Beispielsweise gestattet der Leitfaden der Christian-Albrechts-Universität zu Kiel den Autoren die Verwendung gebräuchlicher Abkürzungen, wenn diese ebenfalls im Duden zu finden sind [27, Seite 9]. Indessen kann im Leitfaden der Fachhochschule für Oekonomie & Management nachgelesen werden, dass Abkürzungen „sparsam“ zu verwenden sind [13, Seite 13]. Dies wiederum bedeutet, dass speziell von der Verwendung gebräuchlicher Abkürzungen abzusehen ist.

Widersprüchlich lautende Richtlinien

Ein Beispiel für in sich widersprüchliche Richtlinien findet sich an der Technischen Universität zu Chemnitz. Darin heißt es, dass ein eineinhalbfacher Zeilenabstand gewählt werden soll. Jedoch wird mit der Begründung einer „Rohstoffersparnis“ direkt im anschließenden Satz empfohlen, dass der Blattrand nicht mehr als drei Zentimeter betragen sollte [22, Seite 7].

Bereits ein simpler Test belegt die Absurdität dieser Begründung. Denn nimmt man exemplarisch an, dass auf einem gängigen Blatt Papier in Größe DIN A4 bei einer gewählten Schriftgröße von zwölf Punkt und einem umgebenden Seitenrand von drei Zentimetern das kleine Wörtchen „test“ bequem 1.000-mal platziert werden kann, dann bewirkt die Änderung des Zeilenabstandes auf die empfohlenen eineinhalb Zeilen eine Verlängerung des Textes um die erwarteten 50 %. Verringert man nun, bei sonst gleichen Randbedingungen, den Blattrand auf zwei Zentimeter, dann beträgt der Überstand des Testtextes immerhin noch zirka 20 %!

An der Justus-Liebig-Universität zu Gießen kann ein Leitfaden nachgeschlagen werden, der teilweise unlogisch erscheinende Forderungen enthält. Darin wird beispielsweise verlangt, dass „Tabellen und Abbildungen (...) für sich verständlich sein [müssen]!“ [16, Seite 6], um diese Feststellung im nachfolgenden Satz zu revidieren! Denn hierin erfolgt der Hinweis, dass „Tabelle“ durch

„Tab.“ und „Abbildung“ durch „Abb.“ zu ersetzen sei [16, Seite 6]. Jedoch existiert innerhalb der gesamten Richtlinie kein Hinweis darauf, dass beide Abkürzungen auch in das Abkürzungsverzeichnis aufzunehmen sind. Denn es sollte nicht davon ausgegangen werden, dass einem jeden Leser die Bedeutung der Abkürzungen „Abb.“ und „Tab.“ geläufig ist.

Im weiteren Verlauf des Leitfadens der Gießener Universität steht dann auch zu lesen, dass alle Tabellen und Abbildungen über eine Quellenangabe verfügen sollen, auch wenn diese durch den Autor angefertigt wurden [16, Seite 6]. Diese Information ist redundant und somit überflüssig. Denn der Umstand, dass ein Autor eine Abbildung oder auch eine Tabelle selbst erstellt hat, spiegelt sich bereits im Fehlen einer entsprechenden Quellenangabe wider.

Nachdem die zuvor gezeigten Beispiele die Existenz absurder und widersprüchlicher Regeln offenbart haben, gilt es festzuhalten, dass diverse der vorhandenen Leitfäden und Richtlinien zur Erstellung wissenschaftlicher Abhandlungen in vielerlei Hinsicht von den Grundregeln einer guten Lesbarkeit abweichen.

Aus diesem Grund werden im weiteren Verlauf der hier vorliegenden Abhandlung die wichtigsten dieser Grundregeln erarbeitet. Der Versuch, vor allem eine sinnvolle Untermenge geltender Regeln zu bestimmen, soll ebenfalls Gegenstand der nachfolgenden Ausführungen sein.

Diese Arbeit hegt auf keinen Fall den Anspruch, Leitfäden oder Richtlinien zur Erstellung wissenschaftlicher Abhandlungen zu ersetzen. Vielmehr sollen geltende Regularien hinterfragt und im Hinblick auf ihren Beitrag zur Verbesserung der Lesbarkeit geprüft werden.

1.2 Abgrenzung

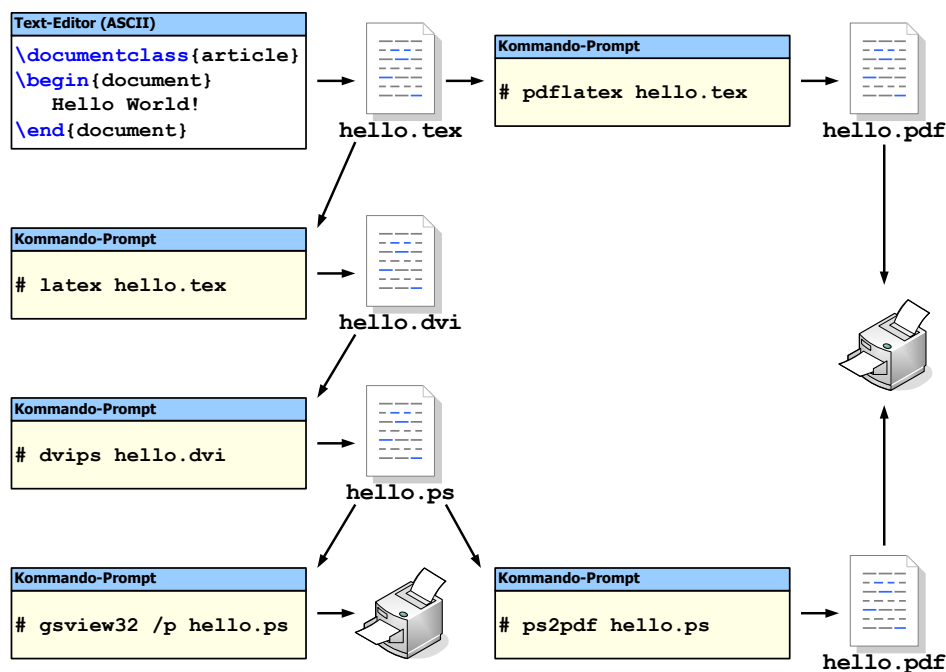
Das im Rahmen dieser Diplomarbeit implementierte *Paket zur Erstellung wissenschaftlicher Arbeiten* basiert auf dem Textsatzsystem \LaTeX . Da der Hintergrund dieses Systems der breiten Masse nicht allgemein geläufig ist, erscheint eine Abgrenzung zu anderen Systemen als sinnvoll. Eine entsprechende Gegenüberstellung soll Inhalt dieses Abschnittes sein.

Führt man sich die derzeitige Marktsituation bezüglich der verfügbaren Computerprogramme zur Erstellung textorientierter Dokumente vor Augen, kann man feststellen, dass zur Erstellung der anfallenden Texte in weiten Teilen der automatisierten Industrie und in fast allen computergestützten Bereichen des Dienstleistungssektors so genannte Textverarbeitungsprogramme zum Einsatz kommen. Diese auch als *Word Processor* bezeichneten Programme sind zumeist Bestandteil eines entsprechenden Office-Pakets.

Wie bereits angedeutet, verfügen derartige Textverarbeiter über einen hohen Verbreitungsgrad, da sie allesamt dem Leitsatz *What you see is what you get* (WYSIWYG) folgen. Somit zeichnen sich Textverarbeitungen dadurch aus, dass sie von deren Anwendern in der Regel leicht und intuitiv bedient werden können. Dies begründet auch, warum sich Textverarbeitungsprogramme einer großen Beliebtheit erfreuen, obgleich das auf Papier gedruckte Ergebnis weit von den im Buchdruck gelten Qualitätsanforderungen abweicht!

Hingegen zeichnen sich Textsatzsystems, wie im hier diskutierten Fall von \LaTeX , dadurch aus, dass der druckfertig ausgegebene Text nach den Erfahrungen und Regularien des Druckerhandwerks formatiert wird. Dieser Vorgang wird auch als „Setzen des Textes“ oder kurz Textsatz bezeichnet.

Also verfolgt ein Satzsystem wie \LaTeX einen zu Textprogrammen völlig gegenteiligen Ansatz. Denn hierbei steht das erzeugte Ergebnis und nicht die einfache und intuitive Bedienbarkeit im Vordergrund! Das hat auch zur Folge, dass sich der gesamte Prozess der Texterstellung vollständig von der eines Textprogramms unterscheidet, wie Abbildung 1 verdeutlicht.



Erstellung des Rohtextes, oder besser des Manuskripts, inklusive zusätzlicher Anweisungen zur Formatierung verwendet wird.

Danach werden ein oder mehrere Programme zur Konvertierung des Rohtextes in das gewünschte Ausgabeformat angewendet, um das generierte Ergebnis anschließend auf einem Drucker auszugeben oder zum Download im Internet bereitzustellen.

Dieser durchaus als komplex zu bezeichnende Vorgang bildet im Grunde genau den Prozess ab, der während des Verlegens eines Buches stattfindet. Denn ein Autor schreibt das Manuskript, dessen Inhalt durch einen Redakteur geprüft wird. Danach überträgt ein Schriftsetzer den Inhalt des Manuskripts auf die einzelnen Buchseiten, die im Anschluss durch einen Drucker auf Papier gebracht werden.

Wie sich jeder Leser sicherlich leicht vorstellen kann, benötigt die Einarbeitung in \LaTeX einen erheblichen zeitlichen Aufwand! Nicht nur, weil viele verschiedene Werkzeuge beherrscht werden müssen, sondern vielmehr aufgrund der Tatsache, dass \LaTeX nahezu unendliche Möglichkeiten der Kombination der Kommandos zur Textformatierung bietet.

Dies alles wirft unweigerlich die Frage auf, warum sich überhaupt jemand die Mühe der Einarbeitung in \LaTeX machen sollte? Zur Beantwortung dieser Frage reicht eine Begründung vollkommen aus. Das ausgegebene Dokument ist von einer derart hohen Qualität, dass es seinesgleichen sucht! Gerade aus diesem Grund findet man häufig auch Publikationen aus wissenschaftlichen Bereichen, die in \LaTeX gesetzt wurden.

1.3 Motivation und Herausforderung

Erfahrungsgemäß ist der zeitliche Aufwand zur Einarbeitung in \LaTeX im Vergleich zu gängigen Textverarbeitungen überdurchschnittlich hoch. Hinzu kommt auch, dass die Verwendung von \LaTeX so manchen Student vor zum Teil beachtliche Schwierigkeiten stellt, wenn er versucht, die geltenden Richtlinien seiner Hochschule einzuhalten. Denn wie bereits erwähnt, handelt es sich bei \LaTeX nicht um ein intuitiv und demzufolge leicht zu bedienendes System. Vielmehr ist das Entdecken und Erlernen neuer Kommandos zu Lösung spezifischer Formatierungsprobleme ein ständiger Bestandteil des täglichen Umgangs mit diesem System. Dennoch mehrt sich die Zahl derer, die ihren Schriftsätzen zumindest rein äußerlich eine hohe Qualität verleihen wollen.

Auch fehlt während des Studiums oft die Zeit. Dies gilt besonders für Studiengänge, die parallel zur eigentlichen Berufstätigkeit absolviert werden, zusätzlichen Aufwand zu betreiben, um sich das notwendige Wissen anzueignen, das

zur Erstellung wissenschaftlicher Abhandlungen unter Verwendung von L^AT_EX unumgänglich wird. Dies bezeugen verschiedene, jedoch wissenschaftlich nicht bewertbare Gespräche mit anderen Studenten. Doch war der Gehalt dieser Gespräche zumeist dadurch geprägt, dass sich die Studierenden davor scheuen, ihre Seminar- oder Abschlussarbeiten mit L^AT_EX zu setzen. Außerdem offenbarten diese Gespräche, dass die Verfügbarkeit einer entsprechenden „Vorlage“ das Schreiben der anstehenden Arbeiten mit L^AT_EX sehr erleichtern würde.

Vor diesem Hintergrund entstand der sicherlich nachvollziehbare Wunsch, die in der einjährigen Einarbeitungszeit gesammelten Erfahrungen mit anderen Kommilitonen teilen zu wollen. Dies ist gewiss Grund genug, um sich im Rahmen einer Diplomarbeit mit der Implementierung eines L^AT_EX-Paketes zur Erstellung wissenschaftlicher Abhandlungen auseinanderzusetzen.

Nachdem nun das Thema der Abschlussarbeit feststand, bestand die grundsätzliche Herausforderung darin, die über zwei Semester hinweg erarbeitete Basiskonfiguration in Form einer entsprechenden L^AT_EX-Präambel in eine für einen Endanwender bequeme Form zu überführen, welche zudem auch noch leicht zu bedienen sein sollte.

Gleichwohl zog diese Nebenbedingung einen tieferen Einstieg in das L^AT_EX-Kernsystem nach sich. Denn zur komfortablen Benutzung musste beispielsweise nicht nur die Frage nach einer sinnvollen Aufteilung der bereits vorhandenen Makros geklärt werden. Nein, auch die Definition zusätzlicher Kommandos, die Art der Verteilung des Implementierungsergebnisses sowie die Bereitstellung einer entsprechenden Endanwenderdokumentation waren gleichfalls ungelöst. Somit bestand in der Umsetzung dieser Punkte die eigentlichen Herausforderung der Implementierung des *Pakets zur Erstellung wissenschaftlicher Arbeiten*.

2 Dokumenterstellung

Dieser Abschnitt befasst sich mit der Beantwortung der Frage nach dem Aussehen einer „gut geschriebenen“ wissenschaftlichen Arbeit. Nachdem allgemeine und dennoch wichtige Grundsätze diskutiert wurden, gilt es, einzelne Elemente schriftlicher Ausarbeitungen näher zu beleuchten.

2.1 Allgemeine Grundsätze

Verfasser wissenschaftlicher Arbeiten sollte immer der Unterstützung der Lesbarkeit die höchste Priorität einräumen! Denn nichts wird als unangenehmer empfunden, als einen ohnehin schwierigen Fachtext lesen zu müssen, in dem der Lesefluss ständig durch zusätzliche Hürden unterbrochen wird [20].

Daher gelten einige Grundsätze, die der Verbesserung der Lesbarkeit wissenschaftlicher Ausarbeitungen dienlich sind. Einige dieser Grundsätze, die als wesentlich und wichtig erachtet werden, sollen Gegenstand dieses Abschnittes sein.

Zeilenlänge

Die Zeilenlänge ist einer der wichtigsten Einflussfaktoren für die Lesbarkeit eines Textes. Darin sind sich alle Forscher einig. Jedoch herrscht Uneinigkeit darüber, wie viele Buchstaben pro Zeile nun sinnvoll und optimal sind. Dabei schwanken die Angaben zwischen 40 und 70 Buchstaben je Textzeile [14, 18, 20, 21, 24].

Die Augensprünge beim Lesen werden als Sakkaden bezeichnet. Der Leser springt damit von Fixationspunkt zu Fixationspunkt. An diesen Stellen ist der Text am schärfsten.

Quelle: *Grundkurs Typografie und Layout* [21, Seite 102]

Abbildung 2: Beispiel für das „Lesen in Sprüngen“

Neuere Studien zeigen, dass das menschliche Auge nicht einzelne Buchstaben erfasst, sondern während des Lesevorgangs vielmehr von „Punkt zu Punkt“ springt, wie Abbildung 2 illustriert. Bei jedem Stopp werden dann die nächsten

drei bis zwölf Buchstaben erfasst, um sie anschließend zu entschlüsseln und weiterzuverarbeiten [10, 19, 21].

Somit kann davon ausgegangen, dass sich der eigentliche Lesevorgang nicht buchstabenorientiert vollzieht, sondern wortorientiert abläuft. Eine Zeilenlänge von sieben bis zwölf Wörtern soll dabei die Lesbarkeit optimal unterstützen [18, 21].

Betrachtet man beide Varianten empirisch etwas genauer, kommt man zu dem Ergebnis, dass eine Zeilenlänge von acht bis zwölf Wörtern im Mittel etwa 60 bis 65 Buchstaben² ergeben. Daher sollte der Verfasser einer wissenschaftlichen Abhandlungen eher dem Grundsatz folgen:

Bewegt sich beim Lesen der Kopf, dann ist die Zeile zu lang!

Diese Faustregel wirkt sich nicht nur positiv auf die Ermüdung des Lesers aus, sondern verbessert auch das Auffinden des Anfangs der nächsten Textzeile erheblich [1, 9, 21, 25].

Zeilenabstand

Ein in einschlägigen Richtlinien und Hinweisen zur Erstellung wissenschaftlicher Arbeiten ebenfalls gern diskutierter Punkt ist der Abstand der einzelnen Textzeilen zueinander. Da wird schon mal ein eineinhalbfacher [13, 15] und sogar bis zu zweizeiliger [12, 20] Abstand angeraten.

Auf der anderen Seite, so hat es zumindest den Anschein, wird von einigen Autoren ein größerer Zeilenabstand bevorzugt, mit dem Effekt, dass hierdurch die Seitenzahl des Werkes künstlich erhöht wird, ohne dabei eine tatsächliche Verbesserung der Lesefreundlichkeit zu erreichen.

In beiden Fällen ist offenbar unbekannt, dass der Durchschuss, wie Schriftsetzer den Abstand der einzelnen Textzeilen zueinander bezeichnen, keine frei wählbare Größe ist, sondern vielmehr durch die Wahl der Schriftart und der Schriftgröße bestimmt wird [20].

Vom Standpunkt des Leseflusses aus betrachtet behindert ein zu geringer, aber auch ein zu groß gewählter Zeilenabstand das menschliche Auge, den Beginn der nächsten Textzeile zu finden [18, 25]. Daher wirkt sich ein unglücklich gewählter Zeilenabstand unmittelbar auf den Lesefluss und somit auf die Lesbarkeit des gesamten Textes aus. Aus diesem Grund sollten Autoren wissenschaftlicher Arbeiten eine Anpassung des durch die Schriftart bestimmten Zeilenabstandes äußerst vorsichtig handhaben.

Grundsätzlich gilt der Zusammenhang, dass eine gute Lesbarkeit durch ein Zusammenspiel der Faktoren Schriftart, Schriftgröße, Wortabstand, Zeilenlänge

² Die Anzahl der Buchstaben pro Textzeile ist von sehr vielen Faktoren abhängig und kann daher nur als ungefähre Richtgröße angesehen werden.

und Zeilenabstand erreicht wird. Dadurch bedingt kann leider keine allgemeingültige Aussage über einen optimalen Zeilenabstand getroffen werden [14].

Schriftbild

Ein weiterer wichtiger Faktor zur Unterstützung der Lesbarkeit ist die Wahl des Schriftbildes. Hierzu zählen im engeren Sinne die verwendete Schriftart, aber auch die Schriftgröße. Im weiteren Sinne kann der Zeilenabstand ebenfalls zum Schriftbild gezählt werden. Weitere Einflussgrößen des Schriftbildes existieren, sollen jedoch im Rahmen dieser Abhandlung vernachlässigt werden [24].

Für längere Abhandlungen, zu denen wissenschaftliche Arbeiten zweifelsohne gerechnet werden können, sollte einer so genannten Serifenschrift in jedem Fall der Vorrang gewährt werden [20]. Der Grund hierfür ist in der Tat im Vorhandensein kleiner Häkchen und Füßchen zu sehen, die bei den hierzu zählenden Schriftarten existieren. Durch die Aneinanderreihung der Serifen der einzelnen Buchstaben entsteht im Gesamtbild eine „unsichtbare Linie“, die das Auge während des Lesens führt. Diese Zeilenführung ist bei serifenlosen Schriftarten nicht gegeben, sodass das Auge aus der Zeile „rutschen“ kann.

Mancherorts scheint es üblich zu sein, besonders große Schriften zu verwenden. Über die Beweggründe kann leider nur spekuliert werden. Fest steht jedoch, dass die Verwendung einer Schriftgröße von zum Beispiel 14 Punkt den Lesefluss negativ beeinflusst [20]. Daher wird für wissenschaftliche Arbeiten ein Schriftgrad von zehn beziehungsweise zwölf Punkt empfohlen.

Hervorhebung

Häufig wird durch die Autoren diverser Abhandlungen vom Mittel der Text hervorhebung Gebrauch gemacht. Dabei wird dieses Mittel allzu oft missbräuchlich eingesetzt, was wiederum der Unterstützung der Lesbarkeit abträglich ist [20].

Somit kommt es zuweilen vor, dass von den technisch zu Verfügung stehenden Schriftarten zehn in ein und demselben Dokument, ja sogar auf einer einzigen Seite, untergebracht werden. Dies kann durchaus als schlechter Schreibstil bezeichnet werden, da als Grundsatz die Einhaltung eines einheitlichen Schriftbildes verfolgt werden sollte.

Üblicherweise weicht die Darstellung von Eigennamen vom Grundsatz des einheitlichen Schriftbildes ab. Hierbei werden in der Regel Personennamen in Kapitälchen und Dateinamen zumeist in einer der Schreibmaschine vergleichbaren Schriftart dargestellt. Weitere Ausnahmen sind ebenfalls möglich.

Mit der Verwendung von Kapitälchen zur Hervorhebung von Personennamen soll dem Leser einer Abhandlung die besondere Ehrerbietung gezeigt werden, die ein Verfasser einer namentlich erwähnten Person entgegenbringt.

Hingegen hat die namentliche Erwähnung von Dateien in schriftlichen Ausarbeitungen einen eher technischen Touch. Aus diesem Grund werden Dateien auch in einer sehr technisch wirkenden Schriftart dargestellt.

Auch ist des Öfteren zu beobachten, dass Autoren ihnen besonders wichtig erscheinende Wörter durch Fettdruck oder sogar durch einfache beziehungsweise durch doppelte Unterstreichung kenntlich machen. Diese Form der Textmarkierung kann beim geneigten Leser jedoch dazu führen, sich durch den Autor bevormundet zu fühlen. Daher sollten Autoren wissenschaftlicher Arbeiten immer ein Auge darauf haben, ob eine Texthervorhebung notwendig und vor allem sinnvoll ist. Denn schließlich sollte der Leser selbst darüber befinden können, welcher Teil des Textes ihm besonders wichtig erscheint.

Linien jeglicher Art wirken sich innerhalb eines Textes besonders störend auf den Lesefluss aus, da sie die durch die Serifenschrift bewirkte Führung des Auges ablenken oder sogar unterbrechen können. Aus diesem Grund sollte gänzlich auf die Verwendung von Linien, dazu zählen neben Wortunterstreichungen auch Rahmen um Abbildungen, verzichtet werden. Von diesem Grundsatz sollte nur in ganz wenigen Ausnahmen abgewichen werden.

Zum einen ist es häufig üblich, Zeilen und Spalten von Tabellen durch Linien hervorzuheben, da durch die Tabelle selbst der Lesefluss bereits unterbrochen wurde. Dennoch sollte bei der Ausgestaltung von Tabellen darauf geachtet werden, dass weniger sehr oft mehr ist.

Zum anderen ist es ebenso häufig üblich, die Kopfzeile durch eine Linie vom restlichen Text zu separieren. Bezogen auf die Unterstützung der Lesbarkeit ist dies auch nicht weiter problematisch, da der Lesefluss ohnehin durch das Ende der vorherigen Seite unterbrochen wurde. Auch wird hierdurch der Beginn des Textteils der nächsten Seite deutlich markiert.

Die zuvor diskutierten Sachverhalte legen die Schlussfolgerung nahe, dass mit der Hervorhebung von Textpassagen äußerst sparsam umgegangen werden sollte. Wenn dennoch die besondere Markierung einzelner Wörter oder Wortgruppen als notwendig erachtet wird, dann sollte diese Kenntlichmachung stets auf eine unaufdringliche Art und Weise, zum Beispiel durch die Verwendung von schräg gestelltem Text, erfolgen.

Gliederungstiefe

Die Tiefe der einzelnen Gliederungsebenen ist ebenfalls ein nicht zu unterschätzender Punkt bei der Gestaltung wissenschaftlicher Arbeiten. Hierbei gilt, dass eine Schachtelungstiefe von mehr als drei Ebenen sich negativ auf die Lesefreundlichkeit des gesamten Werkes auswirkt.

Von diesem Grundsatz wird nur in wenigen und sehr spezifischen Abhandlungen aus dem Bereich der Rechtswissenschaften und in streng naturwissen-

schaftlichen Abhandlungen abgewichen, da hier nicht selten eine Herleitungskette als Beweis erforderlich ist [20].

Sollte dennoch die Unterteilung eines Abschnittes in mehrere unabhängige Unterabschnitte erforderlich sein, dann empfiehlt sich die Verwendung sinnvoller Absatzüberschriften, die über keinerlei Nummerierung verfügen und auch nicht im Inhaltsverzeichnis eingetragen werden [20]. Auf die Verwendung einer anderen Schriftart und auch einer anderen Schriftgröße sollte bei der Gestaltung dieser Unterüberschriften verzichtet werden, da „fett“ zu deren Hervorhebung völlig ausreichend ist. Der Abstand zum nachfolgenden Text sollte mit ebensoviel Bedacht gewählt werden, da andererseits der Zusammenhang zum Absatz verloren gehen könnte.

Abkürzungen

Bedingt der Inhalt eines gewählten Themas die Verwendung von Abkürzungen, dann sollte in jedem Fall darauf geachtet werden, dass die entsprechenden Erläuterungen zur Abkürzung im Text selbst vorzunehmen sind. Zudem sollten diese Erläuterungen bei der ersten Verwendung der jeweiligen Abkürzungen erfolgen. Das ist nicht nur höflich, sondern erspart dem Leser auch unnötiges Umblättern in das Abkürzungsverzeichnis und somit die Unterbrechung seines Leseflusses.

Weiterhin sollte jede wissenschaftliche Arbeit über ein Verzeichnis verfügen, in dem alle im Text vorhandenen Abkürzungen samt ihrer Erklärung aufgelistet werden. Enthält hingegen der Text keinerlei Abkürzungen, dann ist auch ein Abkürzungsverzeichnis überflüssig.

Auf die Benutzung von Trivialabkürzungen, wie beispielsweise *grds.*, *bzw.* oder auch *z. B.* sowie deren Erläuterung im Abkürzungsverzeichnis sollte in Abhandlungen mit wissenschaftlichem Anspruch immer verzichtet werden. Dies hat nicht nur etwas mit der Unterbrechung des Leseflusses zu tun, sondern wird auch dem Leser gegenüber als sehr unhöflich erachtet.

Des Weiteren besteht ein gewisses Risiko, dass die Verwendung von Trivialabkürzungen das Abkürzungsverzeichnis [13] und somit die Abhandlung also solches trivialisieren. Zudem verfügen die Verfasser längerer Aufsätze über genügend Zeit, alle Trivialabkürzungen auch auszuschreiben.

Fußnoten und andere stilistische Hilfsmittel

Ein immer wieder gern und heiß diskutiertes Thema ist die Verwendung von Fußnoten. Dabei liegen die Meinungen über Sinn und Zweck dieses Werkzeuges zum Teil so weit auseinander, dass kaum noch klar wird, wofür sie eigentlich gedacht sind.

Fußnoten dienen im Kern der Bereitstellung zusätzlicher Informationen, die aus verschiedenen Gründen nicht im übrigen Fließtext untergebracht werden können. Dabei gilt jedoch, dass der gesamte Text auch ohne diese Zusatzinformationen lesbar und vor allem verständlich sein sollte.

Hieraus kann geschlussfolgert werden, dass der Informationsgehalt einer Fußnote nicht notwendigerweise zum Verständnis des Gesamtzusammenhangs der entsprechenden Abhandlung beiträgt und somit getrost überlesen werden kann.

Überdies neigt der Leser beim Auftreten einer Fußnote zur abrupten Unterbrechung des Leseflusses. Daher sollte in wissenschaftlichen Abhandlungen grundsätzlich auf eine Verwendung von Fußnoten verzichtet werden.

Ein allzu oft angewendetes verbales Stilmittel ist der Verweis auf andere Kapitel oder Textpassagen. Dabei muss unterschieden werden, ob ein Hinweis auf bereits zuvor gelesene Passagen erfolgt oder ob sich der Autor auf spätere Kapitel bezieht. Grundsätzlich sind Querverweise auf bereits gelesene Abschnitte möglich und auch durchaus gängige Praxis. Jedoch wird dringend von der Verwendung der Verweise auf spätere Abschnitte abgeraten. Generell sollte man auf Querverweise dieser Art verzichten, da diese den Leser zum Umblättern animieren.

In Büchern ist häufig zu beobachten, dass sich im Randbereich einer Seite zusätzliche Erläuterungen zum Text oder auch Begriffserklärungen befinden. Dies mag sicherlich für Buchautoren ein geeignetes Werkzeug darstellen, sollte aber vom Verfasser einer wissenschaftlichen Arbeit gemieden werden.

Verfasser längerer Abhandlungen sehen sich immer wieder mit dem Problem konfrontiert, dass Textzeilen allein und verloren entweder am Ende der vorherigen oder am Anfang der nächsten Seite auftreten. Diese als Schusterjungen und Hurenkinder bezeichneten Ausreißer trüben das gesamte Schriftbild und gelten allgemein als sehr unerwünscht. Jedoch kann diesen Erscheinungen in der Regel durch die Umformulierung des Absatzes entgegengesteuert werden.

2.2 Quellenzitate

Das Zitieren der Werke anderer Autoren ist ein ebenso wesentlicher wie häufig wiederkehrender Vorgang während des Verfassens wissenschaftlicher Abhandlungen. Aus diesem Grund werden, so hat es zumindest den Anschein, von jeder Institution eigene Notationen zur Angabe von Zitatquellen präferiert. Die Fülle vorhandener Informationen über die verschiedenen Zitierweisen belegt diese Tatsache.

Daher wird sich dieser Abschnitt nicht mit dem „Wie“ des Zitierens auseinandersetzen, sondern sich lediglich auf den Vergleich und die Bewertung aus-

gewählter und gängiger Varianten der Quellenangaben im laufenden Text beschäftigen. Des Weiteren werden alle Betrachtungen vor dem Hintergrund der Lesefreundlichkeit durchgeführt. Zuvor soll jedoch einmal festgestellt werden, welche Varianten von Quellverweisen überhaupt existieren.

Zitierweisen im Überblick

Grundsätzlich kann nur zwischen einer Quellenangabe in Form eines Vollbelegs und der eines Kurzbelegs unterschieden werden. Alle weiteren Variationen können auf eine dieser beiden Methoden zurückgeführt werden, wie Abbildung 3 verdeutlicht.

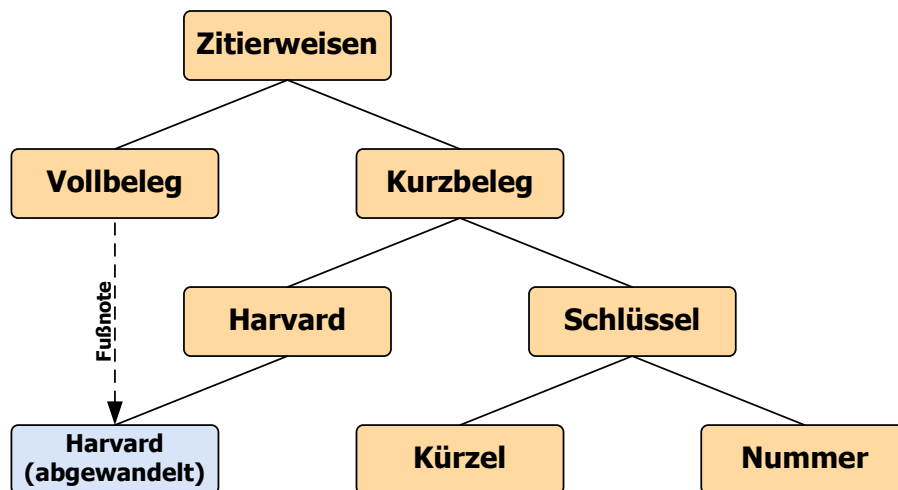


Abbildung 3: Varianten gängiger Zitierweisen im Überblick

Unter der Quellenangabe mit Vollbeleg, die oft auch als traditionelle Zitierweise bezeichnet wird [11], versteht man die vollständige Angabe aller bibliographischen Daten einer zitierten Quelle im laufenden Text. Da die Fülle von Informationen nur schwerlich im Fließtext unterzubringen ist, wird hierfür üblicherweise auf das Mittel der Fußnote zurückgegriffen. Abbildung 4 zeigt beispielhaft die Verwendung einer Quellangabe in Form eines Vollbelegs.

Hingegen wird bei der Variante des Zitierens mit Kurzbeleg auf eine vollständige Angabe aller bibliographischen Daten im laufenden Text verzichtet [28]. Hierbei werden die Quellenangaben zumeist auf den Namen des Autors sowie das Erscheinungsjahr des referenzierten Werkes beschränkt, wie Abbildung 5 beispielhaft darstellt. Diese Art des Zitierens setzt jedoch voraus, dass die im Kurzbeleg angegebene Quelle zumindest einmal im vorangegangenen Text mit Vollbeleg zitiert wurde. Ist dies der Fall, kann unter Umständen sogar auf ein Literaturverzeichnis verzichtet werden.

„Eine zur Lösung einer Aufgabe vollständige Anweisung an einen Rechner bezeichnet man als **Programm** (engl.: program); der Vorgang der Erstellung einer derartigen Anweisung heißt **Programmieren** (engl.: programming).“¹

¹ Hansen, Hans Robert; Neumann, Gustav: Wirtschaftsinformatik 1. Grundlagen und Anwendungen, 9. Auflage, Lucius & Lucius, Stuttgart et al. 2005, Seite 10.

Abbildung 4: Zitatquellenangabe mit Vollbeleg

„Eine Sammlung von gleichartigen Daten heißt **Datei** (engl.: file). Eine **Datenbank** (engl.: data base) bezeichnet eine Sammlung von mehreren Dateien, die gemeinsam verwaltet werden. Die Verwaltungsfunktionen dienen beispielsweise der Abfrage, Definition, Zugriffskontrolle und der Zugriffscoordination auf diesen Datenbestand.“²

² Vgl. Hansen/Neumann (2005), S. 11.

Abbildung 5: Zitatquellenangabe mit Kurzbeleg

In natur- und ingenieurwissenschaftlichen Schriften wird häufig eine weitere Form der Quellenangabe durch Kurzbeleg verwendet [28]. Hierbei wird der Informationsgehalt des Kurzbelegs bis auf einen eindeutigen Schlüssel reduziert, der zumeist durch eckige Klammern eingeschlossen wird. Dieser Schlüssel kann dann entweder aus einer einfachen Nummer oder aus der Zusammenfassung der Abkürzung des Autorennamens und dem Erscheinungsjahr bestehen. Der Leser muss nun mit Hilfe dieses Schlüssels die zum Zitat gehörenden bibliographischen Daten im Literaturverzeichnis der Abhandlung nachschlagen, da keine Referenzierung des Werks in der Fußnote erfolgt.

Eine Abwandlung des Quellenzitats durch Kurzbeleg ist die unter dem Begriff Harvard-Notation [4] bekannte Zitierweise. Hierbei erfolgt die Quellenangabe direkt hinter dem Zitat im laufenden Text. Ein Beispiel für dieser Art der Quellenangabe zeigt Abbildung 6.

Derartig reduzierte Quellenangaben sind jedoch im Hinblick auf eine Nachprüfbarkeit des Zitats unzureichend. Daher hat die vollständige Angabe der bibliographischen Daten des referenzierten Werkes in Form eines am Ende der Abhandlung befindlichen Literaturverzeichnisses zu erfolgen.

„Wirtschaftsinformatik ist die Wissenschaft von der Entwicklung und Implementierung betriebswirtschaftlicher computergestützter Informationssysteme.“ (SCHEER 1990, S. 1).

Abbildung 6: Zitatquellenangabe in Harvard-Notation

Eine spezielle Form der Quellenangabe ist die unter dem Begriff der abgewandelten Harvard-Notation geläufige Zitierweise [8]. Hierbei handelt es sich um eine in der Fußnote dargestellten Variante der Harvard-Notation, die sehr häufig im Bereich der Wirtschaftsinformatik und auch in betriebswirtschaftlichen Fachbereichen angewendet wird.

„Wirtschaftsinformatik ist die Wissenschaft von der Entwicklung und Implementierung betriebswirtschaftlicher computergestützter Informationssysteme.“¹

¹ SCHEER 1990, S. 1.

Abbildung 7: Zitatquellenangabe in abgewandelter Harvard-Notation

Abbildung 7 zeigt beispielhaft die Verwendung der abgewandelten Harvard-Notation. Auch bei dieser Zitiervariante erfolgt die Angabe aller bibliographischen Informationen mit Hilfe eines am Ende der Abhandlung befindlichen Literaturverzeichnisses.

Bewertung gängiger Zitierweisen

Nachdem nun alle gängigen Varianten der Quellenangaben für Zitate kurz vorgestellt wurden, soll vor dem Hintergrund der Auswirkungen auf die Lesbarkeit eine Bewertung der einzelnen Zitierweisen erfolgen.

Eine Quellenangabe durch Vollbeleg enthält alle bibliographischen Daten des Referenzwerkes in der Fußnote. Dieser Umstand bewirkt, dass Abhandlungen, die vielfach aus wenigen Quellen zitieren, unweigerlich redundante Informationen enthalten. Diese Redundanzen bergen nicht nur die Gefahr von inhaltlichen Fehlern der jeweiligen Quellenangabe, sondern sind auch unpraktisch, da der Leser immer wieder animiert wird, den Inhalt der Fußnote zu entschlüsseln. Dies unterbricht fortwährend den Lesefluss und führt dadurch zu einer Verschlechterung des inhaltlichen Verständnisses.

Auf Grund der Problematik der angesprochenen Informationsredundanz für Quellenangaben durch Vollbeleg wird in der Praxis häufig der Kurzbeleg für

weitere Nennungen eines zuvor vollständig zitierten Werkes verwendet. Da die Angabe des Kurzbelegs ebenfalls in der Fußnote erscheint, wird der Fluss des Lesens hierdurch genauso negativ beeinflusst, wie das beim Vollbeleg der Fall ist. Weiterhin wird ein Leser dazu genötigt, für detaillierte Informationen zur Quellenangabe das gesamte Schriftstück rückwärtig zu durchsuchen.

Der ständige Blick in die Fußnote und eine hieraus resultierende Leseunterbrechung spricht auch gegen die Verwendung der abgewandelten Harvard-Notation.

Ein weiterer Nachteil aller bis hierhin angesprochenen Zitierweisen besteht in der deutlichen Vermehrung der Fußnoten auf einer einzelnen Textseite. Das kann unter Umständen sogar dazu führen, dass sich auf einer Seite mehr Fußnoten finden lassen als inhaltlicher Text.

Positiv anzumerken gilt es jedoch, dass eine Nennung der Quelle in der Fußnote den Vorteil bietet, sofort und ohne umzublättern Rückschlüsse über den Hintergrund des Textes ziehen zu können, wenn den Lesern das zitierte Werk bereits bekannt ist.

Die Anwendung der Harvard-Notation löst zwar das Problem des Blicks in die Fußnote. Jedoch wird hierbei der Lesefluss durch einen verhältnismäßig hohen Informationsgehalt beeinträchtigt. Dadurch sieht sich der Leser mit der Herausforderung konfrontiert, die Fortführung des Inhalts hinter dem Zitat wieder zu finden. Denn wie bereits erwähnt, erfolgt das Lesen eines Textes anhand von Sprüngen zu frei gewählten Fixationspunkten.

Bleibt noch die überwiegend in naturwissenschaftlichen Abhandlungen vorzufindende Zitierweise zu betrachten. Aus Sicht der Verbesserung des Lesefflusses ist die Reduzierung des Informationsgehalts einer Quellenangabe auf einen kurzen Schlüssel die am besten geeignete Form der Zitatmarkierung!

Der Grund hierfür liegt in der Tatsache begründet, dass durch die Kürze des Schlüssels der Lesefluss am wenigsten gestört wird. Denn hierdurch werden vom Auge des Lesers keine zusätzlichen Sprünge benötigt, um die Fortführung des Inhalts wieder zu finden. Somit ist das Überlesen einer Quellenangabe leicht möglich. Der inhaltliche Zusammenhang geht ebenfalls nicht verloren.

Dennoch sollten sich Autoren wissenschaftlicher Arbeiten darüber im Klaren sein, dass zum Beispiel die Entschlüsselung einer Quellenangabe der Form [Han05] höhere Anforderungen an das Gehirn stellt, als die Entschlüsselung einer einfachen Zahl zwischen zwei eckigen Klammern.

Zusammenfassend kann nunmehr festgestellt werden, dass alle Zitierweisen, die Informationen in der Fußnote verwenden, sich negativ auf den Lesefluss auswirken. Ebenfalls negativ in Bezug auf eine Verbesserung der Lesbarkeit wirken sich die gekürzten Angaben direkt hinter einem Zitat aus.

Somit kann geschlussfolgert werden, dass von den hier diskutierten Zitiervarianten der Kurzbeleg direkt hinter einer Zitatstelle in Verbindung mit der Verwendung eines Schlüssels mit Abstand die geringste Beeinträchtigung des Leseflusses bietet. Daher sollten Autoren wissenschaftlicher Abhandlungen der in den natur- und ingenieurwissenschaftlichen Bereichen üblichen Zitierweise den Vorzug einräumen!

Möchte ein Leser dennoch mehr über das zitierte Werk in Erfahrung bringen, dann wurde der Lesefluss bereits unterbrochen und einem Blick in das Literaturverzeichnis steht nichts mehr im Wege.

2.3 Objektbeschriftungen

Wissenschaftliche Abhandlungen enthalten sehr häufig Objekte, die dem Leser spezielles Wissen in kompakter Form darbieten. Diese Darstellungen werden unter dem Begriff „Gleitobjekt“ zusammengefasst. Hierzu zählen unter anderem jegliche Form von Abbildungen, aber auch alle Arten von Tabellen. Codelistings können ebenfalls als eine Form von Gleitobjekten betrachtet werden.

Wie der Name bereits vermuten lässt, ist die Platzierung eines Gleitobjekts innerhalb des gedruckten Textes nicht exakt vorhersagbar. Daher sollten Autoren wissenschaftlicher Arbeiten immer auf eine direkte Bezugnahme, wie zum Beispiel „wie folgende Abbildung zeigt“ oder auch „stellt vorherige Abbildung dar“, verzichten. Stattdessen sollte eine indirekte Referenzierung über die Nummer des jeweiligen Gleitobjektes bevorzugt werden.

Da Gleitobjekte nicht unmittelbar zum Fließtext einer Abhandlung gehören, widerfährt ihnen auch eine Sonderbehandlung. Denn alle Gleitobjekte verfügen, mit nur ganz wenigen und spezifischen Ausnahmen, über eine Beschriftung. Dabei ist die Position der Beschriftung vielen Autoren wissenschaftlicher Abhandlungen nicht immer ganz geläufig.

Dieser Zusammenhang soll nun im Folgenden hergeleitet werden. Zu diesem Zweck wird sich jedoch auf Abbildungen, Tabellen und Codelistings, also die gängigen Gleitobjektformen einer wissenschaftlichen Arbeit in Bereichen der Informationstechnologie, beschränkt.

Beschriftung von Abbildungen

Abbildungen enthalten zumeist komplexe Zusammenhänge in graphisch aufbereiteter Form. Diese grundlegende Eigenschaft hat zur Folge, dass Abbildungen in den meisten Fällen auf einer einzelnen Dokumentseite untergebracht werden können. Dieser Umstand bewirkt, dass Abbildungen in der Praxis eine unterhalb befindliche Beschriftung erhalten.

Weiterhin stammen Abbildungen oft aus urheberrechtlich geschützten Quellen, sodass in diesem Fall einer Abbildung noch eine Quellenangabe hinzugefügt werden muss. Da dieser Teil der Beschriftung der eigentlichen Abbildung zuzurechnen ist, befindet sich der Verweis auf den Urheber auch direkt unterhalb der jeweiligen Abbildung, also genau zwischen Abbildung und Abbildungsbeschriftung.

Beschriftung von Tabellen und Codelistings

Tabellen, aber auch Quellcodelistings, haben die Eigenschaft, dass sie in der Regel über deutlich mehr textuelle Informationen verfügen, als das bei Abbildungen der Fall ist. Daher ist es nicht selten, dass dieser Typus Gleitobjekt sich über mehr als eine Seitenlänge erstreckt.

Nun könnte angenommen werden, dass Tabellen und auch Codelistings über eine Beschriftung unterhalb verfügen, wie dies bei Abbildungen der Fall ist. Dem ist aber nicht so! Denn grundsätzlich erhalten alle Gleitobjekte, mit Ausnahme der Abbildungen, ihre Beschriftung oberhalb³.

Das wirft unweigerlich die Frage nach dem „Warum“ auf. Der Grund hierfür ist ebenso leicht verständlich wie plausibel. Der Lesefluss wird durch die Tatsache der bloßen Existenz eines Gleitobjektes unterbrochen. Somit will der Leser unmittelbar wissen, wodurch sein Fluss unterbrochen wurde. Hierfür ist die Beschriftung oberhalb eines Gleitobjekts sehr hilfreich.

Dazu kommt auch, dass innerhalb des Textes auf Tabellen und alle anderen Gleitobjekte Bezug genommen wird. Um nun dem Leser ein leichtes Auffinden des jeweiligen Objektes zu ermöglichen, ist hierfür eine einleitende Beschriftung sehr hilfreich. Er muss also nicht erst umständlich das Objektende suchen. Somit wird das Gesamtverständnis des Fachtextes durch eine Objektüberschrift positiv beeinflusst.

Ein ebenfalls wichtiger Aspekt ist die Angabe des Ursprungs des Gleitobjektes. Dabei hat die Angabe der Quelle grundsätzlich unterhalb des Objekts zu erfolgen. Dies gilt im Speziellen für Tabellen, aber auch im Allgemeinen für alle anderen Gleitobjekte, die zur Untermauerung der eigenen Thesen aus anderen Arbeiten entnommen wurden.

Codelistings weichen in wissenschaftlichen Arbeiten oft von diesem Grundsatz ab, da sie zumeist aus der eigenen Feder stammen und demzufolge die Erstellung des Codes, zumindest teilweise, Inhalt der entsprechenden Abhandlung ist. Aber auch hier gilt: wurde der Code einer Fremdquelle entnommen, dann hat die Angabe des Ursprungs unterhalb des Listings zu erfolgen.

³ Die Feststellung, dass Tabellen über eine Überschrift verfügen, steht im Widerspruch zu den an der Fachhochschule für Oekonomie und Management (FOM) anerkannten, allgemeinen Richtlinien zur Erstellung wissenschaftlicher Arbeiten [13, Seite 14].

Regel für Objektbeschriftung

Zusammenfassend kann festgestellt werden, dass eine Beschriftung dem Gleitobjekt vorangestellt wird. Dies begründet sich durch die Tatsache, dass hierdurch zum einen dem Leser direkt mitgeteilt wird, warum sein Lesefluss unterbrochen wurde und zum anderen wird hierdurch ein schnelleres Auffinden des jeweiligen Objektes ermöglicht.

Einzige Ausnahme von diesem Grundsatz bilden Abbildungen, da diese auf einer Textseite Platz finden und somit deren Beschriftung auf einen Blick wahrgenommen werden kann.

Die Quellenangabe hat in jedem Fall unterhalb des entsprechenden Objektes zu erfolgen, da sie als eine Art Unterschrift angesehen werden kann!

2.4 Dokumentstruktur

Ein ebenfalls wesentlicher Gesichtspunkt bei der Erstellung wissenschaftlicher Abhandlungen ist deren Struktur. Jedoch sollte diese Struktur nicht willkürlich gewählt werden, sondern vielmehr allgemeinen und anerkannten Grundsätzen folgen, die überdies noch außerordentlich hilfreich für den Leser sind.

Im Allgemeinen lassen sich schriftliche Ausarbeitungen jeglicher Art in drei voneinander getrennte Bereiche unterteilen. Hierbei handelt es sich im Einzelnen um

- den Beginn des Dokuments,
- den Textteil und um
- das Ende des Dokuments.

Dabei sind Dokumentbeginn und Dokumentende vom Vorhandensein diverser Verzeichnisse geprägt. Die Diskussion über den Sinn und Zweck dieser Verzeichnisse soll Gegenstand der hieran anschließenden kurzen Einführung in das Thema der Dokumentstruktur sein.

Zweck eines Verzeichnisses

Wozu nützt dem Leser ein Verzeichnis? Verzeichnisse dienen in erster Linie der kompakten Darstellung spezieller Inhalte. Daher eignen sie sich sehr gut als zusätzliche Informationsquelle und bieten somit dem Leser einen nicht unerheblichen Mehrwert. Denn der Leser kann durch den Blick in ein Verzeichnis direkt zur Seite wechseln, auf dem sich das Objekt seines Interesses finden lässt.

Das setzt jedoch voraus, dass in den Verzeichnissen entsprechende Verweise auf die einzelnen Seiten vorhanden sind. Folgerichtig kann festgehalten werden, dass Verzeichnisse in der Regel über die Angabe einer Seitenzahl verfügen. Ansonsten bestünde für den Leser auch kein signifikanter Vorteil durch ein Verzeichnis.

Somit besteht die Aufgabe eines Verzeichnisses darin, dem Leser einer wissenschaftlichen Ausarbeitung das schnelle Auffinden einer bestimmten Information oder Sachverhaltes zu erleichtern beziehungsweise erst zu ermöglichen.

Platzierung eines Verzeichnisses

Die nächste Frage, die im Rahmen dieses Abschnittes betrachtet werden soll, ist die Frage, warum sich Verzeichnisse am Beginn oder am Ende einer Abhandlung befinden. Zur Beantwortung dieser Frage soll nun das Verhalten eines Leser etwas genauer betrachtet werden.

Ein Leser trifft auf ein unbekanntes Buch

Man stelle sich vor, man erhält ein unbekanntes Buch. Nachdem nun der Titel weiteres Interesse geweckt hat, möchte man mehr über den Inhalt in Erfahrung bringen. Dazu wird das betreffende Buch aufgeschlagen und man findet als Erstes das Inhaltsverzeichnis!

Ein Inhaltsverzeichnis ermöglicht dem Leser, neben der inhaltlichen Übersicht, auch etwas über die Struktur und die Aufteilung des Werkes in Erfahrung zu bringen. Dieser Umstand ist für den Leser mehr als hilfreich, da er auf einen Blick entscheiden kann, eventuell tiefer in das jeweilige Thema einzusteigen.

Dieser Zusammenhang belegt die in der Praxis vorzufindende Tatsache, dass Verzeichnisse, welche die Struktur einer Abhandlung widerspiegeln, sich auch am Anfang dieser Abhandlung befinden.

Das Inhaltsverzeichnis bildet zweifelsfrei die Struktur einer Ausarbeitung ab. Jedoch ist dies bei einem Abkürzungs-, einem Abbildungs- und auch einem Tabellenverzeichnis, die sich ebenfalls am Beginn eines Dokuments befinden sollen, nicht so eindeutig gegeben.

Vielmehr erweckt die Praxis den Anschein, dass diese Verzeichnisse nach „Gutdünken“ in den jeweiligen Abhandlungen verteilt werden. Da findet man schon mal Bücher, in denen die Abbildungen und Tabellen im Anhang gelistet werden. Ist das sinnvoll? Oder ist es doch besser, Abkürzungen, Abbildungen und Tabellen an den Anfang des literarischen Werkes zu stellen?

Diese Frage kann leider nicht hinreichend genau beantwortet werden! Somit muss sich ein Autor oft, zumindest in diesem Punkt, auf seinen eigenen guten Geschmack stützen, um die Entscheidung für eine Platzierung dieser Verzeichnisse treffen zu können.

Aber dies betrifft vorwiegend Buchautoren. Denn die Autoren wissenschaftlicher Arbeiten unterliegen nicht selten den Regularien ihres Fachgebiets. Und hier herrscht die einhellige Meinung, dass Abbildungen, Tabellen und auch Abkürzungen am Beginn der Ausarbeitung zu finden sind [4, 13, 15, 5].

Ein Leser trifft auf ein bekanntes Buch

Betrachtet man nun den Umstand, dass ein Leser zu einem ihm bekannten Buch greift, beispielsweise um etwas nachzuschlagen, dann kann es für ihn durchaus hilfreich sein, darin Verzeichnisse mit nützlichen Informationen vorzufinden. Als Vertreter eines solchen Verzeichnisses sei der Index hier namentlich erwähnt.

Denn unter der Voraussetzung, dass der geneigte Leser den Inhalt einer Abhandlung bereits kennt, ist das schnelle Auffinden einer dort enthaltenen, spezifischen Information außerordentlich zielführend.

Da derartige Verzeichnisse häufig sehr viele dieser nützlichen Informationen enthalten und demzufolge recht lang sein können, empfiehlt es sich, diese auch am Ende des Werkes unterzubringen.

Manch einer wird sich sicherlich fragen, warum diese Verzeichnisse nicht doch besser am Anfang platziert werden sollten. Die Antwort darauf ist recht einfach und einleuchtend zugleich. Ein Index als Beispiel enthält überhaupt keine Informationen über die Struktur der betreffenden Abhandlung. Er bietet somit dem Leser auch keinen zusätzlichen Nutzen, wenn er das Buch zum ersten Mal aufschlägt.

Zudem kommt auch, dass sich zum Beispiel ein Index nicht selten über viele Seiten erstreckt. Stünde der Index am Anfang, dann würde dies die Suche nach dem inhaltlichen Beginn der Abhandlung noch zusätzlich erschweren.

Alle zuvor diskutierten Zusammenhänge legen die Schlussfolgerung nahe, dass sich am Anfang einer wissenschaftlichen Ausarbeitung nur Verzeichnisse befinden, die den strukturellen Aufbau der Arbeit widerspiegeln. Hingegen befinden sich am Ende der Arbeit nur solche Verzeichnisse, die ein schnelles Auffinden von Informationen innerhalb des Dokuments ermöglichen.

2.4.1 Dokumentbeginn

Der einleitende Teil einer wissenschaftlichen Arbeit besteht in der Regel aus einem Vorspann und einer strukturellen Übersicht in Form diverser Verzeichnisse. Die genauere Betrachtung der einzelnen Komponenten des Dokumentenbeginns soll Gegenstand dieses Abschnittes sein.

Titelseite

Die Titelseite ist die erste Seite einer wissenschaftlichen Arbeit! Diese Trivialaussage zielt darauf ab, den geneigten Leser dafür zu sensibilisieren, dass der Vorspann einer wissenschaftlichen Arbeit nicht nur aus einem einzelnen Blatt mit Angaben zum Institut, Thema und Autor besteht, sondern auch weitere Textpassagen bereithalten kann.

Da wäre zunächst einmal der Sperrvermerk zu nennen. Er wird vor allem bei Diplomarbeiten verwendet, um festzulegen, dass der Inhalt der Abhandlung ausdrücklich nur einem befugten Personenkreis zugänglich sein darf. Also ist ein Sperrvermerk häufig in den wissenschaftlichen Arbeiten zu finden, die auf Basis firmenspezifischer und zumeist geheimer Daten beruhen [13]. Denn die Preisgabe derartiger Informationen könnte unter Umständen die Existenz des jeweiligen Unternehmens gefährden.

Sollte nun aufgrund des behandelten Themas ein Sperrvermerk für wissenschaftliche Abhandlungen erforderlich werden, dann empfiehlt es sich, diesen unbedingt auf der Seite direkt hinter dem Titelblatt zu platzieren!

Ein Vorwort und auch eine Danksagung sind ebenfalls potenzielle Kandidaten für den Vorspann einer wissenschaftlichen Abhandlung. Sie werden jedoch der Arbeit nur bei Bedarf beigelegt.

All diese zusätzlichen Seiten weisen die Gemeinsamkeit auf, dass sie typischerweise vor dem Inhaltsverzeichnis zu platzieren sind und somit von einer Seitennummerierung ausgenommen werden. Dies ist durch die Tatsache begründet, dass derartige Seiten nicht zum eigentlichen Inhalt der Abhandlung gezählt werden können!

Inhaltsverzeichnis

Das Inhaltsverzeichnis ist das erste Verzeichnis, das sich an den Vorspann einer wissenschaftlichen Arbeit anschließt. Es ist für die Abhandlung von immenser Wichtigkeit, da es neben der inhaltlichen Übersicht auch die Gesamtstruktur des Werkes widerspiegelt.

Weitere Verzeichnisse

Sollte die wissenschaftliche Abhandlung, wie in den meisten Fällen üblich, auch Tabellen, Abbildungen oder Abkürzungen enthalten, dann sind durch den Autor entsprechende Verzeichnisse bereitzustellen. Dabei sind diese Verzeichnisse direkt hinter dem Inhaltsverzeichnis zu positionieren.

Leider kann die Reihenfolge dieser Verzeichnisse nicht eindeutig festgelegt werden, da sie in den jeweils geltenden Regelungen zur Erstellung wissenschaftlicher Arbeiten variieren [13, 15, 5]. Dennoch erscheint eine Reihenfolge von

- Abkürzungsverzeichnis,
- Abbildungsverzeichnis und
- Tabellenverzeichnis

als sinnvolle Wahl, da diese der Häufigkeit ihrer Benutzung am nächsten kommt. Vermutlich werden die Leser einer wissenschaftlichen Abhandlung das Abkürzungsverzeichnis viel öfter frequentieren als beispielsweise das Abbildungs- oder das Tabellenverzeichnis. Daher bietet es sich förmlich an, das Abkürzungsverzeichnis direkt hinter dem Inhaltsverzeichnis zu platzieren.

Seitenvorschub bei Verzeichnissen

Grundsätzlich sollte sich jedes einzelne Verzeichnis auf einer separaten Seite befinden. Jedoch führt dies im Falle sehr kurzer Verzeichnisse zu einem weniger schönen Aussehen des Dokumentbeginns. In diesem Fall kann ein deutlich besseres Layout der Arbeit durch die Zusammenfassung der einzelnen, kurzen Verzeichnisse auf fortlaufenden Seiten erreicht werden.

Diese Ausnahmeregel gilt jedoch nicht für das Inhaltsverzeichnis! Denn diesem Verzeichnis wird eine gewisse Sonderrolle zuteil und es beginnt somit immer auf einer neuen Textseite. Alles Weitere muss dann ebenfalls auf einer neuen Seite beginnen.

2.4.2 Textteil

Wie sich leicht vorstellen lässt, ist das Treffen allgemeingültiger Aussagen über den inhaltlichen Abschnitt einer wissenschaftlichen Arbeit ein eher schwieriges Unterfangen. Daher kann und muss sich der Inhalt dieses Kapitels auf einige wenige grundsätzliche Fragen beschränken.

Textzwischenraum

Wie bereits angesprochen, ist eine isolierte Betrachtung des idealen Zeilenabstandes nur sehr schwer möglich, da eine gute Lesbarkeit von gedrucktem Text durch das Zusammenspiel der Faktoren Schriftart, Schriftgröße, Wortabstand, Zeilenlänge und Zeilenabstand bestimmt wird [21, 24].

Grundsätzlich kann davon ausgegangen werden, dass die meisten verfügbaren Schriftarten die Anforderungen für eine gute Lesbarkeit bezüglich ihrer horizontalen und vertikalen Abstände bereits erfüllen. Daher ist in der Regel auch ein einzeiliger Abstand für wissenschaftlichen Abhandlungen ausreichend. Sollte der Text durch die verwendete Schrift dennoch gedrungen wirken, dann ist zu dessen Auflockerung ein etwa 1,2-zeiliger Abstand empfehlenswert.

An dieser Stelle soll jedoch nicht verheimlicht werden, dass die zuvor gegebene Empfehlung im Grunde nur ihre Gültigkeit für auf Papier gedruckte Texte besitzt. Das liegt daran, dass ein zu wählender Zeilenabstand nicht ausschließlich von der Schrift abhängt, sondern auch durch das eingesetzte Anzeigemedium tangiert wird. Beispielsweise existieren Empfehlungen, dass Texte, die vorwiegend am Bildschirm gelesen werden, eine Verbesserung der Lesbarkeit durch einen Abstand von mindestens eineinhalb Zeilen erfahren [18, 25].

Dieser kurze Exkurs deutet eine inhärente Komplexität an, auf deren tief greifende Diskussion bezüglich der Lesbarkeit von Bildschirmtexten innerhalb dieser Abhandlung leider verzichtet werden muss. Für den interessierten Leser, der mehr über eine Beeinflussung des Leseflusses durch elektronische Medien in Erfahrung bringen möchten, sei auf die Existenz verschiedентlicher Studien zu diesem Thema hingewiesen.

Seitenvorschub nach Kapitelende

Jeder, der bereits einmal ein Buch in der Hand hielt, konnte feststellen, dass die Kapitel der Hauptebenen auf einer neuen Seite beginnen. Häufig ist auch zu beobachten, dass Hauptkapitel sogar nur auf den ungeraden Seiten ihren Anfang nehmen. Das hat einen guten Grund! Denn wurde ein Kapitel beendet, dann wurde auch ein gewisser Themenkomplex abgeschlossen. Der Leser hat nun die Gelegenheit, den Inhalt des Kapitels zu reflektieren. Der Lesefluss ist somit unterbrochen.

Also stellt sich die Frage, warum nicht auch in wissenschaftlichen Abhandlungen jedes Hauptkapitel auf einer neuen Seite beginnen sollte. Die Antwort ist simpel. Auch in wissenschaftlichen Arbeiten sollten die Kapitel der Hauptgliederungsebenen auf einer neuen Seite beginnen. Das sieht nicht nur besser aus, sondern erleichtert auch dem Leser die Suche nach einem bestimmten Kapitel.

Nebenbei sein noch angemerkt, dass der Seitenvorschub am Kapitelende den verschiedenen Textverarbeitungswerkzeugen die Formatierung der Kopfzeile stark erleichtert, da hierdurch Konflikte zwischen konkurrierenden Kapitelüberschriften vermieden werden können.

Allgemeiner Aufbau

Wie bereits einleitend erwähnt, kann über den Inhalt einer wissenschaftlichen Ausarbeitung keine allgemeingültige Aussage getroffen werden, da dieser ausschließlich vom bearbeiteten Thema abhängt. Dies ist für den grundlegenden Aufbau der Arbeit dennoch möglich. Denn im Kern kann fast jede Abhandlung mit wissenschaftlichem Hintergrund bis auf vier Bereiche abstrahiert werden. Bei diesen Bereichen handelt es im Einzelnen um

- die Einleitung,

- den Grundlagenteil,
- den Kern der Abhandlung und um
- die Schlussbetrachtung.

Die Einleitung der Abhandlung

Der Teil der Einleitung ist einer der wichtigsten Bereiche der gesamten Abhandlung, da hier sämtliche Randbedingungen vereinbart und festgelegt werden können. Dazu zählen unter anderem die Definition der Zielgruppe, die Abgrenzung des Themas und die Motivation des Autors.

Weiterhin besteht die Möglichkeit, eine Auswertung und auch eine Bewertung der verwendeten Literatur sowie deren Quellen vorzunehmen. Dies ist zum Beispiel dann sehr sinnvoll, wenn der Autor über ein bisher stark vernachlässigtes oder auch über ein brandneues Thema schreibt und demzufolge die verfügbaren Literaturquellen nur sehr spartanisch ausfallen.

Die Unterbringung historischer Hintergrundinformationen, wenn sie zur allgemeinen Aufhellung des Kernthemas hilfreich sind, können ebenfalls in der Einleitung der wissenschaftlichen Arbeit untergebracht werden.

Die Grundlagen einer Abhandlung

Im Teil der Grundlagen werden in der Regel vom Autor die Basisinformationen bereitgestellt, die zum Verständnis des im Abhandlungskern befindlichen Teils der Arbeit notwendig sind. Dabei sollte sich das vermittelte Basiswissen an den Bedürfnissen der Zielgruppe orientieren. Es ist sicherlich überflüssig, einer aus Informatikern bestehenden Zielgruppe den grundsätzlichen Aufbau eines Computers zu erläutern.

Der Kern einer Abhandlung

Der inhaltliche Abschnitt der wissenschaftlichen Arbeit besteht zumeist aus der Beschreibung einer zur Abhandlung gehörenden „Durchführung“. Dieser stark themenspezifische Teil kann beispielsweise die Implementierung einer Problemlösung oder auch die Analyse sowie die Auswertung einer Umfrage enthalten.

Je nach Art der Abhandlung kann dessen Kern aus einem oder mehreren Kapiteln bestehen. Dabei hängt die Anzahl der Kapitel zumeist von der bearbeiteten Aufgabe ab. Jedoch sollte in einer wissenschaftlichen Arbeit für diesen Komplex mindestens ein Kapitel spendiert werden.

Die Schlussbetrachtung

Eine Schlussbetrachtung, oder auch ein Fazit, sollte in einer wissenschaftlichen Abhandlung niemals fehlen, da sie die Reflexion und Bewertung der eigenen

Ergebnisse zulässt. Sie bietet somit auch Raum zur Selbstkritik an diesen Ergebnissen.

Des Weiteren stellt die Schlussbetrachtung eine Plattform zur Benennung ungelöster Probleme und unbeantworteter Fragen dar. Ein Ausblick in zukünftige Entwicklungen wird ebenfalls gern in diesem Abschnitt untergebracht.

2.4.3 Dokumentende

Der Abschluss einer wissenschaftlichen Arbeit ist ebenfalls durch die Existenz verschiedenlicher Verzeichnisse gekennzeichnet. Jedoch finden sich hier zumeist nur Verzeichnisse wieder, die für den Leser nützliche und hilfreiche Zusatzinformationen bereithalten. Die detaillierte Betrachtung der am Ende einer wissenschaftlichen Abhandlung vorzufindenden Verzeichnisse soll Gegenstand dieses Abschnittes sein.

Abgrenzung

Oft werden die Verzeichnisse des Dokumentenendes lapidar als Anhang bezeichnet. Doch trifft dies bei genauerer Betrachtung nur bedingt zu. Denn der Anhang im eigentlichen Sinne ist der Teil einer Publikation, der inhaltlich zur Abhandlung selbst gehört [2, 12, 7].

Hingegen hat zum Beispiel ein Literaturverzeichnis die Aufgabe, den Nachweis zu den durch den Autor aufgestellten Behauptungen zu erbringen und somit eine Nachprüfbarkeit der Arbeit zu ermöglichen. Also gehört ein Literaturverzeichnis nicht unmittelbar zum Abhandlungsinhalt.

Dieser Vergleich ermöglicht nun die grundsätzliche Unterscheidung, welches der in wissenschaftlichen Arbeiten gängigen Verzeichnisse in einen Anhang gehören und welche davon separiert werden.

Anhänge

Wie zuvor bereits festgestellt, dient der Anhang dazu, zusätzliche, zum Inhalt der Arbeit gehörende Elemente aufzunehmen, weil sie beispielsweise für eine Unterbringung im Kerntext zu umfangreich sind. Typische Vertreter derartiger Elemente sind Statistiken sowie lange, oft mehrseitige Tabellen, ganzseitige Abbildungen, Musterformulare, Fragebögen und auch Codelistings.

Die Sonderstellung des Anhangs innerhalb einer wissenschaftlichen Arbeit wird dadurch betont, dass die Abschnitte ebenfalls in die Kapitelnummerierung einbezogen werden. Nur wird dabei in der Regel die Nummerierung der einzelnen Ebenen durch einen Buchstaben eingeleitet [6, 5]. Somit könnte sich beispielsweise ein Anhang einer Abschlussarbeit, basierend auf der Auswertung statistischer Erhebungen, wie in Abbildung 8 dargestellt, ergeben.

A Basis und Ergebnisse der Umfrage**A.1 Verwendeter Fragebogen**

...

A.2 Erhobene statistische Daten

...

A.3 Diagramme der Auswertung**Abbildung 8:** Beispielanhang einer Abhandlung

Bei sehr umfangreichen Anhangsinhalten kann es unter Umständen sinnvoll sein, einem Anhang ein eigenes Verzeichnis voranzustellen [6]. Dies erleichtert dem Leser die Suche nach speziellen Inhalten des Anhangs.

Verzeichnisse

Die Verzeichnisse des Dokumentenendes gehören nicht unmittelbar zum Inhalt der wissenschaftlichen Abhandlung, sondern dienen dem Leser vielmehr als zusätzliche Informationsquelle.

Häufige Vertreter dieser Verzeichnisse sind das Literaturverzeichnis, das Glossar und auch der Index. Alle diese Verzeichnisse sind grundsätzlich als optional zu betrachten. Doch wenn sie in einer Abhandlung verwendet werden, dann weisen sie alle die Gemeinsamkeit auf, dass sie über keinerlei Nummerierung ihrer Kapitelüberschriften verfügen.

Literaturverzeichnis

Unter der Voraussetzung, dass entweder keine Fremdliteratur verwendet wurde oder alle Literaturverweise durch Vollbeleg bereits im laufenden Text abgedeckt wurden, kann zum Beispiel auf ein Literaturverzeichnis verzichtet werden. Jedoch ist diese Art der Betrachtung von eher akademischer Natur, da sie für wissenschaftliche Arbeiten in der Regel ausgeschlossen werden kann. Somit ist das Literaturverzeichnis als Pflichtbestandteil einer Abhandlung mit wissenschaftlichem Hintergrund zu sehen.

Glossar und Index

Anders verhält es sich hingegen bei einem Glossar, das auch als Sachwortverzeichnis geläufig ist, und einem Index, der häufig auch als Stichwortverzeichnis bezeichnet wird. Denn die Erstellung beider Verzeichnisse bedingt zusätzliche Anstrengungen des Autors, die zum Teil erheblichen Aufwand bedeuten können. Diese Betrachtung ist jedoch im Zeitalter der elektronischen Datenverarbeitung von eher untergeordneter Bedeutung, da eine Verzeichniserstellung in der Regel automatisiert erfolgen kann.

Somit kann geschlussfolgert werden, dass eine „gut geschriebene“ wissenschaftliche Abhandlung zumindest über einen Index zum Nachschlagen bestimmter und wichtiger Begriffe verfügen sollte. Auf die Bereitstellung eines Glossars zur Erläuterungen spezieller Fachbegriffe sollte in wissenschaftlichen Arbeiten ebenfalls nicht verzichtet werden.

Eidesstattliche Versicherung

Ein gerade für Diplomarbeiten überaus wichtiger Gesichtspunkt ist das einem Schwur gleichgestellte Versprechen des Autors, alle Inhalte seiner Arbeit selbst verfasst und alle Zitate entsprechend gekennzeichnet zu haben. Diese, auch als „Ehrenwörtliche Erklärung“ bezeichnete, Eidesstattliche Versicherung kommt der Unterschrift des Autors gleich und hat sich somit auf jeden Fall am Ende der wissenschaftlichen Abhandlung zu befinden.

Festlegung der Reihenfolge

Eine genau definierte Reihenfolge der Verzeichnisse am Ende einer wissenschaftlichen Arbeit konnte nicht nachvollzogen werden, da sich in diesem Punkt die Meinungen an den jeweiligen Präferenzen zu orientieren scheinen [2, 6, 13, 15]. Zur Klärung dieser oft umstrittenen Frage kann jedoch eine ganz pragmatische Betrachtungsweise herangezogen werden.

Betrachtet man den Inhalt eines speziellen Verzeichnisses etwas genauer, dann kann festgestellt werden, dass dessen Inhalt einen stärkeren oder schwächeren Bezug zum Inhalt der restlichen Arbeit aufweist. Somit kann die Stärke dieses Bezugs als Indikator zur Festlegung der Position des betreffenden Verzeichnisses herangezogen werden.

| | |
|------------------|---|
| <i>Anhang</i> | Wie zuvor oben angedeutet, beinhaltet der Anhang nur die Informationen, die aus verschiedenen Gründen im Kerntext nicht untergebracht werden konnten. Hierdurch besteht eine sehr stark ausgeprägte Bindung zum Inhalt der restlichen Abhandlung und daher sollte sich der Anhang direkt an den Textteil anschließen. |
| <i>Literatur</i> | Da ein Literaturverzeichnis zum Nachweis und zur Nachvollziehbarkeit der aufgestellten Thesen dient, ist hierdurch eine stark ausgeprägte Bindung zum restlichen Inhalt festzustellen. Aus diesem Grund sollte sich das Literaturverzeichnis an den Anhang anschließen. |
| <i>Glossar</i> | Das Glossar erläutert spezielle Begriffe, auf die sich innerhalb der Abhandlung bezogen wurde. Diese Begriffserklärungen sind aber zum Verständnis der eigentlichen Abhandlung |

nicht zwingend erforderlich, zumindest sollte dies so sein. Daher ist für ein Glossar eine auch eher schwache Bindung zum restlichen Inhalt festzustellen und somit sollte das Glossar dem Literaturverzeichnis folgen.

Index Ein Index listet bestimmte Begriffe einer Abhandlung in alphabetischer Reihenfolge, ohne dabei auf bestimmte Bedeutungen Rücksicht zu nehmen. Daher kann davon ausgegangen werden, dass die Bindung des Indexes zum restlichen Inhalt am schwächsten von allen anderen ausfällt und daher sollte der Index dem Glossar folgen.

Versicherung Für die Eidesstattliche Versicherung könnte die zuvor gezeigte Argumentationskette ebenfalls fortgeführt werden, jedoch ist ihre Bedeutung eine andere. Denn die Eidesstattliche Versicherung ist die Erklärung des Autors, dass er nicht „abgeschrieben“ hat und wirkt damit wie eine Unterschrift. Und eine Unterschrift steht nun mal unter einem Dokument. Somit muss die ehrenwörtliche Erklärung der letzte Bestandteil einer wissenschaftlichen, respektive einer Diplomarbeit sein!

Seitenvorschub bei Verzeichnissen

Grundsätzlich gilt auch für die Verzeichnisse des Dokumentenendes das bereits in Kapitel 2.4.1 auf Seite 23 Gesagte. Jedoch sind die hier befindlichen Verzeichnisse häufig sehr ausladend. Daher wird in der Praxis zumeist darauf verzichtet, die am Ende einer wissenschaftlichen Abhandlung vorhandenen Verzeichnisse auf fortlaufenden Seiten unterzubringen.

Dies gilt aber auf keinen Fall für die Eidesstattliche Versicherung. Denn diese hat sich immer auf der letzten Seite der Abhandlung zu befinden und sie ist stets von allen anderen Seite separiert zu betrachten!

2.5 Seitennummerierung

Dieser Abschnitt greift erneut die einzelnen Komponenten einer wissenschaftlichen Arbeit auf und betrachtet dabei die Nummerierung der einzelnen Seiten. Diese Betrachtungen können und sollten losgelöst von allen anderen Blickwinkeln geschehen, da eine Seitennummerierung bis auf wenige wichtige Zusammenhänge abstrahiert werden kann.

Dokumentbeginn

Wie bereits geschildert, besteht der Beginn eines wissenschaftlichen Dokuments aus einem Vorspann, der von verschiedentlichen Verzeichnissen mit römischer Seitennummerierung, beginnend bei eins, gefolgt wird [12, 13, 5]. Tabelle 1 zeigt eine Übersicht der einzelnen Bestandteile in Abhängigkeit von ihrer jeweils anzuwendenden Seitennummerierung.

Textteil und Dokumentende

Beginnend mit dem ersten Abhandlungskapitel, also zu Anfang des Textteils, wird von der zuvor römischen auf eine arabische Seitennummerierung gewechselt [12, 13, 5], die ebenfalls bei eins beginnt.

Die zu diesem Zeitpunkt vorgenommene Änderung der Art der Nummerierung verfolgt den Zweck, dem Leser deutlich mitzuteilen, dass an dieser Stelle nun mit dem inhaltlichen Teil begonnen wird.

Tabelle 1: Seitennummerierung des Dokumentbeginns

| Komponente | Bestandteile | Seitennummerierung |
|---------------|-----------------------|-----------------------------|
| Vorspann | Titelseite | keine |
| | Sperrvermerk | keine |
| | Danksagung | keine |
| | Vorwort | keine |
| Verzeichnisse | Inhaltsverzeichnis | römisch, bei eins beginnend |
| | Abkürzungsverzeichnis | römisch, fortlaufend |
| | Abbildungsverzeichnis | römisch, fortlaufend |
| | Tabellenverzeichnis | römisch, fortlaufend |

Diese arabische Seitenzählung wird bis zum Ende der Abhandlung beibehalten. Tabelle 2 illustriert die Auswirkungen der neu gewählten Nummerierung für alle restlichen Teile der wissenschaftlichen Arbeit.

Tabelle 2: Seitennummerierung des restlichen Dokuments

| Komponente | Bestandteile | Seitennummerierung |
|-------------------|-------------------------------|------------------------------|
| Textteil | Kapitel eins | arabisch, bei eins beginnend |
| | weitere Kapitel | arabisch, fortlaufend |
| Anhang | alle Bestandteile | arabisch, fortlaufend |
| Verzeichnisse | Literatur | arabisch, fortlaufend |
| | Glossar | arabisch, fortlaufend |
| | Index | arabisch, fortlaufend |
| | Versicherung (eidesstattlich) | keine |

3 Implementierung

Dieser Abschnitt vertieft Details, die im Zusammenhang mit der Umsetzung des Pakets `fomsdt` stehen. Dabei werden neben der eigentlichen Implementierung auch die Bereiche der Erstellung des Handbuchs und die Erzeugung einer zur Auslieferung geeigneten Version diskutiert. Doch vorab aber noch einige kurze Anmerkungen zu verschiedenen Schwierigkeiten und Problemen, die während der Umsetzung auftraten.

Basierend auf einer \LaTeX -Präambel, die zum Zweck der Erstellung von Fallstudien und Seminararbeiten im Fachbereich Informatik der Fachhochschule für Oekonomie & Management in Essen erstellt wurde, begann im Sommer 2007 die Implementierungsphase des *Pakets zu Erstellung wissenschaftlicher Arbeiten*.

Diese Präambel beinhaltete bereits zu Implementierungsbeginn einige wesentliche Kommandos zur Textformatierung. Jedoch war relativ schnell abzusehen, dass sich eine simple Präambel schlecht bis ungenügend als allgemeine Vorlage für wissenschaftliche Arbeiten eignen würde. Dieser Umstand bedingte dann die Suche nach einer geeigneten und vernünftigen Strukturierung der einzelnen Bestandteile der ursprünglichen Präambel.

Nach umfangreichen Recherchen und einem Vergleich der verfügbaren Möglichkeiten wurde schlussendlich die Entscheidung getroffen, eine neue Dokumentenklasse zu erzeugen, die ihre funktionsorientierten Bestandteile in voneinander unabhängige Komponenten auslagert. Dieser Entschluss hatte jedoch weitere Anstrengungen zur Folge, um allen zukünftigen Anwendern das Umsetzungsergebnis auch bereitstellen zu können. Hierdurch wurde dann die Generierung eines in \LaTeX üblichen Installationsskriptes notwendig.

Die eventuellen Bedürfnisse zukünftiger Autoren standen ebenfalls im Mittelpunkt der planerischen Betrachtungen. Hierbei ergab sich, dass die Fähigkeiten der in der Basispräambel verfügbaren Kommandos stark auf spezifische Belange abgestimmt waren. Somit musste auch eine allgemeingültige Form der Bereitstellung von Kommandos zur Textformatierung gefunden werden. Dabei war zu berücksichtigen, dass die Konfiguration der einzelnen Kommandos mit Hilfe von optionalen und obligatorischen Parametern möglich sein sollte.

Da es sich bei \LaTeX um eine der Hypertext Markup Language (HTML) vergleichbare Makrosprache handelt, brachte die Forderung nach verschiedenen optionalen und obligatorischen Parametern zwei grundlegende Schwierigkeiten hervor. Zum einen besteht innerhalb des \LaTeX -Systems die Einschränkung, dass Kommandos nur maximal neun Parameter verwenden dürfen und zum anderen ist die Anzahl der möglichen optionalen Parameter auf einen Kann-Wert pro Kommando limitiert.

Die Beschränkung auf maximal neun Argumente je Kommando kann zwar mit Hilfe eines Tricks⁴ durchbrochen werden, jedoch war dessen Anwendung für eine erfolgreiche Umsetzung nicht notwendig. Denn für das einzige davon betroffene Kommando wurde aus Gründen der besseren Übersichtlichkeit eine andere Alternative gewählt.

Optionale Kommandoparameter

Eine größere Herausforderung stellte die Bereitstellung mehrerer optionaler Kommandoparameter dar. Zur Lösung dieses Problems wurde das aus der objektorientierten Programmierung bekannte Verfahren der Methodenüberladung angewendet und an die Gegebenheiten des L^AT_EX-Systems angepasst.

Da leider eine genaue Adaption der Überladung von Kommandos in L^AT_EX nicht umsetzbar ist, wurde hierfür ein kleiner Trick angewendet. Dabei wird ein zweites, beinahe gleichnamiges Kommando erzeugt, das über eine äquivalente Liste von Parametern verfügt. Nur sind die Argumente des Zwillings so definiert, dass sie bei einem Aufruf angegeben werden müssen. Die Übergabe aller optionalen Parameter erfolgt dann in eckigen Klammern.

Die Aufgabe des Hauptkommandos, oder besser des Hauptmakros, besteht nun darin, alle durch den Anwender angegebenen Parameter in die benötigte Form zu überführen und mit diesen das Zwillingmakro aufzurufen. Wurden durch den Anwender ein oder mehrere optionale Parameter ausgelassen, dann wird das Hauptmakro an seinen Zwilling stattdessen die hier definierten Standardwerte übergeben. Die Definition eines Makros mit zwei optionalen und einem Pflichtparameter zeigt das Beispiel in Quelltext 1.

Quelltext 1: Makro mit zwei optionalen Parametern (Beispiel)

```
% Makro für den Benutzer
\newcommand{\Macro}[1][A]{% A = Standard für Parameter 1!
  \ifnextchar[%] % Zweiter optionaler Parameter vorhanden?
  {\@iiMacro[#1]}
  {\@iiMacro[#1][B]} % B = Standard für Parameter 2!
}

% Internes Makro mit allen Parametern.
\def\@iiMacro[#1][#2]#3{ % Übergabe aller Parameter.
  Param1=#1; % Verwendung optionaler Parameter 1.
  Param2=#2; % Verwendung optionaler Parameter 2.
  Param3=#3; % Verwendung Pflichtparameter des Makros!
```

Zur Vermeidung eines direkten Durchgriffs auf das nur intern verwendete Makro wird diesem ein @-Zeichen beigelegt. Hierdurch wird sichergestellt, dass

⁴ Der Trick zur Umgehung der limitierten Anzahl der Kommandoargumente kann in den *TeX Frequently Asked Questions on the Web* unter <http://www.tex.ac.uk/cgi-bin/texfaq2html?label=moren9> (letzter Zugriff: 2007-12-25) nachgeschlagen werden.

ein Anwender von außen nur das Hauptmakro aufrufen kann. Zusätzlich wird die Benennung des internen Makros durch ein Doppel-I eingeleitet. Dabei handelt es sich um einen in L^AT_EX üblichen Weg, dem Betrachter des Quellcodes die Anzahl der optionalen Parameter anzuzeigen.

In diesem Zusammenhang muss jedoch klargestellt werden, dass L^AT_EX keine lokalen Makros und auch keine privaten Variablen unterstützt. Zwar erzeugt die Verwendung des @-Zeichens beim Kompilieren eines Dokuments einen Fehler, doch kann dieser mit leichter Hand unterdrückt werden, wenn dem Aufruf eines damit gespickten Makros das Kommando `\makeatletter` vorangestellt wird.

Optionale Umgebungsparameter

Lange Zeit war ungeklärt, ob in L^AT_EX überhaupt eine Möglichkeit besteht, auch so genannte Umgebungen mit optionalen Parameterlisten auszustatten. Dieses Problem konnte nach mehreren vergeblichen Versuchen und weiteren intensiven Recherchen schließlich gelöst werden.

Letztlich stellte sich heraus, dass der gesuchte Mechanismus bereits bei allen zuvor durchgeführten Makroimplementierungen erfolgreich angewendet wurde. Somit musste das dort verwendete Verfahren nur auf die Besonderheiten einer L^AT_EX-Umgebung angepasst werden.

Quelltext 2: Umgebung mit zwei optionalen Parametern (Beispiel)

```
% Umgebung für den Benutzer
\newenvironment{Environment}[1][A]{% A = Standard für ...
...Parameter 1!
\@ifnextchar[%] % Zweiter optionaler Parameter vorhanden?
{\begin{@iiEnvironment}[#1]}
{\begin{@iiEnvironment}[#1][B]}% B = Standard für ...
...Parameter 2!
}{
\end{@iiEnvironment}
}

% Interner Umgebungsbeginn mit allen Parametern.
\def\@iiEnvironment[#1][#2]#3{ % Übergabe aller Parameter.
Param1=#1; % Verwendung optionaler Parameter 1.
Param2=#2; % Verwendung optionaler Parameter 2.
Param3=#3} % Verwendung Pflichtparameter der Umgebung!

% Internes Umgebungsende ohne Parameter.
\def\end@iiEnvironment{} %
```

Quelltext 2 zeigt das Grundgerüst zur Definition einer Umgebung mit zum Beispiel zwei optionalen und einem Pflichtparameter, dessen Funktionsweise mit der Parameterverarbeitung der zuvor diskutierten Makrodefinition identisch ist.

Beide zuvor gezeigten Methoden zur Verwendung mehrerer optionaler Argumente wurden während der Implementierung häufig angewendet. Sie bilden somit die Basis aller Makros und aller Umgebungen, die dem Endanwender durch das Paket `fomsdt` bereitgestellt werden.

3.1 Paketstruktur

Wie zuvor bereits angedeutet, war neben der eigentlichen Umsetzung auch die Frage nach einer geeigneten Paketstruktur zu klären. Eine Betrachtung dieses Teilbereichs der Implementierungen soll Gegenstand der nun anschließenden Ausführungen sein.

Insgesamt standen zur Strukturierung des Pakets `fomsdt` drei mögliche Lösungsansätze zur Auswahl, bei denen es sich im Einzelnen um die Bereitstellung einer einfachen Präambel, die Implementierung einer einfachen Stildatei sowie um die Definition einer völlig neuen Dokumentenklasse handelt.

Eine Analyse der Vor- und Nachteile aller zur Entscheidung stehenden Strukturvarianten ergab, dass die Verwendung einer eigenständigen Dokumentenklasse, die über zusätzliche und funktionsorientierte Stile verfügt, einem Endbenutzer die größtmögliche Flexibilität bieten würde. Daher fiel die Wahl der Paketstruktur auf eine Kombination aus

- Erzeugung einer neuartigen Dokumentenklasse und
- Bereitstellung weiterer Stildateien.

Dabei obliegt der Dokumentenklasse die Verarbeitung allgemeingültiger Grundeinstellungen und die verschiedenen Stildateien dienen der Behandlung spezieller Anwendungsfälle, die ihrer Art entsprechend verschiedenen Gruppen zugeordnet werden können.

Im weiteren Analyseverlauf konnte festgestellt werden, dass eine Unterscheidung zwischen textorientierten und gleitobjektorientierten Elementen sich hervorragend als Basis für eine Gruppierung eignen würde.

Weiterhin konnte eine sinnvolle Unterteilung der verschiedenen Bereiche einer wissenschaftlichen Arbeit in Elemente für den Beginn und Elemente für das Ende eines Dokuments getroffen werden.

Übrig blieben nun noch einige wenige Makros der Urpräambel, die keiner der anderen Gruppe zugeordnet werden konnten. Daher wurden diese Elemente in einer eigenen Gruppe „Vermischtes“ zusammengefasst.

Das Ergebnis der zuvor angestellten Betrachtungen hatte eine Aufteilung der jeweiligen Gruppen in verschiedene Dateien zur Folge. Diese Dateien sollen nun erst einmal kurz mit ihren Kernaufgaben vorgestellt werden.

| | |
|-------------------------|--|
| <code>fomsdt.cls</code> | Die Definition der Dokumentenklasse zur Erstellung wissenschaftlicher Arbeiten ist Kernaufgabe der Datei <code>fomsdt.cls</code> . Nebenbei werden hierin auch verschiedene Einstellungen zum Blattrand, der Absatzgrundformatierung, der Formatierung von Kopf- und Fußzeilen und auch der Fußnotenformatierung vorgenommen. Weiterhin werden durch diese Datei alle Makros zur Konfiguration und Erzeugung des Titelblattes bereitgestellt. Die Makros zur Anpassung und Verwendung von Sperrvermerk und Eidesstattlicher Versicherung sind ebenfalls in der Datei <code>fomsdt.cls</code> hinterlegt. |
| <code>fomdss.sty</code> | Die Bereitstellung der Makros zur Erzeugung der im Dokumentenbeginn vorzufindenden Verzeichnisse ist Aufgabe der Datei <code>fomdss.sty</code> . Dazu zählen das Inhaltsverzeichnis, das Abkürzungsverzeichnis, das Abbildungsverzeichnis und auch das Tabellenverzeichnis. |
| <code>fomdes.sty</code> | Die Definition der Makros zur Erzeugung der Verzeichnisse am Ende eines Dokumentes obliegt der Datei <code>fomdes.sty</code> . Hierin befindet sich unter anderem das Marko zur Erstellung des in wissenschaftlichen Abhandlungen außerordentlich wichtigen Literaturverzeichnisses. Makros zur Generierung des Indexes, eines Glossars und auch des Verzeichnisses für Codelistings werden ebenfalls in dieser Datei definiert. |
| <code>fompar.sty</code> | Speziell zur Formatierung von Absätzen und anderen Textpassagen gedachte Makros sind Bestandteil der Datei <code>fompar.sty</code> . Hierzu zählen beispielsweise Makros zur Erzeugung eingerückter Absätze, die über einen vorangestellten Bezeichner verfügen. Eine Umgebung zur beidseitigen Randeinrückung, die sich zum Beispiel sehr gut zur Hervorhebung von Zitaten eignet, ist ebenfalls in dieser Datei enthalten. |
| <code>fomfod.sty</code> | Makros zur formatierten Einbindung von Abbildungen, Tabellen und Codelistings in wissenschaftliche Abhandlungen werden durch die Datei <code>fomfod.sty</code> bereitgestellt. |

`fomisc.sty` Alle anderen, nicht kategorisierbaren Makros des Pakets `fomsdt` sind in der Datei `fomisc.sty` zusammengefasst. Beispielsweise wird hierin ein Makro zur normungskonformen Darstellung des Datums oder auch Makros zur Einbindung der speziellen Symbole für *Registered*, *Copyright* und *Trademark* definiert.

Ein unabhängig von einer Gruppierung nach Zugehörigkeit zu betrachtender Aspekt stellt die Manipulation des Indexes dar. Da dieser Teil des Pakets `fomsdt` eine Ausnahme in Bezug auf die Verwendung durch den Endanwender bedeutet, muss dennoch an dieser Stelle zumindest darauf eingegangen werden. Denn die zur Anpassung des Indexes notwendige Konfiguration muss den Regeln des L^AT_EX-Systems folgen und ist somit in einer separaten Datei auszulagern! Vom inhaltlichen Zusammenhang gehört die Indexkonfiguration jedoch zu den Elementen des Dokumentenendes.

`fomidx.ist` Die Datei `fomidx.ist` enthält verschiedene Konfigurationen für die ansprechendere Gestaltung des Indexes einer wissenschaftlichen Arbeit. Sie wird durch das Programm `makeindex` verwendet und ist daher nur indirekt zu verwenden. Zudem enthält diese Datei keine Makros im klassischen Sinne.

3.2 Paketelemente

Dieses Kapitel riskiert nun endlich einen Blick auf die Ergebnisse der Implementierung des Pakets `fomsdt`. Um einen Gesamtüberblick zu erhalten und das Zusammenspiel der jeweiligen Bestandteile des *Pakets zur Erstellung wissenschaftlicher Arbeiten* zu erarbeiten, wird sich hierfür auf eine abstrakte Betrachtung der einzelnen Komponenten beschränkt. Für das eingehendere Studium der einzelnen Quelltexte wird empfohlen, die im Anhang befindlichen vollständigen Sourcecodes einzusehen.

3.2.1 Dokumentenklasse

Basierend auf der L^AT_EX-eigenen Klasse `article` definiert die Datei `fomsdt.cls` die neue, eigenständige Dokumentenklasse des *Pakets zur Erstellung wissenschaftlicher Arbeiten*. Dabei meinen die einzelnen Buchstaben des Dateinamens nichts anderes als *FOM Scientific Document Template*, also Vorlage für wissenschaftliche Dokumente.

Neben einer Bereitstellung der Dokumentenklasse hat die Datei `fomsdt.cls` noch die Aufgabe, die verschiedenen Grundeinstellungen für wissenschaftliche

Abhandlungen vorzunehmen. Dazu zählen unter anderem das Einstellen der Schriftgröße auf zwölf Punkt und die vollständige Verwendung des beschreibbaren Bereichs eines Blattes der Größe DIN A4.

Das Laden verschiedener Basispakete zur Unterstützung spezieller Belange der deutschen Sprache gehört ebenso zu den Aufgaben der Datei `fomsdt.cls` wie auch die Konfiguration der Kopf- und Fußzeilen. Verschiedene Anpassungen für Fußnoten werden ebenfalls hierin vorgenommen.

Ein weiteres Merkmal der Datei `fomsdt.cls` ist die Festlegung der zu Grunde liegenden Absatzformatierung. Hierunter fallen unter anderem die Justage des Abstands zwischen aufeinander folgende Absätze und auch die Unterdrückung des Einzugs der ersten Absatzzeile. Weitere Einstellungen dienen der Verhinderung einzelner Zeilen auf vorherigen und folgenden Textseiten, die in der Sprache der Schriftsetzer als Schusterjungen und Hurenkinder bezeichnet werden.

Diese Einstellungen sind allesamt von grundlegender Natur und im deutschen Sprachraum allgemein gängige Randbedingungen zur Dokumentenerstellung. Daher bietet die Dokumentenklasse des Pakets `fomsdt` noch weitere Komponenten, die speziell auf die Erstellung wissenschaftlicher Ausarbeitungen zugeschnitten sind. Diese Komponenten sollen nun näher betrachtet werden.

Klassensoptionen

Die Dokumentenklasse selbst verfügt über eigene nützliche Optionen zur Konfiguration des Layouts der gesamten Arbeit, die nun der Reihe nach vorgestellt werden sollen.

coloredlinks

Die Option `coloredlinks` forciert die Verwendung farbiger Hyperlinks zur Hervorhebung von Querverweisen innerhalb eines Dokuments, da standardmäßig alle Querverweise in Schwarz dargestellt werden. Möchte ein Autor die Querverweise seines Dokuments nun farbig darstellen, dann kann er dies durch Angabe dieser Option erreichen. In diesem Fall wird ein dunkles Blau zur Hervorhebung der Querverweise verwendet.

noheadrule

Die Option `noheadrule` stellt die Strichstärke der Trennlinie zwischen der Kopfzeile und dem restlichen Text auf den Wert Null ein, da diese Linienstärke vom Paket `fomsdt` per Grundeinstellung auf einen Wert von $\frac{1}{2}$ Punkt voreingestellt wird. Hierdurch wird ein Autor befähigt, die optische Trennung zwischen Seitenkopf und Textteil zu unterdrücken.

nochapter

Die Option **nochapter** unterdrückt die Anzeige der Überschrift des Hauptkapitels in der Kopfzeile eines Dokuments. Somit muss ein Autor diese Kapitelüberschrift explizit ausschalten, wenn dies von ihm gewünscht wird.

Werden in den Klassenladeoptionen die Optionen **noheadrule** und **nochapter** miteinander kombiniert, dann kann hierdurch die Verwendung der Kopfzeile gänzlich unterdrückt werden!

moderntitle

Die Option **moderntitle** verwendet ein modern anmutendes Layout für die Titelseite der wissenschaftlichen Arbeit.

classictitle

Die Option **classictitle** verwendet ein klassisch anmutendes Layout für die Titelseite der wissenschaftlichen Arbeit.

Wird in den Ladeoptionen der Klasse als Stil der Titelseite weder **moderntitle** noch **classictitle** angegeben, dann können zur Erzeugung des Titelblattes weiterhin die L^AT_EX-eigenen Makros `\title`, `\author` et cetera verwendet werden. In diesem Fall wird vom Paket **fomdsd** jedoch eine Warnmeldung erzeugt, die den Benutzer darauf aufmerksam machen soll, dass er eventuell die Informationen zur Erzeugung der Titelseite vergessen hat.

Werden hingegen beide Stiloptionen gemeinsam verwendet, die Reihenfolge ist dabei unerheblich, dann wird als Standard für die Titelseite der klassisch anmutende Stil verwendet.

lockflag

Die Option **lockflag** schaltet die Verwendung des Sperrvermerks ein. Da ein Sperrvermerk in der Regel nur bei Diplomarbeiten anzutreffen ist, muss dieser durch den Autor ausdrücklich aktiviert werden!

affidavit

Die Option **affidavit** schaltet die Verwendung der Eidesstattlichen Versicherung ein. Da die ehrenwörtliche Erklärung grundsätzlich nur in Diplomarbeiten verwendet wird, ist diese ebenfalls zuvor durch den Autor zu aktivieren!

Titelseite

Das mit Abstand quellcode-intensivste Einzelmakro innerhalb des gesamten Pakets **fomdsd** ist das Makro zur Erstellung der Titelseite. Dieser Umstand ist aber nicht auf eine hohe inhärente Komplexität zurückzuführen, wie man

eventuell vermuten könnte, sondern begründet sich dadurch, dass vielerlei Anstrengungen notwendig sind, um einem Benutzer wenigstens zwei unterschiedliche Varianten eines Abhandlungstitels zu Verfügung stellen zu können.

Zudem bestehen diverse Abhängigkeiten zu anderen Makros, die einer Reduzierung der bereits vorhandenen Komplexität nicht gerade zuträglich sind. Die Aufschlüsselung dieser Vernetzungen soll Gegenstand dieses Teilabschnittes sein.

Üblicherweise werden in L^AT_EX die Titelseiten der Dokumente mit Hilfe des vordefinierten Makros `\maketitle` erstellt. Dieses Makro wird im Paket `fomstdt` ebenfalls zu Erstellung des Titels verwendet. Zu diesem Zweck wird es durch die Datei `fomstdt.cls` neu definiert. Dabei wird eine Vorauswahl anhand der angegebenen Klassenoptionen `moderntitle` und `classictitle` getroffen.

| Klassenladeoption für die Titelseite? | | |
|--|--|---|
| Moderner Titel | Klassischer Titel | keine |
| <ul style="list-style-type: none"> • Titelblatt erzeugen • Randbereich zurücksetzen • Texthöhe und -breite speichern • Texthöhe und -breite neu einstellen • Linke Textbox (rotiert) erzeugen • Text in linker Textbox ausgeben <ul style="list-style-type: none"> • Verwende <code>\General{}</code> • Rechte Textbox erzeugen • Text in rechter Textbox rechtsbündig • Text in rechter Textbox ausgeben <ul style="list-style-type: none"> • Verwende <code>\Institute{}</code> • Verwende <code>\Course{}</code> • Verwende <code>\Semester{}</code> • Verwende <code>\Subject{}</code> • Verwende <code>\Title{}</code> • Untere Textbox erzeugen • Text in unterer Textbox ausgeben <ul style="list-style-type: none"> • Verwende <code>\Professor{}</code> • Verwende <code>\Student{}</code> • Verwende <code>\Date{}</code> • Alte Texthöhe und -breite zurücksetzen | <ul style="list-style-type: none"> • Titelblatt erzeugen • Obere Textbox erzeugen • Text in oberer Textbox zentriert • Text in oberer Textbox ausgeben <ul style="list-style-type: none"> • Verwende <code>\Institute{}</code> • Verwende <code>\Course{}</code> • Verwende <code>\Semester{}</code> • Verwende <code>\Subject{}</code> • Verwende <code>\Title{}</code> • Untere Textbox erzeugen • Text in unterer Textbox ausgeben <ul style="list-style-type: none"> • Verwende <code>\Professor{}</code> • Verwende <code>\Student{}</code> • Verwende <code>\Date{}</code> | <ul style="list-style-type: none"> • Warnmeldung ausgeben! |
| Sperrvermerk ausgeben? | | |
| JA | NEIN | |
| <ul style="list-style-type: none"> • Neue Seite erzeugen • Sperrvermerk ausgeben | | |

Abbildung 9: Funktionsweise des neu definierten Makros `\maketitle`

Wurde durch Angabe der Option `lockflag` in den Klassenladeoptionen die Verwendung des Sperrvermerks aktiviert, dann werden zu dessen Erzeugung zusätzliche Angaben erforderlich!

Da eine detaillierte Betrachtung des Sperrvermerks im hieran anschließenden Teilabschnitt erfolgt, sollte an dieser Stelle zur Diskussion des Erstellverfahrens der Titelseite die Information „Der Sperrvermerk wird oder wird nicht verwendet“ hinreichend genau sein.

Nachdem durch den Benutzer das Layout der Titelseite festgelegt wurde, hat er die Aufgabe, weitere Angaben zur Titelseite vorzunehmen. Dies geschieht mit Hilfe der in Abbildung 9 angedeuteten Makros, die Gegenstand der nun folgenden Betrachtungen sein sollen. Dabei reicht ein flüchtiger Blick völlig aus, da alle gezeigten Makros ihre Daten nur speichern und nicht verarbeiten.

`\General`

Das Makro `\General` legt die Ausprägung der wissenschaftlichen Ausarbeitung fest. Hier könnte der Benutzer zum Beispiel „Diplomarbeit“ oder auch „Studienarbeit“ angeben. Dabei handelt es sich um einen optionalen Parameter, dessen Standardwert vom Paket auf *Hausarbeit* eingestellt wird.

`\Institute`

Das Makro `\Institute` legt den Namen der Hochschule fest, an der die wissenschaftliche Ausarbeitung geschrieben wurde. Es handelt sich hierbei ebenfalls um einen optionalen Parameter, dessen Standardwert auf *FOM – Fachhochschule für Oekonomie & Management Essen* voreingestellt ist.

`\Course`

Das Makro `\Course` definiert den Studiengang, für den die wissenschaftliche Arbeit erstellt wurde. Auch hier handelt es sich um einen optionalen Parameter, dessen Standardwert *Berufsbegleitender Studiengang* ist.

`\Semester`

Das Makro `\Semester` definiert das Semester, in dem die wissenschaftliche Ausarbeitung erstellt wurde. Dieser Wert ist optional und verfügt daher über keinen vordefinierten Standardwert.

`\Date`

Das Makro `\Date` legt das Datum der wissenschaftlichen Ausarbeitung fest. Hierbei handelt es sich auch um einen optionalen Parameter, dessen Standardwert *Essen, den \today* ist.

`\Subject`

Das Pflichtmakro `\Subject` wird verwendet, um das Studienfach beziehungsweise das Abhandlungsthema anzugeben. Bei Diplomarbeiten kann dieses

Kommando beispielsweise hervorragend zur Angabe von „Diplomarbeit zum Thema“ verwendet werden.

`\Title`

Das Pflichtmakro `\Title` definiert den Titel der wissenschaftlichen Arbeit. Dem ist sicherlich nichts weiter hinzuzufügen.

`\Professor`

Das Makro `\Professor` definiert den Namen des Betreuers der wissenschaftlichen Arbeit. Es verfügt, im Gegensatz zu allen vorherigen Makros, sowohl über einen optionalen als auch über einen Pflichtparameter. Dabei ist der Standardwert des optionalen Arguments $\langle Bezeichner \rangle$ auf *Betreuer*: eingestellt.

`\Student`

Das Makro `\Student` definiert den Namen des Autors der wissenschaftlichen Ausarbeitung. Es verfügt ebenfalls über einen optionalen und einen Pflichtparameter. Der Standardwert des optionalen Arguments $\langle Bezeichner \rangle$ wird hierbei mit dem Wert *Autor*: initialisiert.

Auch wenn die durchgeführten Implementierungen zur Erzeugung der Titelseite relativ umfangreich waren, ist dennoch die Benutzung der hierfür notwendigen Makros sehr leicht möglich, wie Quelltext 3 beispielhaft demonstriert.

Quelltext 3: Konfiguration und Erzeugung der Titelseite

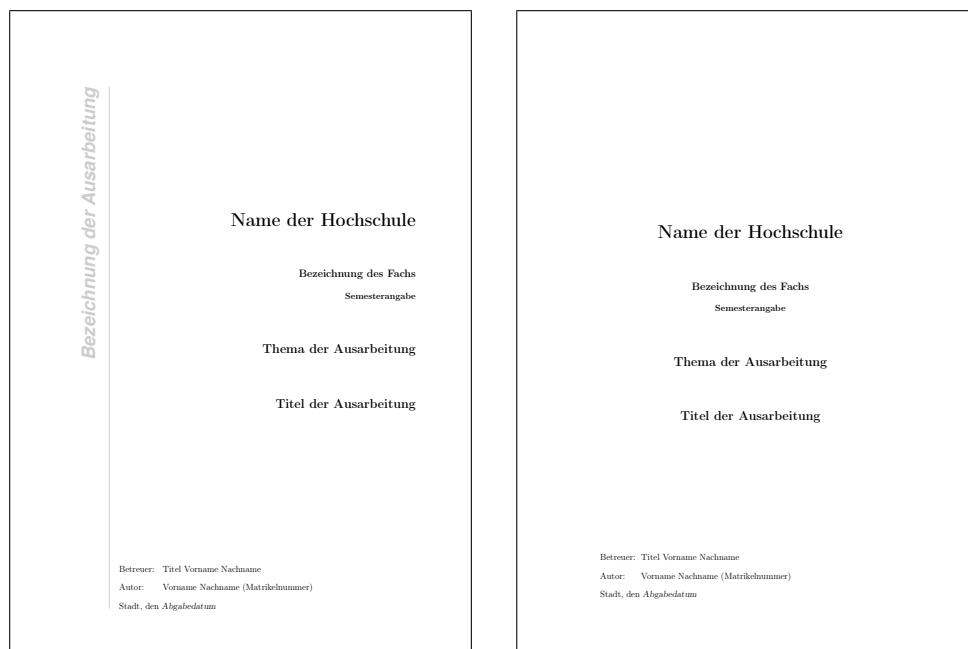
```
% Klasse laden mit modernem Titel
\documentclass[moderntitle]{fomsdt}
\begin{document}
% Titelkonfiguration
\General {Bezeichnung der Ausarbeitung}
\Date {Stadt, den \sl Abgabedatum}
\Institute{Name der Hochschule}
\Course {Bezeichnung des Fachs}
\Semester {Semesterangabe}
\Subject {Thema der Ausarbeitung}
\Title {Titel der Ausarbeitung}
\Professor[Betreuer:]{Titel Vorname Nachname}
\Student [Autor:]{Vorname Nachname ...
... (Matrikelnummer)}
% Titelerzeugung
\maketitle
\end{document}
```

Das in Quelltext 3 gezeigte Codefragment beinhaltet ein vollständiges und mit L^AT_EX kompilierbares Dokument, das über die moderne Variante des Titelblattes verfügt. Dabei übernimmt das im Paket `fomsdt` überschriebene Makro `\maketitle` alle notwendigen Aufgaben zu dessen Erzeugung.

Da in den optionalen Parametern des Makros `\documentclass` des gezeigten Beispiels keine Angabe zur Verwendung des Sperrvermerks erfolgte, also die Klassenoption `lockflag` nicht angegeben wurde, besteht ein hieraus generiertes Dokument einzig und allein aus der Titelseite.

Soll anstatt des modernen Titels doch lieber die klassische Variante zum Einsatz kommen, dann ist im Quelltext lediglich die Option `moderntitle` durch `classictitle` zu ersetzen und das Dokument erneut zu kompilieren.

Ein Beispiel für beide Variationen der Titelseite, die durch das Paket `fomstdt` unterstützt werden, zeigt Abbildung 10.



(a) Klassenoption `moderntitle` (b) Klassenoption `classictitle`

Abbildung 10: Beispiele zur Variation des Titelblattes im Paket `fomstdt`

Sperrvermerk

Da per Definition der Sperrvermerk auf der zweiten Seite der wissenschaftlichen Arbeit vorhanden sein soll, kann dieser dem Abhandlungsvorspann und somit der Titelseite zugerechnet werden. Aus diesem Grund wurde die Aufgabe des Makros `\maketitle` entsprechend erweitert, sodass die Erzeugung des Titels und die Anzeige des Sperrvermerks Hand in Hand gehen können. Dieses Verhalten deutete bereits Abbildung 9 an.

Allgemein betrachtet verfügen Sperrvermerke über einen Inhalt, der normalerweise nicht angepasst werden muss. Jedoch konnten während der Umsetzungs-

phase verschiedene Parameter ausgemacht werden, die eine Flexibilisierung des vom Paket `fom` bereitgestellten Sperrvermerks als sinnvoll erscheinen ließen.

Wie bereits weiter oben erwähnt, wird zur Aktivierung des Sperrvermerks ausschließlich die Angabe der Klassenoption `lockflag` benötigt, da dieser zusammen mit dem Abhandlungstitel automatisch generiert wird. Jedoch sind die durch den Titel festgelegten Informationen zu dessen Erzeugung unzureichend! Daher sind durch den Anwender weitere Details bereitzustellen, die nun etwas genauer unter die Lupe genommen werden sollen.

`\LockflagContent`

Mit Hilfe des Makros `\LockflagContent` ist es möglich, den grundlegenden Inhalt des Sperrvermerks zu steuern. Hierbei bestimmt der Pflichtparameter `{\Inhalt}` den zu selektierenden Basistext des Sperrvermerks wie folgt:

- `std` Wird `\LockflagContent{std}` verwendet, dann enthält der Sperrvermerk einen Standardtext. Dieser Standardtext wird bereits zum Zeitpunkt des Ladens der Dokumentenklasse voreingestellt.
- `fom` Wird `\LockflagContent{fom}` verwendet, dann enthält der Sperrvermerk einen Text, der auf die besonderen Belange der Fachhochschule für Oekonomie & Management abgestimmt ist.
- `...` Wird weder `std` noch `fom` als Parameter angegeben, dann enthält der Sperrvermerk den in den Punkten angegebenen und somit frei konfigurierbaren Text.

Ebenfalls für das Makro `\LockflagContent` verfügbar ist der optionale Parameter `[\Signatur]`. Dieser Parameter erlaubt die Aktivierung der Signatur des Sperrvermerks folgendermaßen:

- `sig` Wird als optionaler Parameter `sig` verwendet, dann enthält der Sperrvermerk eine aus den Feldern *(Ort, Datum)* und *(Eigenhändige Unterschrift)* bestehende Signatur.

Zur Anpassung des linken Signaturteils, also zur Manipulation des Inhalts des Feldes *(Ort, Datum)*, kann der Anwender das Makro `\LockflagSigLeft{\linker Signaturteil}` anwenden. Die Änderung des rechten Teils der Signatur, also die Anpassung des Inhalts von *(Eigenhändige Unterschrift)* ist ebenfalls möglich. Zu diesem Zweck kann der Benutzer das Makro `\LockflagSigRight{\rechter Signaturteil}` verwenden.

`\LockflagCompany`

Da durch einen Sperrvermerk die geheimen Informationen eines bestimmten Unternehmens geschützt werden sollen, darf der entsprechende Firmenname auch nicht im Sperrvermerk fehlen. Zur Bekanntgabe dieses Namens definiert das Paket `fomsdt` das Makro `\LockflagCompany`.

Im Zusammenhang mit dem Sperrvermerk des Pakets `fomsdt` stehen einem Anwender noch weitere, jedoch weniger wichtige Makros zur Verfügung, auf die hier nur kurz eingegangen werden soll.

`\LockflagDate`

Das Makro `\LockflagDate` kann verwendet werden, um den realen Ort sowie das Datum der Erstellung innerhalb der Signatur des Sperrvermerks zu setzen. Somit ist der Anwender frei vom Zwang, diese Information handschriftlich in das Feld (*Ort, Datum*) eintragen zu müssen.

`\LockflagHeader`

Zur Änderung der Standardüberschrift des Sperrvermerks steht dem Anwender das Makro `\LockflagHeader` zur Verfügung. Diese Makro ist beispielsweise dann besonders hilfreich, wenn anstelle der Überschrift *Sperrvermerk* ein gänzlich anderer Text verwendet werden soll.

Basierend auf dem bereits in Quelltext 3 verwendeten Codefragment sollen zur Demonstration der Möglichkeiten des Sperrvermerks hier nun noch zwei Beispiele gezeigt werden.

Beispiel für den Standardsperrvermerk ohne Signatur

Das Beispiel in Quelltext 4 zeigt den Code, der zur Erstellung eines Standardsperrvermerks notwendig ist. Dabei ist zu beachten, dass der generierte Sperrvermerk ohne zusätzlicher Signatur erzeugt wird.

Quelltext 4: Erzeugung des Standardsperrvermerks ohne Signatur

```
% Klasse laden mit modernem Titel und
% Aktivierung des Standardsperrvermerks
\documentclass[moderntitle,lockflag]{fomsdt}
\begin{document}
% Titelkonfiguration
\General {\it Bezeichnung der Ausarbeitung}
\Date    {Stadt, den \sl Abgabedatum}
\Institute{Name der Hochschule}
\Course   {Bezeichnung des Fachs}
\Semester {Semesterangabe}
\Subject  {Thema der Ausarbeitung}
\Title    {Titel der Ausarbeitung}
\Professor[Betreuer:]{Titel Vorname Nachname}
\Student  [Autor:]{Vorname Nachname ...
... (Matrikelnummer)}
```

```

% Sperrvermerkkonfiguration
\LockflagCompany{\it Name des Unternehmens}
% Titelerzeugung
\maketitle
\end{document}

```

Wurde das zugehörige Dokument kompiliert, dann stellt sich das Ergebnis des im Quelltext 4 verwendenden Codes, wie in Abbildung 11 gezeigt, dar. Zu beachten ist hierbei, dass sich der erzeugte Sperrvermerk auf der zweiten Seite des Ausgabedokuments befindet, da die erste Seite dem Titelblatt vorbehalten ist.

Sperrvermerk

Die vorliegende *Bezeichnung der Ausarbeitung* beinhaltet interne vertrauliche Informationen der Firma *Name des Unternehmens*. Die Weitergabe des Inhalts der *Bezeichnung der Ausarbeitung* im Gesamten oder in Teilen sowie das Anfertigen von Kopien oder Abschriften – auch in digitaler Form – sind grundsätzlich untersagt. Ausnahmen bedürfen der schriftlichen Genehmigung der Firma *Name des Unternehmens*.

Abbildung 11: Ausgabe des Standardsperrvermerks ohne Signatur

Beispiel für den FOM-spezifischen Sperrvermerk mit Signatur

Das Beispiel in Quelltext 5 zeigt den Code, der zur Erstellung des speziell für die Fachhochschule für Oekonomie & Management (FOM) zu verwendenden Sperrvermerks notwendig ist. Dabei ist zu beachten, dass der generierte Sperrvermerk mit zusätzlicher Signatur erzeugt wird. Das angegebene Signaturdatum wird ebenfalls verwendet.

Quelltext 5: Erzeugung des Sperrvermerks der FOM mit Signatur

```

% Klasse laden mit modernem Titel und
% Aktivierung des Standardsperrvermerks
\documentclass[moderntitle,lockflag]{fomsdt}
\begin{document}
% Titelkonfiguration
\General {\it Bezeichnung der Ausarbeitung}
\Date {\Stadt, den \sl Abgabedatum}
\Institute{Name der Hochschule}
\Course {Bezeichnung des Fachs}
\Semester {Semesterangabe}
\Subject {Thema der Ausarbeitung}
\Title {\it Titel der Ausarbeitung}
\Professor[Betreuer:]{Titel Vorname Nachname}

```



```

\Student [Autor:]{Vorname Nachname ...
... (Matrikelnummer)}
% Sperrvermerkkonfiguration
\LockflagCompany{\it Name des Unternehmens}
\LockflagDate{\it Stadt, Datum}
\LockflagContent[sig]{fom}
% Titelerzeugung
\maketitle
\end{document}

```

Nach erfolgter Kompilierung des zugehörigen Dokuments stellt sich das Ergebnis des im Quelltext 5 verwendeten Codes, wie in Abbildung 12 gezeigt, dar. Auch hier gilt es zu beachten, dass sich der erzeugte Sperrvermerk auf der zweiten Seite des Ausgabedokuments befindet.

Sperrvermerk

Die vorliegende *Bezeichnung der Ausarbeitung* mit dem Titel *Titel der Ausarbeitung* enthält unternehmensinterne Daten der Firma *Name des Unternehmens*. Daher ist sie nur zur Vorlage bei der FOM sowie den Begutachtern der Arbeit bestimmt. Für die Öffentlichkeit und dritte Personen darf sie nicht zugänglich sein.

Stadt, Datum
(Ort, Datum)

(Eigenhändige Unterschrift)

Abbildung 12: Ausgabe des Sperrvermerks der FOM mit Signatur

Abschließend gilt noch festzuhalten, dass die Verwendung des Sperrvermerks einige Abhängigkeiten zu anderen Makros des Pakets `fomsdt` verursacht! Diese Abhängigkeiten sollen nun näher aufgeschlüsselt werden.

- Wird das Makro `\LockflagContent` mit dem Argument `fom` verwendet, dann werden durch den für die Fachhochschule für Oekonomie & Management spezifischen Sperrvermerk die Daten der Makros `\General`, `\Title` und `\LockflagCompany` verwendet!
- Wird hingegen das Makro `\LockflagContent` mit dem Argument `std` verwendet, dann werden durch den Standardsperrvermerk die Daten der Makros `\General` und `\LockflagCompany` verwendet!

- Wird als Parameter des Makros `\LockflagContent` ein freier Text angegeben, dann besteht keine Abhängigkeit zu anderen Makros!
- Sollte durch die Verwendung des optionalen Parameters `sig` im Makro `\LockflagContent` die Signatur des Sperrvermerks aktiviert worden sein, dann werden in diesem Fall die Daten des Makros `\LockflagDate` verwendet!

Ehrenwörtliche Erklärung

Im Gegensatz zum Sperrvermerk ist die ehrenwörtliche Erklärung oder auch Eidesstattlichen Versicherung auf der letzten Seite einer wissenschaftlichen Abhandlung zu platzieren. Dadurch wird ihr Charakter nochmals betont. Denn sie gilt als Versprechen des Autors, alle Inhalte selbstständig erarbeitet und nicht von anderen abgeschrieben zu haben. Dies wird dann auch durch die eigenhändige Unterschrift des Autors bezeugt.

Da es sich bei der Eidesstattlichen Versicherung um einen Pflichtbestandteil für Diplomarbeiten handelt, wird diese vom Paket `fomsdt` ebenfalls in vorkonfigurierter Form zur Verfügung gestellt. Dabei orientiert sich deren Verwendung sehr stark an der Verwendung des Sperrvermerks. Somit muss auch die ehrenwörtliche Erklärung explizit aktiviert werden. Hierfür ist dann, ähnlich wie beim Sperrvermerk, die Angabe der Klassenladeoption `affidavit` erforderlich.

Wurde die Verwendung der ehrenwörtlichen Erklärung durch den Benutzer angestoßen, dann wird diese während der Erstellung des Dokuments automatisch erzeugt. Zur Sicherstellung, dass sich das Autorenversprechen auch tatsächlich auf der letzten Abhandlungsseite befindet, wird in der Datei `fomsdt.cls` das L^AT_EX-eigene Makro `\AtEndDocument` verwendet.

Wie schon angedeutet, bestehen zwischen Sperrvermerk und Eidesstattlicher Versicherung sehr viele Parallelen. Diese Gemeinsamkeiten spiegeln sich sowohl im Umfang als auch in der Benennung der zur Konfiguration verfügbaren Kommandos wieder. Daher wird es als sinnvoll erachtet, an dieser Stelle nicht allzu intensiv auf sämtliche Details der Erzeugung der ehrenwörtlichen Erklärung einzugehen, sondern besser die Unterschiede herauszuarbeiten.

Unüblich und somit überflüssig ist die Verwendung eines Firmennamens innerhalb einer Eidesstattlichen Versicherung, da der Autor ja hierdurch seine eigene Ehrlichkeit verbrieft soll. Daher wird vom Paket `fomsdt` auch kein dem Makro `\LockflagCompany` äquivalentes Makro für die Einbeziehung einer Firma in die ehrenwörtliche Erklärung angeboten.

Das Makro `\AffidavitContent` dient, analog zum Makro `\LockflagContent`, der Festlegung des Inhalts der Eidesstattlichen Versicherung. Es verwendet

ebenfalls sowohl den Pflichtparameter $\{\langle\textit{Inhalt}\rangle\}$ als auch den optionalen Parameter $[\langle\textit{Signatur}\rangle]$. Beide Parameter verarbeiten die Benutzereingaben auf die gleiche Art und Weise, wie dies beim Makro `\LockflagContent` der Fall ist.

Dennoch gibt es hier einen Unterschied! Denn die vorkonfigurierte Eidesstattliche Versicherung verfügt nicht über das Argument `fom`. Dieses Argument ist auch überflüssig, da hierbei eine Unterscheidung zwischen einer standardisierten Eidesstattlichen Versicherung und einer speziell für die Fachhochschule für Oekonomie & Management erstellten ehrenwörtlichen Erklärung unnötig ist.

Im Kern bedeutet das nichts anderes als dass die durch das Paket `fomsdt` verwendete Standarderklärung in ihrem Wortlaut mit der durch die Fachhochschule für Oekonomie & Management bevorzugten ehrenwörtlichen Erklärung übereinstimmt! Einzige Ausnahme ist, dass an zwei Stellen der Begriff „Arbeit“ der originären Version durch den Inhalt des Makros `\General` ersetzt wird.

Daher ist ein Abweichen von der standardisierten Erklärung nicht notwendigerweise gegeben, da diese bereits alle wichtigen Aspekte einer Eidesstattlichen Versicherung enthält. Wünscht ein Anwender dennoch eine eigene Variante, dann bleibt ihm immer noch die Verwendung des frei konfigurierbaren Textes im Makro `\AffidavitContent`.

Die anderen zur Konfiguration der Eidesstattlichen Versicherung verfügbaren Kommandos `\AffidavitDate`, `\AffidavitHeader`, `\AffidavitSigLeft` und `\AffidavitSigRight` verarbeiten ihre Daten auf die exakt gleiche Weise, wie dies bereits ihre namensverwandten Markos für den Sperrvermerk vollführen.

Wie auch schon im vorherigen Teilabschnitt, hier nun ein Beispiel zur Veranschaulichung der Verwendung der Eidesstattlichen Versicherung, basierend auf dem bereits in Quelltext 3 verwendeten Codefragment.

Beispiel für die ehrenwörtliche Erklärung ohne Signatur

Das Beispiel in Quelltext 6 zeigt den Quelltext, der für die Erstellung einer Eidesstattlichen Versicherung notwendig ist. Dabei ist zu beachten, dass die generierte Erklärung mit zusätzlicher Signatur erzeugt wird. Das angegebene Signaturdatum wird ebenfalls verwendet. Zudem wird in diesem Beispiel noch die auszugebende Überschrift geändert.

Quelltext 6: Erzeugung der ehrenwörtlichen Erklärung mit Signatur

```
% Klasse laden mit modernem Titel und
% Aktivierung der ehrenwörtlichen Erklärung
\documentclass[moderntitle,affidavit]{fomsdt}
\begin{document}
% Titelkonfiguration
\General {\it Bezeichnung der Ausarbeitung}
```

```

\Date      {Stadt, den \sl Abgabedatum}
\Institute{Name der Hochschule}
\Course    {Bezeichnung des Fachs}
\Semester  {Semesterangabe}
\Subject   {Thema der Ausarbeitung}
\Title     {Titel der Ausarbeitung}
\Professor[Betreuer:]{Titel Vorname Nachname}
\Student   [Autor:]{Vorname Nachname (Matrikelnummer)}
% Konfiguration der ehrenwörtlichen Erklärung
\AffidavitHeader{\it Versicherung (eidesstattlich)}
\AffidavitDate{\it Stadt, Datum}
\AffidavitContent[sig]{std}
% Titelerzeugung
\maketitle
% Automatische Erzeugung der ehrenwörtlichen ...
...Erklärung
\end{document}

```

Die Kompilierung des zum Codebeispiel aus Quelltext 6 gehörenden Dokuments ergibt dann das in Abbildung 13 dargestellte Ergebnis.

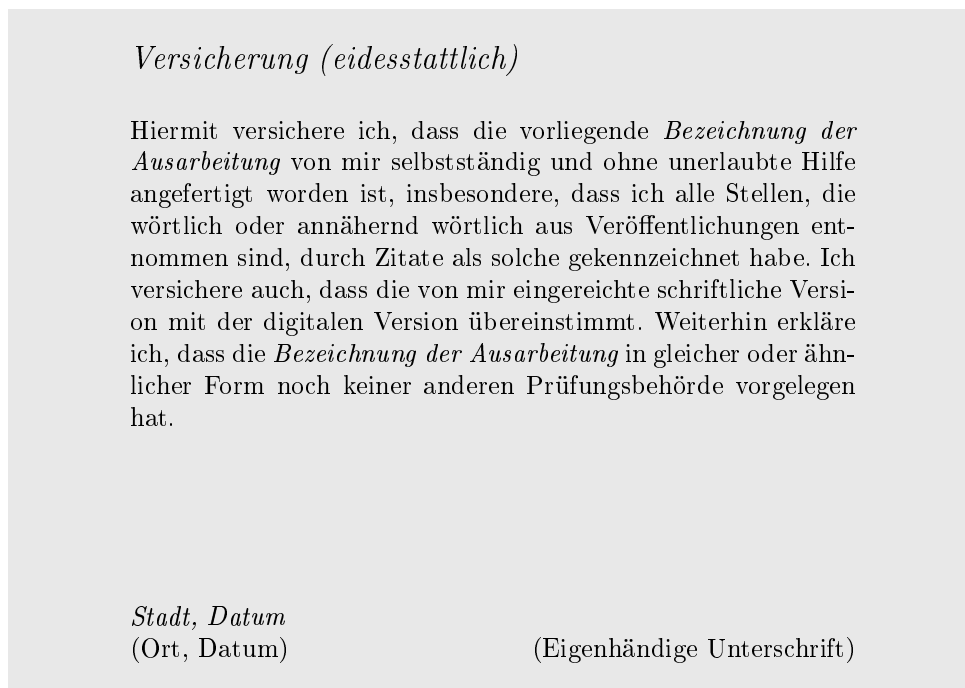


Abbildung 13: Ausgabe der ehrenwörtlichen Erklärung mit Signatur

Bleibt noch anzumerken, dass die Verwendung der ehrenwörtlichen Erklärung verschiedene Abhängigkeiten zu anderen Makros des Pakets `fomsgdt` verursacht. Diese Abhängigkeiten werden hier nun kurz zusammengefasst.

- Wird das Makro `\AffidavitContent` zusammen mit dem Argument `std` verwendet, dann wird von der Eidesstattlichen Versicherung der Inhalt des Makros `\General` verwendet!
- Wird als Parameter des Makros `\AffidavitContent` ein freier Text angegeben, dann besteht keine Abhängigkeit zu anderen Makros!
- Sollte durch die Verwendung des optionalen Parameters `sig` im Makro `\AffidavitContent` die Signatur der Eidesstattlichen Versicherung aktiviert worden sein, dann werden in diesem Fall die Daten des Makros `\AffidavitDate` verwendet!

Spezielle Makros für PDF-Dateien

Das Paket `fomsdt` bietet den Benutzern spezielle Makros, die zusammen mit der Erstellung von Dateien im PDF-Format sehr nützlich sind. Dabei besteht deren Aufgabe darin, ausgewählte Informationen der PDF-Ausgabedatei so zu verändern, dass sie in den PDF-eigenen Eigenschaften hinterlegt werden. Dies funktioniert aber nur dann, wenn zur Kompilierung des Dokuments auch das Programm `pdfLaTeX` verwendet wurde. Anderenfalls werden eventuell gesetzte Informationen ignoriert.

`\PDFTitle`

Das Makro `\PDFTitle` ändert die Eigenschaft `\Title` der PDF-Ausgabedatei. Ein Standardwert wird von diesem Makro nicht verwendet.

`\PDFAuthor`

Das Makro `\PDFAuthor` ändert die Eigenschaft `\Author` der PDF-Ausgabedatei. Ein Standardwert wird von diesem Makro nicht verwendet.

`\PDFSubject`

Das Makro `\PDFSubject` ändert die Eigenschaft `\Subject` der PDF-Ausgabedatei. Ein Standardwert wird von diesem Makro nicht verwendet.

`\PDFProducer`

Das Makro `\PDFProducer` ändert die Eigenschaft `\Producer` der PDF-Ausgabedatei und verwendet `PDFLaTeX` als Standardwert.

Wie leicht festzustellen ist, existieren bei den hier vorgestellten Makros gewisse Analogien zu einigen für das Setup der Titelseite verfügbaren Makros. Manch einer wird sich sicherlich die Frage stellen, warum nicht bereits die vorhandenen Informationen für die Speicherung in den PDF-Eigenschaften verwendet wurden. Die Begründung ist recht einleuchtend.

Da grundsätzlich die Möglichkeit besteht, zusätzliche Anweisungen zur Formatierung, sprich weitere Makros, in den Makros zur Titelerzeugung anzugeben, können die so gesetzten Informationen nicht vollständig in die PDF-eigenen Eigenschaften der Ausgabedatei übernommen werden. Somit kann es im einfachsten Fall zu hieroglyphenartigen Zeichenketten in den PDF-Eigenschaften kommen. Im schlimmsten Fall wird sogar der Kompilervorgang mit einer Fehlermeldung abgebrochen. Aus diesem Grund existieren parallele Makros, die unter Umständen auch redundante Informationen enthalten können.

3.2.2 Dokumentenbeginn

Der Beginn einer Abhandlung mit wissenschaftlichem Hintergrund ist, wie bereits mehrfach angesprochen, gekennzeichnet durch die Existenz typischer Verzeichnisse. Die Bereitstellung von Makros zur Erzeugung und Verwaltung dieser Verzeichnisse ist Aufgabe der Datei `fomdss.sty`. Dabei bedeuten die Buchstaben des Dateinamens im Einzelnen *FOM Document Start Section*.

Optionen des Dokumentbeginns

Auch der Dokumentenbeginn verfügt über eigene nützliche Optionen zur Konfiguration des Layouts der einzelnen Verzeichnisse, die hier der Reihe nach einmal vorgestellt werden sollen.

toctotoc

Die Option `toctotoc` hat die Aufgabe, die Überschrift des Inhaltsverzeichnisses ebenfalls mit in das Inhaltsverzeichnis aufzunehmen. Einige werden sich sicherlich fragen, was diese Option überhaupt für einen Nutzen hat.

Stimmt, diese Option hat keinen Nutzen und dennoch ist sie notwendig! Denn leider existieren so manche Richtlinien, in denen tatsächlich befürwortet wird, die Überschrift des Inhaltsverzeichnisses ebenfalls im Inhaltsverzeichnis aufzuführen; mit Angabe der Seitenzahl versteht sich [13, 15].

abbreviation

Die Option `abbreviation` aktiviert die Verwendung des Abkürzungsverzeichnisses. Die Einführung dieser Option wurde notwendig, da die Erstellung des Abkürzungsverzeichnisses auf Basis des Pakets `nomencl` erfolgt. Leider gehört dieses Paket noch nicht zum Standardumfang der verschiedenen \LaTeX -Distributionen und muss somit eventuell nachträglich installiert werden.

Um von vornherein Kompilierungsfehler zu vermeiden, die aufgrund des Fehlens des Pakets `nomencl` entstehen können, muss die Verwendung des Abkürzungsverzeichnisses vom Anwender ausdrücklich gewollt werden!

Außerdem kommt hinzu, dass ein mit `nomencl` erstelltes Abkürzungsverzeichnis nur mit Hilfe des Programms `makeindex` erzeugt werden kann. Daher wird von `fomdss` während der Dokumentkompilierung die Meldung

```
*****
* PLEASE NOTE! The usage of fomdss' abbreviations
* requires the nomencl package! Please ensure that
* nomencl package is installed on your LaTeX system
* in version 4.2, dated at 2005/09/22 (or higher).
* Furthermore you have to run the program 'makeindex'
* with options:
* makeindex <file>.nlo -s nomencl.ist -o <file>.nls
*****
```

ausgegeben, die zum einen den Benutzer an den Sonderstatus des Abkürzungsverzeichnisses erinnern soll und zum anderen auch direkt die weiteren Schritte zur Verzeichniserstellung aufzeigt.

abbrevrefpage

Ebenfalls im Zusammenhang mit dem Abkürzungsverzeichnis steht die Option `abbrevrefpage`. Sie kann verwendet werden, um die Seitenreferenz des ersten Auftretens der jeweiligen Abkürzung mit in das Abkürzungsverzeichnis aufzunehmen. Standardmäßig wird keine Referenzseite im Abkürzungsverzeichnis angezeigt.

Sicherlich ist jedem bewusst, dass die Option `abbrevrefpage` nur dann wirksam ist, wenn `fomdss` ferner mit der Option `abbreviation` geladen wird. Anderenfalls verliert diese Option ihre Bedeutung.

nopagebreak

Die Option `nopagebreak` verhindert den erzwungenen Seitenumbruch nach dem Ende jedes Verzeichnisses. Diese Option ist dann besonders nützlich, wenn am Anfang des Dokuments nur sehr kurze Verzeichnisse vorhanden sind.

Grundsätzlich sollte aber in wissenschaftlichen Arbeiten auf die Verwendung der Option `nopagebreak` verzichtet werden, da dies für diese Art Schriftstücke eher unüblich ist.

Kommandos des Dokumentbeginns

Ungeachtet der Ladeoptionen werden durch `fomdss` unterschiedliche Makros zur Erstellung der verschiedenen Verzeichnisse angeboten. Diese Makros sollen im weiteren Verlauf eingehender betrachtet werden.

\PrintContents

Das Makro `\PrintContents` dient der Erzeugung des Inhaltsverzeichnisses. Es verfügt nur über den optionalen Parameter [`\Überschrift`] zur Anpassung des

Verzeichnistitels. Der Standardwert ist mit *Inhalt* initialisiert. Weitere Einstellungen zur Generierung des Inhaltsverzeichnisses sind nicht notwendig!

`\PrintFigures`

Das Makro `\PrintFigures` dient der Erzeugung des Abbildungsverzeichnisses. Es verwendet ebenfalls nur den optionalen Parameter [*Überschrift*] zur Anpassung des Titels dieses Verzeichnisses. Als Standardwert wird *Abbildungen* verwendet. Auch hier sind keine weitere Einstellungen notwendig!

`\PrintTables`

Zur Erstellung des Tabellenverzeichnisses wird das Makro `\PrintTables` verwendet. Auch dieses Kommando unterstützt nur den einzelnen optionalen Parameter [*Überschrift*], der zur Anpassung der entsprechenden Überschrift dient. Standardwert hierfür ist *Tabellen*. Zusätzliche Einstellungen können nicht vorgenommen werden!

Die Darstellung von Beispielen für die soeben beschriebenen Makros ist an dieser Stelle leider nur schwerlich möglich. Und dennoch sind innerhalb dieser Arbeit entsprechende Anschauungsobjekte vorhanden. Denn alle korrespondierenden Verzeichnisse dieses Dokuments wurden mit Hilfe der Makros `\PrintContents`, `\PrintFigures` und `\PrintTables` erstellt!

Wurde hingegen `fomdss` zuvor mit der Option `abbreviation` geladen, dann stehen dem Anwender spezielle Makros zur Erzeugung des Abkürzungsverzeichnisses zur Verfügung.

`\PrintAbbrev`

Möchte der Anwender auch ein Abkürzungsverzeichnis erzeugen, dann kann er hierfür das Makro `\PrintAbbrev` verwenden. Auch dieser Makro verfügt über den optionalen Parameter [*Überschrift*] zur Anpassung des Titels. Als Standardtitel wird *Abkürzungen* verwendet.

Im Gegensatz zur Erstellung der Verzeichnisse für Inhalt, Abbildungen und Tabellen muss der Benutzer die Inhalte des Abkürzungsverzeichnisses eigenständig eintragen. Dabei besteht seine Aufgabe darin, die im laufenden Text benutzten Abkürzungen bei ihrer ersten Verwendung einmalig zu definieren. Zu diesem Zweck verwendet er dann die im Folgenden gezeigten Kommandos.

`\Abbrev`

Mit Hilfe des Makros `\Abbrev{<Abkürzung>}{<Beschreibung>}` werden innerhalb des normalen Textes die Passagen markiert, die später in das Abkürzungsverzeichnis übernommen werden sollen. Dabei beinhaltet der erste Parameter die Abkürzung selbst und der zweite Parameter enthält die Bedeutung der

im ersten Parameter angegebenen Abkürzung. Siehe hierzu nachfolgendes Beispiel.

`\Mark`

Das Makro `\Mark{⟨Buchstabe⟩}` dient der Hervorhebung der relevanten Buchstaben innerhalb der Abkürzungsbeschreibung. Alle hierdurch eingefassten Buchstaben werden später im Abkürzungsverzeichnis unterstrichen dargestellt.

Das Zusammenspiel der Makros `\PrintAbbrev`, `\Abbrev` und `\Mark` lässt sich am besten anhand einer beispielhaften Anwendung zeigen. Daher sollen nun insgesamt zwei Beispiele für etwas mehr Aufhellung im Umgang mit dem Abkürzungsverzeichnis sorgen.

Beispiel zur Erstellung eines Abkürzungsverzeichnisses

Das Beispiel in Quelltext 7 illustriert, wie ein Anwender die Makros `\Abbrev` und `\Mark` verwenden soll, um mit Hilfe des Makros `\PrintAbbrev` hieraus ein aus zwei Einträgen bestehendes Abkürzungsverzeichnis zu erzeugen, dessen Überschrift *Verwendete Abkürzungen* lautet.

Quelltext 7: Erzeugung des Abkürzungsverzeichnisses

```
\documentclass{fomstdt}
\usepackage[abbreviation]{fomdss}
\begin{document}
% Erzeugung des Abkürzungsverzeichnisses
\PrintAbbrev[Verwendete Abkürzungen]
% Verwendung im Textteil
... Der \Abbrev{IC}{\Mark{I}ntegrated \Mark{C}ircuit} ist
einer der Kernkomponenten moderner Computersystem ...
... Die \Abbrev{PIN}{\Mark{P}ersonal ...
... \Mark{I}dentification
\Mark{N}umber} wird zur Authentifizierung verwendet ...
\end{document}
```

Nennenswert ist hierbei, dass die vollständige Definition eines für das Abkürzungsverzeichnis bestimmten Eintrages im laufenden Text vollzogen wird. Dies verbessert nicht unbedingt die Lesbarkeit des Dokumentenquelltextes, gehört aber zu den in L^AT_EX üblichen Methoden. Das Makro `\footnote` sei hierfür als Beispiel genannt.

Nach Erzeugung des durch Quelltext 7 beschriebenen Dokuments enthält das Abkürzungsverzeichnis den in Abbildung 14 dargestellten Inhalt.

Beispiel zur Anpassung des Abkürzungsverzeichnisses

Die Anpassung des Layouts des Abkürzungsverzeichnisses ist ebenfalls gegeben. Hierzu können einfach die vom Paket `nomencl` angebotenen Makros verwendet werden. Einige nützliche und hilfreiche Beispiele zur Anwendung der

Verwendete Abkürzungen

IC Integrated Circuit
 PIN Personal Identification Number

Abbildung 14: Ausgabe des Abkürzungsverzeichnisses

Kommandos aus `nomencl` sind auch Teil der Benutzerdokumentation des Pakets `fomdsd`.

Ist es beispielsweise gewünscht, die Punktfüllung zwischen der Abkürzung und ihrer Erläuterung zu entfernen und soll dann auch die Abkürzung selbst als schräg gestellter Text dargestellt werden, dann kann dies auf einfache Weise durch die Manipulation des Kommandos `\nomlabel` erreicht werden. Wenn zusätzlich noch „fett“ statt „unterstrichen“ zur Hervorhebung der einzeln markierten Erläuterungsbuchstaben verwendet werden soll, dann wird hierfür einfach das Kommando `\Mark` neu definiert.

All diese wortreich beschriebenen Anpassungen können durch schlichtes Einfügen zweier Zeilen Code in das aus Quelltext 7 bekannte Beispiel erreicht werden. Das daraus entstandene Ergebnis zeigt der Codeschnipsel in Quelltext 8.

Quelltext 8: Layoutanpassungen im Abkürzungsverzeichnis

```
\documentclass{fomdsd}
\usepackage[abbreviation]{fomdsd}
\begin{document}
% Abkürzungslabel ändern (schräg und ohne Punkte)
\renewcommand{\nomlabel}[1]{\textsl{#1}}
% Markierung der Abkürzung in 'fett' ändern
\renewcommand{\Mark}[1]{\textbf{#1}}
% Erzeugung des Abkürzungsverzeichnisses
\PrintAbbrev[Verwendete Abkürzungen]
% Verwendung im Textteil
... Der \Abbrev{IC}{\Mark{I}ntegrated \Mark{C}ircuit} ist
einer der Kernkomponenten moderner Computersystem ...
... Die \Abbrev{PIN}{\Mark{P}ersonal ...
... \Mark{I}dentification
\Mark{N}umber} wird zur Authentifizierung verwendet ...
\end{document}
```

Nach erneuter Kompilierung des durch Quelltext 8 beschriebenen Dokuments erhält das Abkürzungsverzeichnis das in Abbildung 15 gezeigte neue Layout.

Abschließend sollte noch darauf hingewiesen werden, dass die gewählte Methode zur Erzeugung des Abkürzungsverzeichnisses nicht unbedingt zu den

| Verwendete Abkürzungen | |
|------------------------|--------------------------------|
| <i>IC</i> | Integrated Circuit |
| <i>PIN</i> | Personal Identification Number |

Abbildung 15: Abkürzungsverzeichnis mit geändertem Layout

elegantesten Varianten zählt, da zum Beispiel auch eine Umsetzung auf Basis einer Art Bibliothek hätte erfolgen können.

Dennoch spricht die relativ einfache Anwendung der Makros `\Abbrev`, `\Mark` und `\PrintAbbrev` für die gewählte und auf dem Paket `nomenc1` basierende Implementierungsvariante, da der Endanwender hierdurch nicht genötigt wird, innerhalb seiner Abhandlung eine weitere Datei pflegen zu müssen.

3.2.3 Dokumentenende

Das Ende einer wissenschaftlichen Abhandlung verfügt, wie auch der Abhandlungsbeginn, über typische Verzeichnisse mit nützlichen Informationen für den Leser. Die Bereitstellung der Makros zur Erzeugung und Verwaltung dieser Verzeichnisse ist Aufgabe der Datei `fomdes.sty`. Dabei bedeuten die Buchstaben des Dateinamens im Einzelnen *FOM Document End Section*.

Optionen des Dokumentenendes

Zur Konfiguration und Anpassung des Aussehens der Verzeichnisse des Dokumentenendes stehen dem Anwender auch hier einige nützliche Optionen zur Verfügung, die nun der Reihe nach vorgestellt werden sollen.

listings

Die Option `listings` aktiviert die Verwendung des Verzeichnisses für Code-listings, da dieses Verzeichnis per Grundeinstellung deaktiviert ist. Notwendig ist diese Option deshalb, da anderenfalls die Möglichkeit der Erzeugung eines leeren Verzeichnisses bestünde, wenn die Abhandlung keine entsprechenden Quellcodes enthält. Darum muss die Verwendung des Listingverzeichnis vom Anwender explizit gewollt sein.

glossary

Die Option `glossary` aktiviert die Möglichkeit zur Erstellung eines Glossars, das häufig auch als Sachwortverzeichnis bezeichnet wird. Als Grundeinstellung ist aber die Verwendung des Glossars ausgeschaltet, da hierdurch eine

Abhängigkeit zum Paket `gloss` aufgebaut wird. Leider gehört auch das Paket `gloss` nicht zum Standardumfang der verschiedenen L^AT_EX-Distributionen und ist somit eventuell noch nachträglich zu installieren.

Falls nun ein Anwender das Paket `gloss` nicht auf seinem System installiert hat, dann darf die Abhängigkeit zum Paket `gloss` keine Fehlermeldungen hervorbringen. Dies ist auch der Grund, warum der Anwender die Verwendung des Glossars ausdrücklich aktivieren muss. Im Falle der Aktivierung wird von `fomdes` während der Kompilierung eines Dokuments die Meldung

```
*****
* PLEASE NOTE! The usage of a glossary requires to
* run: bibtex <file>.gls
*****
```

ausgegeben. Diese Paketmeldung soll den Anwender darauf hinweisen, dass zur endgültigen Erzeugung des Glossars auch ein Aufruf des Kommandos `bibtex` erforderlich wird.

glossrefpage

Ebenfalls im Zusammenhang mit der Glossarerstellung steht die Option `glossrefpage`. Sie wird dazu eingesetzt, auch die Seitenzahl eines Glossar-begriffs anzuzeigen. Standardmäßig wird keine Referenzseite im Glossar verwendet.

Sicherlich ist jedem bewusst, dass die Option `glossrefpage` nur dann wirksam ist, wenn `fomdes` ferner mit der Option `glossary` geladen wird. Anderenfalls verliert diese Option ihre Bedeutung.

nobibliography

Das Literaturverzeichnis ist ein Muss einer jeden wissenschaftlichen Abhandlung und sollte daher in diesen Arbeiten niemals ausgelassen werden. Gibt es dennoch einmal einen Grund, dass die Verwendung des Literaturverzeichnisses unterdrückt werden muss, dann kann zu diesem Zweck die Option `nobibliography` verwendet werden.

Wird hingegen diese Option *nicht* in den Stilladeparametern angegeben, dann wird durch `fomdes` die Meldung

```
*****
* PLEASE NOTE! The usage of a bibliography requires
* to run: bibtex <file>.aux
*****
```

ausgegeben. Diese Meldung dient dem Anwender als Hinweis, dass zur endgültigen Erstellung des Literaturverzeichnisses weitere Eingriffe erforderlich sind.

index

Auch die Erstellung des Indexes beziehungsweise des Stichwortverzeichnisses muss durch den Anwender zuvor aktiviert werden, da per Standardeinstellung die Indexerzeugung ausgeschaltet ist. Die Option **index** ermöglicht dem Endanwender diese Aktivierung.

Wurde in den Ladeparametern von **fomdes** die Option **index** angegeben, dann wird während der Dokumenterstellung der Benutzer durch die Meldung

```
*****
* PLEASE NOTE! Index creation requires to run:
* makeindex <file>.idx
* But it is recommended to create an index based on
* ISO style. For this run:
* makeindex <file>.idx -s fomidx.ist
* Especially for German index creation users should
* use option '-g' to ensure German word ordering
* rules according to DIN 5007. In this case run:
* makeindex <file>.idx -g -s fomidx.ist
*****
```

darin erinnert, dass noch der Aufruf des Programms **makeindex** erforderlich ist, um die tatsächliche Generierung des Indexes durchzuführen.

nopagebreak

Die Option **nopagebreak** verhindert den erzwungenen Seitenumbruch nach dem Ende jedes Verzeichnisses. Diese Option ist dann besonders nützlich, wenn sich am Dokumentenende nur sehr kurze Verzeichnisse finden lassen.

Grundsätzlich sollte in wissenschaftlichen Arbeiten jedoch auf die Anwendung der Option **nopagebreak** verzichtet werden, da dies in der Regel die Übersichtlichkeit stört und dem Leser keinen nennenswerten Vorteil bietet.

Kommandos des Dokumentenendes

In Abhängigkeit der angegebenen Ladeoptionen stehen dem Benutzer verschiedene Makros zur Erzeugung der jeweiligen Verzeichnisse zur Verfügung. Die Erläuterung der einzelnen Makros soll Gegenstand des hier anschließenden Teilabschnittes sein. Vorab jedoch noch einige Worte zum allgemeinen Verhalten aller hier vorgestellten Verzeichnismakros.

Wird einer oder auch mehrere der Parameter **listings**, **glossary**, **nobibliography** und **index** in den Ladeoptionen von **fomdes** nicht angegeben, dann bleiben dennoch alle korrespondierenden Makros erhalten. In diesem Fall enthalten die betroffenen Makros keinerlei Funktionalität!

Dieses Verhalten wurde implementiert, damit der Anwender nur an einer Stelle seiner Dokumentquellen Änderungen vornehmen muss, wenn er gedenkt, eines

der Verzeichnisse auszublenden. Das erleichtert den praktischen Umgang mit `fomdes` enorm, da es hierdurch nicht mehr zu der oft lästigen Fehlermeldung „Undefined control sequence“ kommt, wenn der Endanwender vergaß, das zugehörige Makro ebenfalls auszuschalten.

`\PrintListings`

Das Makro `\PrintListings` dient der Erzeugung des Verzeichnisses für Code listings. Es verfügt über den optionalen Parameter [`\Überschrift`], der zur Änderung des Standardtitels *Listings* verwendet wird. Weitere Einstellungen zur Erzeugung des Listingverzeichnis sind nicht notwendig! Jedoch erfolgt eine tatsächliche Verzeichniserstellung erst durch die Angabe des Parameters `listings` in den `fomdes`-Ladeoptionen. Anderenfalls bleibt die Verwendung von `\PrintListings` wirkungslos.

`\PrintGlossary`

Das Makro `\PrintGlossary` wird zum Zweck der Erstellung des Glossars verwendet. Es verfügt im Gegensatz zum vorherigen Makro über insgesamt zwei optionale Argumente sowie über einen Pflichtparameter.

Analog zum vorherigen Makro dient der erste optionale Parameter [`\Überschrift`] der Anpassung des Glossartitels, dessen Standardwert mit *Glossar* initialisiert wird.

Weiterhin besteht die Möglichkeit der Übernahme aller in der Glossarbibliothek befindlichen Einträge in das Ausgabedokument. Hierfür kann als zweiter optionaler Parameter der Wert `[all]` angegeben werden. Nur ist zu beachten, dass das Paket `gloss` im Falle der Verwendung aller Glossareinträge die Warnmeldung

```
Package gloss Warning: Term '*' on page <n> not
defined in glossary 'default' on input line <m>
```

ausgibt! Daher ist es nicht unbedingt empfehlenswert, alle in der Glossarbibliothek verfügbaren Einträge auch in das Ausgabedokument zu übertragen.

Der dritte und letzte Parameter des Makros `\PrintGlossary` ist als Pflichtargument implementiert und dient der Bekanntgabe der zu verwendenden Bibliotheksdatei. Dieser Dateiname muss ohne Dateierweiterung angegeben werden, da anderenfalls die entsprechende Bibliothek nicht gefunden werden kann.

Abschließend gilt es noch zum Makro `\PrintGlossary` anzumerken, dass die Umsetzung der automatisierten Glossarerzeugung auch ohne eine Abhängigkeit zum Paket `gloss` hätte erfolgen können. Dennoch sprachen einige Gründe für den eingeschlagenen Weg!

Zum einen verfügte bereits die Standardkonfiguration der durch Paket `gloss` bereitgestellten Glossareinträge über ein sehr ansprechendes Layout, das nur in wenigen Nuancen angepasst werden musste, und zum andern zählt die bibliotheksbasierte Erstellung des Sachwortverzeichnisses zu den mit Abstand elegantesten Varianten möglicher Verfahren. Daher wurde dieser Teil der Implementierung auf der Grundlage des Pakets `gloss` durchgeführt, auch auf die Gefahr hin, dass für eine fehlerfreie Glossarerstellung eventuell die nachträgliche Paketinstallation durch den Anwender unumgänglich ist.

`\PrintBibliography`

Das Makro `\PrintBibliography` dient der Erzeugung des so wichtigen Literaturverzeichnisses. Es verwendet ebenfalls insgesamt drei Parameter, von denen die ersten beiden als freiwillige Argumente und das dritte als Pflichtparameter ausgelegt sind.

Auch hier kann der Titel des Verzeichnisses mit Hilfe des ersten optionalen Parameters [*Überschrift*] eingestellt werden, dessen Standardwert auf *Literatur* festgelegt ist.

Hingegen kann der zweite optionale Parameter zur Optimierung des Layouts des Literaturverzeichnisses verwendet werden. Der hier angegebene Wert wird dann von `fomdes` ohne weitere Prüfung an das L^AT_EX-eigene Makro `\bibliographystyle` übergeben. Somit sollte der Anwender zumindest ein gewisses Grundverständnis dafür besitzen, was hier anzugeben ist. Der Standardwert dieses optionalen Parameters ist *dinat*, also der Dateiname `dinat.bst` ohne Dateierweiterung. Dennoch wird von der Verwendung dieses Parameters abgeraten, da durch die Datei `dinat.bst` bereits alle notwendigen Einstellungen für ein DIN-gerechtes Literaturverzeichnis vorgenommen werden!

Die Bekanntgabe der Bibliotheksdatei ist auch hier dem dritten Makroparameter, also dem Pflichtargument, vorbehalten. Dieser Dateiname muss ebenfalls ohne Dateierweiterung angegeben werden, da ansonsten die entsprechende Bibliothek vom L^AT_EX-System nicht gefunden wird.

`\PrintIndex`

Das Makro `\PrintIndex` ist das letzte Kommando, das durch `fomdes` zur Verzeichniserstellung angeboten wird. Es verfügt nur über insgesamt zwei optionale Parameter, die einer Anpassung des Verzeichnislayouts dienen.

Wie bei allen zuvor aufgeführten Makros kann auch hier der erste optionale Parameter [*Überschrift*] zur Änderung des Verzeichnistitels verwendet werden. Die Standardüberschrift des Stichwortverzeichnisses ist *Index*.

Mit Hilfe des zweiten freiwilligen Arguments [*Spalten*] ist eine Variation der Spaltenzahl des Indexes gegeben. Dabei können hier die gültigen Werte wie folgt angegeben werden:

- single** Wird `\PrintIndex[⟨Überschrift⟩]` `[single]` verwendet, dann besteht der Index aus nur einer Spalte.
- double** Wird `\PrintIndex[⟨Überschrift⟩]` `[double]` verwendet, dann wird der Index auf zwei Spalten verteilt.
- triple** Wird `\PrintIndex[⟨Überschrift⟩]` `[triple]` verwendet, dann stehen dem Index insgesamt drei Spalten zur Verfügung.

Wird der optionale Parameter `[⟨Spalten⟩]` nicht angegeben, dann verwendet der Index *double* als Standardwert für die Anzahl der Spalten und die Angabe eines fehlerhaften Arguments führt zur Ausgabe der Warnmeldung

```
Package fomdes Warning: PrintIndex: Unknown
column option '⟨value⟩'. Use 2 columns instead.
```

Leider ist auch an dieser Stelle die beispielhafte Demonstration der Verwendung der soeben beschriebenen Makros nur schwerlich möglich. Jedoch sei angemerkt, dass die Erstellung der am Ende dieser Abhandlung befindlichen Verzeichnisse mit Hilfe der zuvor vorgestellten Makros `\PrintListings`, `\PrintGlossary`, `\PrintBibliography` und `\PrintIndex` erfolgte.

`\SeeAlso`

Das Makro `\SeeAlso{⟨Querverweis⟩}` ist für die einheitliche Formatierung von Verweisen gedacht und innerhalb eines Indexes und eines Glossars ebenso nützlich wie hilfreich, da es einen kleinen Pfeil mit dahinter befindlichen schräg gestelltem Text erzeugt. Diese Art der Formatierung soll der besseren Hervorhebung der entsprechenden Verweise dienen.

Mit Hilfe des Parameters `{⟨Querverweis⟩}` kann nicht nur reiner Text verarbeitet werden! Vielmehr besteht auch die Möglichkeit, weitere Makros einzubinden, wie das kleine Beispiel in Quelltext 9 demonstriert.

Quelltext 9: Verwendung des Makros `\SeeAlso`

```
... Dieser Monitor\index{Monitor\SeeAlso{\textbf{Bildschirm}}}  
zeigt die Zustandsparameter des Gesamtsystems, ...
```

Nach erneuter Kompilierung des auf Quelltext 9 basierenden Dokuments stellt sich das fertig formatierte Ausgabeergebnis des Stichwortverzeichnisses wie in Abbildung 16 gezeigt dar.

Praktisch bestünde auch die Möglichkeit, das Makro `\SeeAlso` im laufenden Text zu verwenden. Jedoch wäre dies wenig sinnvoll, wenn hierfür keine entsprechende Referenz verfügbar ist.



M
Monitor → **Bildschirm** <Seite>
...

Abbildung 16: Ergebnis der Makroverwendung `\SeeAlso`

3.2.4 Absatzdefinition

Die spezielle Formatierung von Textpassagen ist mit \LaTeX nicht gerade einfach zu realisieren. Um nun dem Endanwender, und hier speziell den \LaTeX -Neulingen, eine einfache Formatanpassung von Fließtexten zu ermöglichen, wurde dem Paket `fomsgdt` die Datei `fompar.sty` beigelegt. Dabei bedeuten die Buchstaben des Dateinamens im Einzelnen *FOM Paragraph Definition*.

Die Angabe von Dateiladeoptionen ist hier nicht erforderlich, da alle bereitgestellten Makros so definiert sind, dass sie unabhängig von etwaigen Randbedingungen angewendet werden können.

Kommandos zur Absatzformatierung

Die in der Datei `fompar.sty` implementierten Kommandos, dazu zählen drei Kernmakros und eine Umgebung, orientieren sich allesamt an grundlegenden Belangen zur textlichen Formatierung innerhalb von wissenschaftlichen Abhandlungen. Sie sind daher sehr stark an die Bedürfnisse angepasst, die im Zusammenhang mit einer praktischen Erstellung derartiger Arbeiten auftreten.

Citation

In wissenschaftlichen Arbeiten werden in der Regel längere Zitate in eigenen und zusammenhängenden Absätzen dargestellt. Dabei sind diese Zitate nicht nur durch doppelte Anführungszeichen hervorgehoben, sondern verfügen oft auch über einen zusätzlichen linken und rechten Absatzeinzug. Die Formatierung dieser Art Textpassagen ist Aufgabe der Umgebung `Citation`.

Neben der optionalen Angabe des Abstandes zum umgebenden Text, die Standardeinstellung hierfür ist der Wert von `\parskip`, muss zur Verwendung dieser Umgebung auch der linke und rechte Einzug angegeben werden, wie das Beispiel in Quelltext 10 zeigt.

Quelltext 10: Verwendung der Umgebung `Citation`

```
... vorhergehender Text.  
\begin{Citation} % Beginn der Umgebung
```

```

{1cm}      % Linker Einzug
{3cm}      % Rechter Einzug
"Text des Zitats ..."
\end{Citation} % Ende der Umgebung
Nachfolgender Text ...

```

Nach Erzeugung des Dokuments, welches das in Quelltext 10 gezeigte Codefragment enthält, ergibt sich das in Abbildung 17 dargestellte Ausgabeergebnis.

```

... vorhergehender Text.

    „Text des Zitats ...“

Nachfolgender Text ...

```

Abbildung 17: Ergebnis der Verwendung der Umgebung `Citation`

Leider ist der unterschiedlich große Einzug auf beiden Seiten des Zitatabsatzes in Abbildung 17 nur schwerlich zu erkennen. Daher der Hinweis, dass die Angabe des linken und rechten Absatzeinzuges unabhängig voneinander erfolgt und beide Werte auch getrennt voneinander verarbeitet werden. In der praktischen Anwendung enthalten beide Parameter jedoch zumeist identische Werte.

`\Remark` und `\Item`

Die Makros `\Remark` und `\Item` dienen beide der Formatierung von Absätzen, die über einen links stehenden Bezeichner und einen davon rechts befindlichen Textkörper verfügen. Da sich beide Makros im Grunde nur in der Ausrichtung ihres Bezeichners unterscheiden, erscheint deren gemeinsame Behandlung als sinnvoll. Schließlich soll der geneigte Leser nicht dazu gezwungen werden, identischen Inhalt doppelt lesen zu müssen.

Beide Makros verfügen über insgesamt vier Parameter, von denen die ersten beiden optionale und die letzten beiden obligatorische Informationen enthalten.

Als erstes freiwilliges Argument kann der [*Einzug*] vom linken Rand an die Makros `\Remark` und `\Item` übergeben werden. Somit steuert dieser Wert den linksseitigen Abstand des Textkörpers zum Seitenrand. Wird dieses Argument nicht angegeben, dann wird für `\Remark` ein Standardeinzug von *3cm* gewählt und bei `\Item` der Standardeinzug auf *4cm* voreingestellt.

Der zweite freiwillige Parameter dient den Makros `\Remark` und `\Item` als Definition des [*Abstandes*] zum umgebenden Text. Somit kann auch die vertikale Ausrichtung des Textkörpers zum restlichen Text beeinflusst werden. Wird dieser Abstand nicht gesetzt, dann benutzen beide Makros den aktuellen Wert von `\parskip` als Standardeinstellung.

Beide Pflichtargumente werden dazu verwendet, den eigentlichen Inhalt der jeweiligen Absätze anzugeben. Dabei sollte aber beachtet werden, dass der Text des ersten Parameters $\{\langle\textit{Bezeichner}\rangle\}$ nicht auf mehrere Zeilen verteilt wird, wenn dieser über die Breite des verwendeten Einzugs hinaus geht.

Hingegen wird der Textkörper selbst wie gewohnt am Zeilenende umbrochen. Es ist sogar möglich, mehrere Unterabsätze im Textkörper anzugeben, die dann gleichfalls um den im Einzug angegebenen Wert eingerückt werden.

Weiterhin ist es möglich und sehr häufig auch empfehlenswert, den Bezeichner eines durch die Makros `\Remark` und `\Item` formatierten Absatzes noch zusätzlich hervorzuheben. Daher wurde während der Implementierung darauf geachtet, dass diese Textmanipulationen gleichfalls in der Ausgabe erscheinen.

Beispiele zur Verwendung der Makros `\Remark` und `\Item`

Das erste Beispiel soll zeigen, wie das Makro `\Remark` verwendet werden kann, um beispielsweise eine besonders wichtige Textpassage in einer wissenschaftlichen Arbeit kenntlich zu machen. Zu diesem Zweck wird angenommen, dass der Abhandlungsautor den Leser darauf hinweisen möchte, welche unter Umständen negativen Folgen die Anwendung eines bestimmten Befehls haben kann. Quelltext 11 zeigt den zur Formatierung dieses Hinweises nötigen Code.

Quelltext 11: Verwendung des Makros `\Remark`

```

\Remark[25mm]{\bf Bemerkung:}{Wird das Kommando {\tt dd}
zusammen mit den Optionen {\tt if=/dev/hda},
{\tt of=/mnt/floppy/mbr}, {\tt bs=512} und
{\tt count=1} verwendet, dann kann der
Bootsektor einer Festplatte auf Diskette
gespeichert werden.}
\Remark[25mm]{\bf Achtung!}{Von der Verwendung des ...
...Befehls
{\tt dd} mit den Parametern {\tt if=/dev/zero}
und {\tt of=/dev/hda} wird dringend abgeraten!}

```

Nach erfolgter Kompilierung des Dokuments, welches den Code aus Quelltext 11 enthält, stellt sich das Ausgabeergebnis wie in Abbildung 18 gezeigt dar.

| | |
|-------------------|---|
| Bemerkung: | Wird das Kommando <code>dd</code> zusammen mit den Optionen <code>if=/dev/hda</code> , <code>of=/mnt/floppy/mbr</code> , <code>bs=512</code> und <code>count=1</code> verwendet, dann kann der Bootsektor einer Festplatte auf Diskette gespeichert werden. |
| Achtung! | Von der Verwendung des Befehls <code>dd</code> mit den Parametern <code>if=/dev/zero</code> und <code>of=/dev/hda</code> wird dringend abgeraten! |

Abbildung 18: Ergebnis der Makroverwendung `\Remark`

Das zweite Beispiel verdeutlicht, wie mit Hilfe des Makros `\Item` eine listenartige Strukturierung aufgebaut werden kann. Dabei basiert das in Quelltext 12 verwendete Beispiel auf der Annahme, dass der Abhandlungsautor einige seiner eigenen Kommandos näher erläutern möchte.

Quelltext 12: Verwendung des Makros `\Item`

```

\Item{\tt create}{Die Funktion {\tt create} reserviert alle
Systemressourcen und weist den benötigten Speicher ...
...zu.}
\Item{\tt connect}{Die Funktion {\tt connect} stellt eine
Verbindung zum Server her.}
\Item{\tt disconnect}{Die Funktion {\tt disconnect} ...
...baut die
zuvor hergestellte Verbindung zum Server wieder ab.}
\Item{\tt destroy}{Die Funktion {\tt destroy} gibt alle
reservierten Systemressourcen und den zugewiesenen
Speicher wieder frei.}

```

Die Erstellung des Dokuments, das den in Quelltext 12 gezeigten Code enthält, ermöglicht das in Abbildung 19 dargestellte Ausgabeergebnis.

| | |
|-------------------------|--|
| <code>create</code> | Die Funktion <code>create</code> reserviert alle Systemressourcen und weist den benötigten Speicher zu. |
| <code>connect</code> | Die Funktion <code>connect</code> stellt eine Verbindung zum Server her. |
| <code>disconnect</code> | Die Funktion <code>disconnect</code> baut die zuvor hergestellte Verbindung zum Server wieder ab. |
| <code>destroy</code> | Die Funktion <code>destroy</code> gibt alle reservierten Systemressourcen und den zugewiesenen Speicher wieder frei. |

Abbildung 19: Ergebnis der Makroverwendung `\Item`

Wie bereits eingangs erwähnt und durch die Abbildungen 18 und 19 belegt, unterscheiden sich die Makros `\Remark` und `\Item` letztlich nur in der Ausrichtung ihres Bezeichners. Dieser Umstand wirft nun unweigerlich die Frage auf, warum nicht nur eines der beiden Makros entwickelt wurde. Die Antwort darauf findet sich in erster Linie in der Verwendung der optionalen Makroparameter.

Würde als Beispiel ein zusätzlicher freiwilliger Parameter zur Ausrichtung des Bezeichners vorhanden sein, dann entstünde hieraus eine unnötige Komplexität! Da in \LaTeX zwar die Implementierung optionaler Parameter möglich ist, ist dennoch ihre Reihenfolge durch das System festgelegt. Das bedeutet nichts anderes, als dass bereits zur Verwendung des zweiten optionalen Parameters die Angabe des ersten optionalen Parameters erforderlich wird. Daraus folgt, dass die Abwägung, welches der freiwilligen Argumente nun wichtiger als ein anderes ist und somit in der Reihenfolge höher priorisiert werden kann, für den hier diskutierten Anwendungsfall nur schwer möglich ist. Daher wurde während der Umsetzung entschieden, auch hier das sinnvolle Limit von maximal zwei optionalen Parametern nicht zu überschreiten.

Konfiguration der Makros `\Remark` und `\Item`

Während der Implementierung der Datei `fompar.sty` konnte festgestellt werden, dass eine global wirkende Konfiguration des Einzugs sowie des Abstandes zum umgebenden Text sinnvoll sein könnte. Zu diesem Zweck wurden weitere Makros implementiert. Da diese Makros ihre Daten analog verarbeiten, soll an dieser Stelle nur kurz darauf eingegangen werden.

Die Makros `\RemarkIndent` und `\ItemIndent` dienen der globalen Anpassung des Absatzeinzuges der korrespondierenden Makros `\Remark` und `\Item` und sind zur alternativen Verwendung zum ersten optionalen Parameter gedacht.

Mit Hilfe der Makros `\DefaultRemarkIndent` und `\DefaultItemIndent` kann ein zuvor global geänderter Absatzeinzug auf seinen Standardwert zurückgesetzt werden.

Ähnlich verhält es sich mit den Makros `\RemarkOffset` und `\ItemOffset`, die als Alternative zum zweiten optionalen Parameter gedacht sind, und den Makros `\DefaultRemarkOffset` und `\DefaultItemOffset`, welche das Zurücksetzen der jeweiligen Standardwerte erlauben. Nur werden diese Makros zur Anpassung des Abstandes zum umgebenden Text verwendet.

Zur Demonstration der Anwendung der hier vorgestellten Makros soll nun das Beispiel aus Quelltext 11 herangezogen und entsprechend modifiziert werden.

Quelltext 13: Globale Anpassung des Einzugs für Makro `\Remark`

```
\RemarkIndent{55mm}
\Remark{\bf Bemerkung:}{Wird das Kommando {\tt dd}
zusammen mit den Optionen {\tt if=/dev/hda},
{\tt of=/mnt/floppy/mbr}, {\tt bs=512} und
{\tt count=1} verwendet, dann kann der
Bootsektor einer Festplatte auf Diskette
gespeichert werden.}
\Remark{\bf Achtung!}{Von der Verwendung des Befehls
{\tt dd} mit den Parametern {\tt if=/dev/zero}
und {\tt of=/dev/hda} wird dringend ...
...abgeraten!}
\DefaultRemarkIndent
```

Nach der Erzeugung des Dokuments, welches den Code aus Quelltext 13 enthält, stellt sich das Ausgabeergebnis wie in Abbildung 19 gezeigt dar.

| | |
|-------------------|---|
| Bemerkung: | Wird das Kommando <code>dd</code> zusammen mit den Optionen <code>if=/dev/hda</code> , <code>of=/mnt/floppy/mbr</code> , <code>bs=512</code> und <code>count=1</code> verwendet, dann kann der Bootsektor einer Festplatte auf Diskette gespeichert werden. |
| Achtung! | Von der Verwendung des Befehls <code>dd</code> mit den Parametern <code>if=/dev/zero</code> und <code>of=/dev/hda</code> wird dringend abgeraten! |

Abbildung 20: Ergebnis der globalen Anpassung für Makro `\Remark`

Erst durch diese Konfigurationsmakros wird es möglich, sowohl den Einzug als auch den Abstand der Makros `\Remark` und `\Item` einheitlich zu setzen, ohne dafür jedes einzelne optionale Argument ändern zu müssen!

`\Description`

Die allgemeingültige Reglementierung, dass die Gliederungstiefe einer wissenschaftlichen Abhandlung nicht mehr als drei Ebenen umfassen darf, verursacht mancherorts einige Schwierigkeiten. Und dennoch kann sich eine sinnvoll gewählte Substrukturierung einzelner Kapitel positiv auf die Lesbarkeit des gesamten Werkes auswirken [20, Seite 8].

Daher bietet fompar das Makro `\Description` an, mit dessen Hilfe die Erzeugung von Pseudoüberschriften ermöglicht wird. Sicher wird sich der ein oder andere L^AT_EX-kundige Leser fragen, warum hierfür nicht die Makros `\paragraph` und `\subparagraph` verwendet werden.

Die Beantwortung dieser Frage ist simpel und lautet: Beide Makros gehören zu den in L^AT_EX verfügbaren Gliederungskommandos! Somit gehen sie einher mit den Makros `\part`, `\chapter`, `\section` und so weiter. Dass die Überschriften der Makros `\paragraph` und `\subparagraph` nicht im Inhaltsverzeichnis auftauchen, liegt einzig und allein daran, dass in L^AT_EX die Standardgliederungstiefe auf drei Ebenen eingestellt ist. Denn die einfache Manipulation der beiden Zähler `tocdepth` und `secnumdepth` veranlasst L^AT_EX, die durchnummerierten Inhalte von `\paragraph` und `\subparagraph` ebenfalls mit im Inhaltsverzeichnis aufzunehmen!

Damit die Eintragung der verschiedenen Teilabschnitte eines Kapitels im Inhaltsverzeichnis von vornherein ausgeschlossen werden kann, sollte immer das fompar-Makro `\Description` bevorzugt werden, anstatt die Makros `\paragraph` und `\subparagraph` zu verwenden.

Zur Benutzung des Makros `\Description` stehen dem Endanwender, neben einem freiwilligen Argument, auch zwei obligatorische Parameter zur Verfügung, die zur Bekanntgabe des Inhalts des entsprechenden Abschnittes dienlich sind. Dabei definiert der erste Pflichtparameter `{\langle Überschrift \rangle}` den Kopf und der zweite Pflichtparameter `{\langle Text \rangle}` den Körper des entsprechenden Absatzes.

Der optionale Parameter `[\langle Format \rangle]` kann zur Formatierung der Absatzüberschrift verwendet werden. Hierfür stehen dann verschiedene vordefinierte Werte bereit, die hier kurz erläutert werden sollen.

- n Die Absatzüberschrift wird nicht formatiert.
- b Die Absatzüberschrift wird **fett** dargestellt (Standardeinstellung).

- s Die Absatzüberschrift wird *schräg gestellt* dargestellt.
- u Die Absatzüberschrift wird unterstrichen dargestellt.

Was noch fehlt, ist ein Beispiel. Da die Auswirkungen des Makros `\Description` auf das Absatzlayout eher weniger spektakulär sind, werden die verschiedenen Möglichkeiten in nur einem Beispiel zusammengefasst. Hierdurch sind dann auch die jeweiligen Unterschiede besser erkennbar.

Quelltext 14: Beispiele zur Verwendung des Makros `\Description`

```
\Description[n] % Keine Formatierung
    {Ohne Formatierung}
    {Unformatierte Überschriften sind zur Hervorhebung
     meist ungeeignet.}\par
\Description[b] % Fett formatiert, kann weggelassen werden
    {Überschrift fett}
    {Als fett formatierte Überschriften sind gut zur
     Hervorhebung von Hauptabschnitten geeignet.}\par
\Description[s] % Schräg formatiert
    {Schräg gestellte Überschrift}
    {Schräg gestellte Überschriften sind gut zur
     Hervorhebung von Unterabschnitten geeignet.}\par
\Description[u] % Unterstrichen formatiert
    {Unterstrichene Überschrift}
    {Unterstreichungen jeglicher Art sollten in
     Ausarbeitungen besser vermieden werden!}
```

Nach erfolgter Kompilierung des Dokuments, das den Code aus Quelltext 14 enthält, wird das in Abbildung 21 gezeigte Layout sichtbar, dass durch Verwendung des Makros `\Description` generiert wurde.

Abbildung 21 lässt erkennen, dass der Zwischenraum von Überschrift und Textblock kleiner ist als der Abstand zum nachfolgenden Absatz. Dieses Verhalten ist keineswegs Zufall! Denn hierdurch soll dem Leser einer Abhandlung besser kenntlich gemacht werden, zu welchem Teilabschnitt die entsprechende Überschrift gehört.

3.2.5 Gleitobjektdefinition

Besonderes Augenmerk gilt in wissenschaftlichen Arbeiten der Formatierung so genannter Gleitobjekte. Dazu zählen Abbildungen, Tabellen und im weiteren Sinne auch Codelistings. Daher bietet die Gleitobjektdefinition `fomfod`, die in der Datei `fomfod.sty` implementiert ist, entsprechende Makros zur einfachen und einheitlichen Integration dieser Gleitobjekte in wissenschaftlichen Abhandlungen an. Wie man sich leicht vorstellen kann, bedeuten hier die einzelnen Buchstaben des Dateinamens *FOM Floating Object Definition*.

Ohne Formatierung

Unformatierte Überschriften sind zur Hervorhebung meist ungeeignet.

Überschrift fett

Als fett formatierte Überschriften sind gut zur Hervorhebung von Hauptabschnitten geeignet.

Schräg gestellte Überschrift

Schräg gestellte Überschriften sind gut zur Hervorhebung von Unterabschnitten geeignet.

Unterstrichene Überschrift

Unterstreichungen jeglicher Art sollten in Ausarbeitungen besser vermieden werden!

Abbildung 21: Mögliche Layouts des Makros \Description

Optionen der Gleitobjektdefinition

Zur Konfiguration und Anpassung des Aussehens der einzelnen Gleitobjekte stehen einem Anwender auch hier einige nützliche Optionen zur Verfügung, die hier erst einmal der Reihe nach vorgestellt werden sollen.

classiccaption

Zur Betonung eines Gleitobjektes ist es allgemein üblich, dessen Beschriftung in einer serifenlosen Schrift vorzunehmen. Diese serifenlose Schrift wird somit auch durch fomfod als Standardeinstellung für Objektbeschriftungen verwendet.

Wenn einem Anwender die Standardformatierung der Beschriftung nicht zusagt, dann kann mit Hilfe der Option `classiccaption` eine Anpassung der Beschriftungen unterdrückt werden. Dies hätte jedoch möglicherweise zur Folge, dass sich der entsprechende Objekttitle weniger vom umgebenden Text abhebt, als dies bei Verwendung einer serifenlosen Schrift der Fall ist.

listingcolored

Die farbliche Hervorhebung bestimmter Quelltextpassagen ist nicht selten ein geeignetes und nützliches Mittel zur Unterstreichung verschiedener Zusammenhänge. Daher wird dem Anwender die Möglichkeit geboten, die in Abhandlungen einzubindenden Quelltexte koloriert darzustellen. Diese Quellcodefärbung kann somit durch Angabe der Option `listingcolored` aktiviert werden.

Kommandos der Gleitobjektdefinition

Für die häufigsten in wissenschaftlichen Abhandlungen vorkommenden Formen von Gleitobjekten definiert die Datei `fomfod.sty` entsprechende Makros. Doch stellte sich während der Umsetzung heraus, dass sich die unterschiedlichen Gleitobjektarten nur sehr ungleich abstrahieren ließen. Dieser Umstand spiegelt sich dann auch in der Anzahl der jeweiligen Makros wieder, die nun im Anschluss einzeln vorgestellt werden sollen.

`\SinglePicture`

Abbildungen bilden in wissenschaftlichen Arbeiten die wohl am häufigsten verwendete Form von Gleitobjekten. Zu diesem Zweck hält die Datei `fomfod.sty` das Makro `\SinglePicture` bereit. Dieses Makro erlaubt dem Endanwender die Integration einer Abbildung auf einfachste Weise, da alle hierfür relevanten Informationen als Übergabeparameter definiert sind.

Das Makro `\SinglePicture` verfügt, neben den vier Pflichtparametern $\{\langle\textit{Beschriftung}\rangle\}$, $\{\langle\textit{Referenzlabel}\rangle\}$, $\{\langle\textit{Bilddateiname}\rangle\}$ und $\{\langle\textit{Quellenangabe}\rangle\}$, auch über die beiden optionalen Parameter $[\langle\textit{Breite}\rangle]$ und `[box]`, die zur individuellen Anpassung einer eingebundenen Graphik verwendet werden können.

Dabei erlaubt der erste optionale Parameter $[\langle\textit{Breite}\rangle]$ die Anpassung der Bildbreite in Abhängigkeit von der aktuell eingestellten Textbreite, welche durch das L^AT_EX-Makro `\textwidth` vorgegeben wird. Eine Angabe der Bildhöhe ist nicht notwendig, da diese proportional skaliert wird. Der Standardwert dieses optionalen Arguments ist *0.9* und somit 90 % der durch `\textwidth` beschriebenen Breite des umgebenden Textes.

Der optionale Parameter `[box]` dient der Erzeugung eines Rahmens um die eingebundene Abbildung. Da die im vorherigen Kapitel gegebenen Empfehlungen von der Darstellung eines Abbildungsrahmens abraten, ist der Standardwert dieses Parameters so eingestellt, dass keine Rahmen für Abbildungen verwendet werden.

Da die Beschriftung einer Abbildung in wissenschaftlichen Abhandlungen ein Muss ist, kann zur Definition des jeweiligen Abbildungstitels der erste Pflichtparameter $\{\langle\textit{Beschriftung}\rangle\}$ verwendet werden.

Mit Hilfe des zweiten Pflichtparameters $\{\langle\textit{Referenzlabel}\rangle\}$ hat der Anwender die Möglichkeit der Vereinbarung eines Bezeichners. Dieser Bezeichner kann dann dazu verwendet werden, um die hierzu zugehörige Abbildung innerhalb des restlichen Dokumentes zu referenzieren. Üblicherweise folgt ein Abbildungslabel dem Schema `fig:label`.

Essenziell für das Einladen externer Graphiken in einer Abhandlung ist der Name der benötigten Abbildungsdatei. Hierfür kann vom Anwender der dritte

obligatorische Parameter $\{\langle\textit{Bilddateiname}\rangle\}$ verwendet werden. Zur Steuerung des Suchpfades zu den einzelnen Abhandlungsabbildungen empfiehlt sich auch die Verwendung des L^AT_EX-eigenen Makros `\graphicspath`.

Wenn die eingefügte bildliche Darstellung einer Fremdquelle entspringt, dann ist es nicht nur redlich, sondern in wissenschaftliche Arbeiten auch unumgänglich, die Herkunft der Graphik entsprechend kenntlich zu machen. Zu diesem Zweck kann der vierte und letzte Pflichtparameter $\{\langle\textit{Quellenangabe}\rangle\}$ verwendet werden. Dabei erfolgt eine Quellenangabe typischerweise rechtsbündig und direkt unterhalb der jeweiligen Abbildung.

Beispiel zur Verwendung des Makros `\SinglePicture`

Ein Beispiel soll helfen, die einfache Anwendung des Makros `\SinglePicture` zu demonstrieren. Dabei wird von der Annahme ausgegangen, dass der Autor einem Leser die Anordnung der Kontakte einer Chipkarte anhand einer Abbildung, die in der Datei `kontakte.pdf` enthalten ist, näher bringen möchte. Das Fragment in Quelltext 15 zeigt den hierfür notwendigen Code.

Quelltext 15: Beispiel zur Verwendung des Makros `\SinglePicture`

```
\SinglePicture[0.7]% 70% der Textbreite
[box]% inklusive Rahmen
{Kontakte einer Chipkarte}% Beschriftung
{fig:ChipKontakte}% Referenzlabel
{kontakte.pdf}% Bilddateiname
{Quelle: \url{http://www.hardware-ecke.de/}}% ...
...Quellenangabe
```

Nach Erstellung des den Code aus Quelltext 15 enthaltenden Dokuments stellt sich das Ausgabeergebnis wie in Abbildung 22 gezeigt dar.

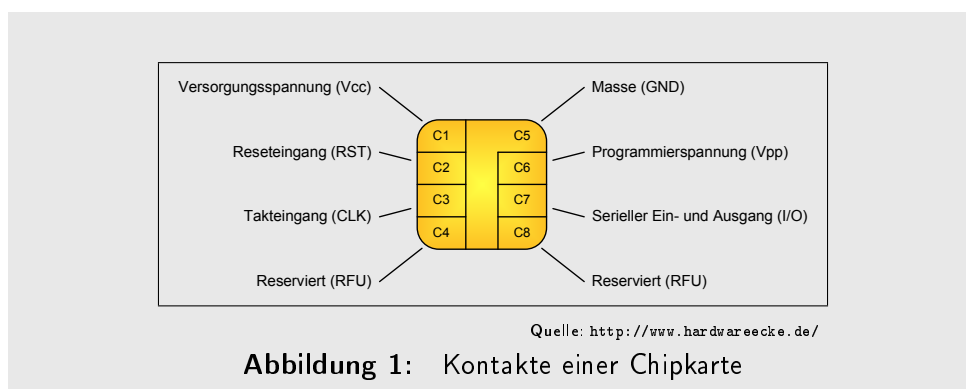


Abbildung 22: Ergebnis der Verwendung des Makros `\SinglePicture`

PlainFigure

Eine andere Variante zur Erzeugung von Abbildungen innerhalb einer wissenschaftlichen Arbeit stellt die Umgebung `PlainFigure` dar. Diese Umgebung bietet dem Anwender beispielsweise die Möglichkeit, Freitext oder auch direkt durch \LaTeX erzeugte Abbildungen in seiner Abhandlung zu integrieren. Dabei erfolgt die Übergabe der einzelnen Parameter analog zur Parameterübergabe des Makros `\SinglePicture`. Hiervon ausgenommen ist natürlich die Verwendung eines Dateinamens, da die Umgebung nicht vorrangig zum Laden externer Bilddateien gedacht ist. Möglich wäre es dennoch.

Da alle Parameter der Umgebung `PlainFigure`, mit Ausnahme der Angabe des Dateinamens, sowohl in ihrer Reihenfolge als auch in den Standardwerten der beiden optionalen Parameter identisch zum Makro `\SinglePicture` sind, soll an dieser Stelle auf deren ausführliche Erläuterung verzichtet werden. Stattdessen erscheint eine beispielhafte Verwendung der Umgebung `PlainFigure` besser geeignet zu sein.

Beispiel zur Verwendung der Umgebung PlainFigure

Angenommen, ein Autor möchte einem Leser den Graphen einer quadratischen Funktion präsentieren, dann kann hierfür die Umgebung `PlainFigure`, wie in Quelltext 16 gezeigt, verwendet werden.

Quelltext 16: Beispiel zur Verwendung der Umgebung `PlainFigure`

```
\begin{PlainFigure}[0.6]% 60% der Textbreite, ohne Rahmen
    {Parabel der quadratischen Funktion}% Beschriftung
    {fig:Parabel}% Referenzlabel
    {Quelle: \url{http://www.matheass.de/}}% Quelle
\centering\setlength{\unitlength}{10mm}
\begin{picture}(4.3,3.6)(-1.92,-0.35)
% X-Achse
\put(-2,0){\vector(1,0){4.4}}\put(2.2,-.3){$x$}
% Y-Achse
\put(0,-.4){\vector(0,1){3.6}}\put(-.18,3.08){\makebox(0,0){$y$}}
% Kurve zeichnen
\linethickness{.3mm}
\qbezier(0.0,0.0)(1.2384,0.0)(2.0,2.7622)
\qbezier(0.0,0.0)(-1.2384,0.0)(-2.0,2.7622)
\end{picture}
\end{PlainFigure}
```

Nach Erstellung des Dokuments, das den Code aus Quelltext 16 enthält, stellt sich das Ergebnis wie in Abbildung 23 gezeigt dar.

\DoublePicture und \DoubleSubPicture

In wissenschaftlichen Abhandlungen kommt es zuweilen vor, dass Abbildungen zum Beispiel zum Vergleich gegenübergestellt werden müssen. Zur Einbindung solcher Doppelabbildungen kann der Endanwender die Makros `\DoublePicture` und `\DoubleSubPicture` miteinander kombinieren. Doch bevor tiefer

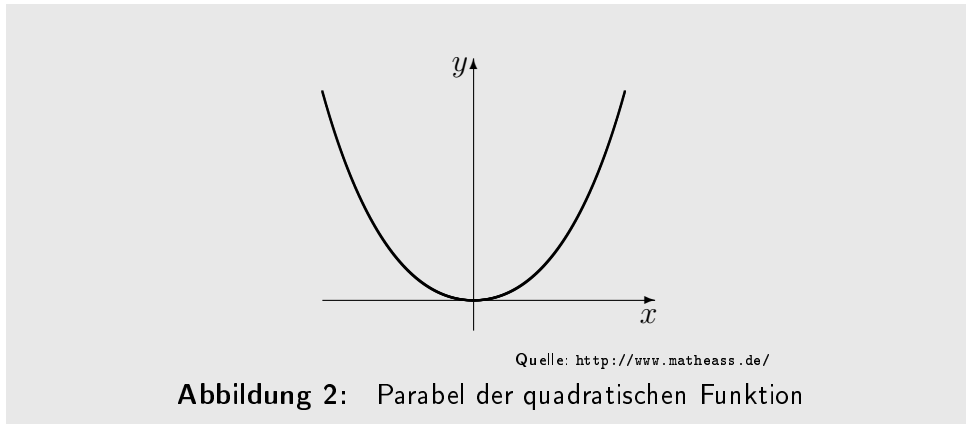


Abbildung 2: Parabel der quadratischen Funktion

Abbildung 23: Ergebnis der Verwendung der Umgebung PlainFigure

in die Materie der Verwendung dieser beiden Makros eingestiegen werden kann, sind noch einige allgemeine Worte zum Hintergrund der durchgeführten Implementierung unerlässlich.

Summiert man alle relevanten Werte, die zur Beschreibung einer Abbildung mit zwei Unterabbildungen vonnöten sind, dann kommt man zu dem Ergebnis, dass hierfür insgesamt zehn Variablen erforderlich wären. Und hierbei wurden nur die wichtigsten Parameter, bestehend aus

- drei Beschriftungen,
- drei Referenzbezeichner,
- zwei Bilddateien und
- zwei Quellenangaben

berücksichtigt. In diesem Zusammenhang sei daran erinnert, dass L^AT_EX-Makros über maximal neun Parameter verfügen dürfen.

Wie bereits auf Seite 33 angesprochen, kann durch einen Trick die Limitierung der Anzahl der Parameter umgangen werden. Jedoch ist es durchaus fraglich, ob eine derartige Umgehung hilfreich für den Endanwender sei. Denn würde man als Beispiel ein einzelnes Makro mit den oben genannten zehn verpflichtenden Argumenten ausstatten, dann hätte dies sicherlich eine verstärkte Unübersichtlichkeit zur Folge, die unmöglich im Interesse des Anwenders liegen kann!

Daher wurde während der Umsetzung zur Unterstützung von Abbildungen mit Unterabbildungen entschieden, die zugehörige Implementierung in zwei

getrennten Makros durchzuführen. Auch wenn die unabhängige Betrachtung der beiden Makros `\DoublePicture` und `\DoubleSubPicture` möglich wäre, gehören sie dennoch zusammen. Denn jedes Makro für sich betrachtet, bietet einem Endanwender wenig bis keinen Nutzen.

Das Makro `\DoublePicture` bildet durch seine vier obligatorischen Parameter $\{\langle\textit{Hauptbeschriftung}\rangle\}$, $\{\langle\textit{Hauptreferenzlabel}\rangle\}$, $\{\langle\textit{linke Abbildung}\rangle\}$ und $\{\langle\textit{rechte Abbildung}\rangle\}$ den notwendigen Rahmen zur Integration von Doppelabbildungen in wissenschaftlichen Abhandlungen. Es verfügt ebenfalls über ein optionales Argument `[box]`, welches die Erzeugung eines separaten Rahmens um jede der beiden Abbildungen erlaubt.

Abweichend von den weiter oben beschriebenen Abbildungskommandos verfügt das Makro `\DoublePicture` über keinen Parameter, der eine Größenanpassung der beiden dargestellten Bilder gestatten würde. Denn beide in den Parametern $\{\langle\textit{linke Abbildung}\rangle\}$ und $\{\langle\textit{rechte Abbildung}\rangle\}$ enthaltenen Unterabbildungen werden automatisch skaliert. Dabei erfolgt die Größenanpassung der gesamten Abbildungsbreite auf 90 % und die Skalierung beider Abbildungshälften auf 42 % der vorhandenen Breite des umgebenden Textes. Die sich hieraus ergebende Differenz bildet den Abbildungszwischenraum.

Die Definition der linken beziehungsweise rechten Unterabbildung obliegt dem Makro `\DoubleSubPicture`, das über die vier obligatorischen Parameter $\{\langle\textit{Unterbeschriftung}\rangle\}$, $\{\langle\textit{Unterreferenzlabel}\rangle\}$, $\{\langle\textit{Unterbilddatei}\rangle\}$ und $\{\langle\textit{Unterquellenangabe}\rangle\}$ verfügt. Dieses Makro stellt dem Anwender jedoch keine optionalen Argumente zur Verfügung!

Mit Rücksicht auf den Leser dieser Abhandlung sollte hier nun besser auf die umfassende und detaillierte Behandlung jedes einzelnen Pflichtparameters verzichtet werden, da deren grundsätzliche Bedeutung identisch mit den analogen Argumenten der Kommandos `\SinglePicture` und `PlainFigure` sind.

Des Weiteren sollten die hier gewählten Benennungen den Zweck des jeweiligen Argumentes bereits erkennen lassen. Zudem erscheint die beispielhafte Verwendung der Makros `\DoublePicture` und `\DoubleSubPicture` als ein geeigneteres Mittel, um das Zusammenspiel der hier vorgestellten Makros zu demonstrieren.

Beispiel zur Verwendung von `\DoublePicture` und `\DoubleSubPicture`

Man stelle sich vor, der Autor einer wissenschaftlichen Abhandlung möchte seinem Lesern anhand einer Gegenüberstellung den Unterschied zwischen den verschiedenen Chipkartenarten verdeutlichen. Zu diesem Zweck würde sich die Kombination der `\DoublePicture` und `\DoubleSubPicture` hervorragend eignen, da hierfür nur das in Quelltext 17 gezeigte Codefragment erforderlich wird.

Quelltext 17: Beispiel zur Verwendung des Makros `\DoublePicture`

```

\DoublePicture[box]% inklusive Rahmen
{Arten von Chipkarten}% Hauptbeschriftung
{fig:ChipkartenArten}% Hauptreferenzlabel
{\DoubleSubPicture% Unterabbildung links
  {Aufbau einer Speicherkarte}% Beschriftung
  {fig:Speicherkarte}% Referenzlabel
  {Speicherkarte.pdf}% Bilddateiname
  {Quelle: \url{http://www.hardware-ecke.de/}}}%Quelle
{\DoubleSubPicture% Unterabbildung rechts
  {Aufbau einer Prozessorkarte}% Beschriftung
  {fig:Prozessorkarte}% Referenzlabel
  {Prozessorkarte.pdf}% Bilddateiname
  {Quelle: \url{http://www.hardware-ecke.de/}}}%Quelle

```

Wurde das Dokument, welches den Code aus Quelltext 17 verwendet, kompiliert, dann enthält die Ausgabe das in Abbildung 24 dargestellte Ergebnis.

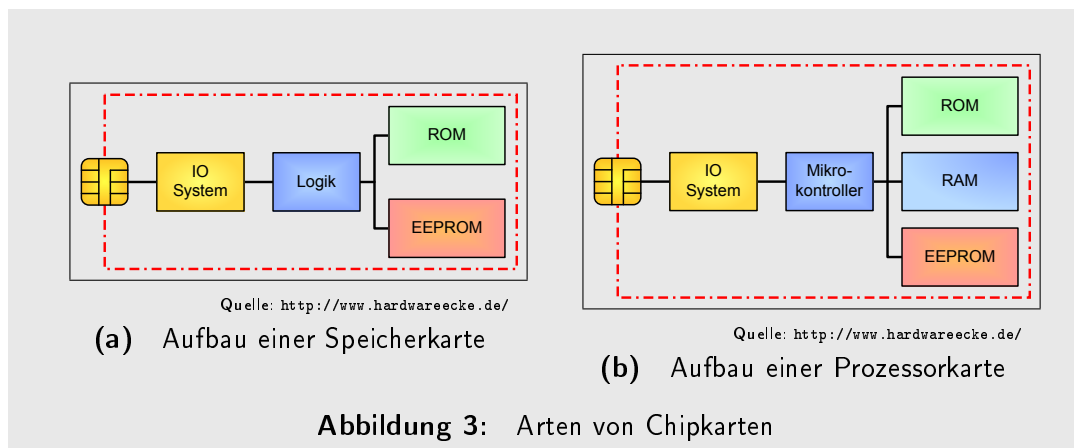


Abbildung 24: Ergebnis der Verwendung des Makros `\DoublePicture`

Table

Hin und wieder kommt es vor, dass Tabellen einer wissenschaftlichen Arbeit nicht standardgerecht formatiert werden. Daher stellt die Datei `fomfod.sty` dem Anwender die Umgebung `Table` zur Verfügung, mit deren Hilfe eine den allgemein anerkannten Empfehlungen genügende Formatierung der Tabellen vorgenommen werden kann.

Wie sich der geeignete Leser eventuell vorstellen kann, ist eine Abstraktion der vielfältigen Varianten existierender Tabellen eine nahezu unlösbare Aufgabe. Aus diesem Grund kann und muss sich der Aufgabenbereich einer Umgebung zur allgemeinen Tabellenkonfiguration auf grundlegende Aktionen beschränken. Hierzu zählen vor allem die standardkonforme Positionierung der Tabellenbeschriftung sowie eine den Richtlinien entsprechende Platzierung der Quellenangabe, falls diese notwendig wird.

Diese Beschränkungen spiegeln sich auch in der Anzahl der benötigten Argumente wider. Denn zur Konfiguration der Umgebung `Table` werden den Anwendern nur die obligatorischen Parameter $\{\langle Beschriftung \rangle\}$, $\{\langle Referenzlabel \rangle\}$ und $\{\langle Quellenangabe \rangle\}$ bereitgestellt. Eine Angabe optionaler Argumente ist hier nicht erforderlich.

Auch in diesem Teilabschnitt sollte besser auf die ausführliche Beschreibung der einzelnen Parameter verzichtet und stattdessen lieber ein Beispiel zur Darstellung der Zusammenhänge herangezogen werden.

Beispiel zur Verwendung der Umgebung `Table`

Das Beispiel in Quelltext 18 soll zeigen, mit welcher Leichtigkeit die standardkonforme Formatierung der Tabellen einer wissenschaftlichen Abhandlung gelingen kann, wenn hierfür die Umgebung `Table` zum Einsatz kommt.

Quelltext 18: Beispiel zur Verwendung der Umgebung `Table`

```
\begin{Table}% Umgebungsbeginn
  {Übertragungsprotokolle für Chipkarten}% Beschriftung
  {tab:ChipProto}% ...
  ...Referenzlabel
  {Quelle: \url{http://www.hardware-ecke.de/}}% Quelle
\begin{tabular}{ll}\hline
  \textbf{Protokoll} & \textbf{Beschreibung}\\\hline
  T = 0 & Asynchron, half-duplex, Byte orientiert\\
  T = 1 & Asynchron, half-duplex, Block orientiert\\
  T = 2 & Asynchron, full-duplex, Block orientiert\\
  T = 3 & Full duplex, wird momentan nicht verwendet\\
  T = 4 & Asynchron, half-duplex, Byte orientiert\\
  T = 14 & für nationale Funktionen\\\hline
\end{tabular}
\end{Table}% Umgebungsende
```

Das Ausgabeergebnis des in Quelltext 18 enthaltenen Codefragments stellt sich nach einer Kompilierung des entsprechenden Dokuments, wie in Abbildung 25 gezeigt, dar.

`CodeListing` und `\CodeListingFile`

Den letzten Zweig der durch `fomfod` bereitgestellten Anweisungen zur Formatierung von Gleitobjekten stellen die Kommandos zur Integration von Quellcodelistings dar. Da sowohl die Umgebung `CodeListing` als auch das Makro `\CodeListingFile` die ihnen übergebenen Parameter auf die gleiche Weise verarbeiten, empfiehlt sich auch deren gemeinsame Behandlung. Denn abgesehen von der abweichenden Anwendung von Umgebungen und Makros besteht der einzige Unterschied dieser Befehle in der Bekanntgabe des angezeigten Inhalts.

Die Umgebung `CodeListing` unterstützt zur Konfiguration einer Quelltextüberschrift den obligatorischen Parameter $\{\langle Beschriftung \rangle\}$ und zur Definition

Tabelle 1: Übertragungsprotokolle für Chipkarten

| Protokoll | Beschreibung |
|-----------|--|
| T = 0 | Asynchron, half-duplex, Byte orientiert |
| T = 1 | Asynchron, half-duplex, Block orientiert |
| T = 2 | Asynchron, full-duplex, Block orientiert |
| T = 3 | Full duplex, wird momentan nicht verwendet |
| T = 4 | Asynchron, half-duplex, Byte orientiert |
| T = 14 | für nationale Funktionen |

Quelle: <http://www.hardware-ecke.de/>**Abbildung 25:** Ergebnis der Verwendung der Umgebung `Table`

eines Bezeichners für Querverweise das Pflichtargument $\{\langle\textit{Referenzlabel}\rangle\}$. Eine Anpassung der linken und rechten Quelltexteinrückung ist ebenfalls möglich. Dies kann dann durch den freiwilligen Parameter $[\langle\textit{Einzug}\rangle]$ bewirkt werden, dessen Standardwert auf sieben Punkte voreingestellt wird.

Naturgemäß dienen L^AT_EX-Umgebungen zur Kapselung von Textpassagen, die in der Regel noch einer speziellen Formatierung unterzogen werden. Solche Textstellen sind vom Grundsatz her eher kürzer gefasst als der umliegende Fließtext. Dieses Grundkonzept lässt auch den bevorzugten Anwendungsfall der hier vorgestellten Umgebung `CodeListing` erkennen. Denn diese Umgebung ist primär zur Erzeugung relativ kurzer Quelltextpassagen gedacht. Die Formatierung längerer Quellcodes ist dennoch möglich.

Auf dem Fachgebiet der Informatik erstellte Arbeiten enthalten nicht selten den Quellcode eines bearbeiteten Softwareprojektes. Die Integration dieser zumeist externen Dateien ist Aufgabe des Makros `\CodeListingFile`. Dieses Makro verfügt, neben dem optionalen Parameter $[\langle\textit{Einzug}\rangle]$ und den obligatorischen Argumenten $\{\langle\textit{Beschriftung}\rangle\}$ und $\{\langle\textit{Referenzlabel}\rangle\}$, zusätzlich auch über den Pflichtparameter $\{\langle\textit{Quellcodedatei}\rangle\}$, der zur Bekanntgabe des Namens der einzubindenden Quelltextdatei verwendet wird.

Ein Beispiel soll an dieser Stelle helfen, die marginalen Unterschiede der beiden Kommandos `CodeListing` und `\CodeListingFile` deutlicher herauszuheben und ebenso die überwiegenden Gemeinsamkeiten besser darzustellen.

Beispiel zur Verwendung von `CodeListing` und `\CodeListingFile`

Angenommen, ein Autor möchte den Programmcode des allseits beliebten und oft zitierten Programms „hello, world!“ in seiner Ausarbeitung einbeziehen, da hieran beispielsweise die Grundzüge der Programmiersprache „C“ erläutert werden sollen, dann ist hierfür die Umgebung `CodeListing` bestens geeignet. Quelltext 19 zeigt das zur Umsetzung dieses Szenarios nötige Codefragment.

Quelltext 19: Beispiel zur Verwendung der Umgebung CodeListing

```

\begin{CodeListing}[35mm]%           Einzug (links/rechts)
{Das Programm 'hello, world!'}%      Beschriftung
{lst:hello}%                          Referenzlabel
#include <stdio.h>
int main( void ) {
    printf( "hello, world!\n" );
    return 0;
}
\end{CodeListing}

```

Eventuell stellt der oben angenommene Autor fest, dass andauernde Programmänderungen eine ständige Aktualisierung der Ausarbeitung zur Folge haben. Daher beschließt selbiger Autor, die hiervon betroffenen Dateien besser in das Dokument zu importieren. Hierfür ist das Makro `\CodeListingFile` hervorragend geeignet. Quelltext 20 veranschaulicht das zum Import der Beispieldatei `hello.c` notwendige Codefragment.

Quelltext 20: Beispiel zur Verwendung des Makros `\CodeListingFile`

```

\CodeListingFile[35mm]%           Einzug (links/rechts)
{Das Programm 'hello, world!'}%      Beschriftung
{lst:hello}%                          Referenzlabel
{hello.c}%                            Quellcodedatei

```

Unter der Prämisse, dass der Inhalt der in Quelltext 20 verwendeten Beispieldatei mit dem in Quelltext 19 verwendeten Programmcode übereinstimmt, würde sich nach einer Erstellung des entsprechenden Dokuments die in Abbildung 26 dargestellte Ausgabe ergeben.

Listing 1: Das Programm 'hello, world!'

```

#include <stdio.h>
int main( void ) {
    printf( "hello, world!\n" );
    return 0;
}

```

Abbildung 26: Ergebnisse der Verwendung von `CodeListing` und `\CodeListingFile`

3.2.6 Vermischtes

Ein Sammelbecken für weitere nützliche und hilfreiche Makros stellt die *FOM Miscellaneous Definition* dar, die in der Datei `fomisc.sty` implementiert ist.

Hierin lassen sich alle Makros finden, die vom Sinnzusammenhang keiner anderen Datei des Pakets `fomsgdt` zugeordnet werden konnten.

Alle bereitgestellten Makros sind so definiert, dass hierfür die Angabe von Dateiladeoptionen nicht erforderlich wird und somit deren Verwendung unabhängig von etwaigen Randbedingungen erfolgen kann.

`\Raise`

Grundsätzlich kann hochgestellter Text mit Hilfe des mathematischen Satzes erzeugt werden, wie das Beispiel $E = m \cdot c^2$ demonstriert. Jedoch führt dessen Verwendung nicht immer zum erwarteten Ergebnis, da beispielsweise hochgestellte Ziffern und hochgestellte Buchstaben unterschiedlich dargestellt werden. Diese Unterschiede könnten auf den Leser einer wissenschaftlichen Arbeit durchaus irritierend wirken. Daher wird zur verbesserten Darstellung hochgestellter Texte die Verwendung des Makros `\Raise` empfohlen.

Zur Positionierung des im obligatorischen Parameter `{<hoch>}` übergebenen Textes wird dieser im ersten Schritt um 85 Prozent der Höhe eines „x“ nach oben verschoben und anschließend auf 70 Prozent der Größe des umliegenden Textes geschrumpft. Dadurch wird sichergestellt, dass bei einer Verwendung der L^AT_EX-eigenen Makros zur Schriftgrößenänderung der durch `\Raise` erzeugte Hochtext ebenfalls seine Größe anpasst, wie das Beispiel in Quelltext 21 eindrucksvoll demonstriert.

Quelltext 21: Beispiele zur Verwendung des Makros `\Raise`

```
{\scriptsize Normal\Raise{hoch}}\
{\footnotesize Normal\Raise{hoch}}\
{\small Normal\Raise{hoch}}\
{\normalsize Normal\Raise{hoch}}\
{\large Normal\Raise{hoch}}\
{\Large Normal\Raise{hoch}}\
{\LARGE Normal\Raise{hoch}}\
{\huge Normal\Raise{hoch}}\
```

Die gedruckte Version des in Quelltext 21 verwendeten Codefragments stellt sich dann wie in Abbildung 27 gezeigt dar.

`\TodayShort`

Normalerweise wird das aktuelle Datum innerhalb eines L^AT_EX-Dokuments durch Verwendung des Makros `\today` eingefügt. Dieses Makro birgt leider den Nachteil, dass der erzeugte Datumstext den ausgeschriebenen, also wörtlich wiedergegebenen Monatsnamen enthält. Das kann unter Umständen hinderlich sein, wenn als Beispiel die Monatsbezeichnung in Form von Ziffern bevorzugt wird. Aus diesem Grund stellt fomisc das Makro `\TodayShort` zur Verfügung.

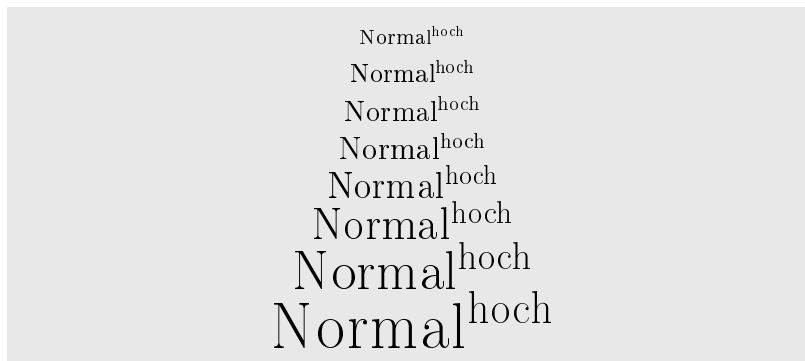


Abbildung 27: Ergebnis der Verwendung des Makros `\Raise`

Dieses Makro verfügt über zwei optionale Parameter, die zur Modifikation des ausgegebenen Datumstextes verwendet werden können. Hierbei wird der erste optionale Parameter zur grundsätzlichen Vorauswahl des Formates verwendet und der zweite freiwillige Parameter dient der Definition des zu verwendenden Trennzeichens. Sowohl Datumsformat als auch Trennzeichen selektieren ihre Anzeigeinformationen anhand verschiedener Schlüsselwörter, die nun kurz vorgestellt werden sollen.

Erlaubte Schlüsselwörter des ersten Parameters [*Format*]

Wird das Makro `\TodayShort` mit `[din]` als erstem Parameter aufgerufen, dann wird zur Ausgabe des aktuellen Datums das Format *dd.mm.yyyy* verwendet. Diese Formatierung entspricht den Empfehlungen der Norm DIN 5008 und ist auch der Standardwert, wenn keine Angabe des ersten Parameters erfolgt.

Wird hingegen das Makro `\TodayShort` mit `[iso]` als erstem Parameter aufgerufen, dann wird zur Ausgabe des aktuellen Datums das Format *yyyy-mm-dd* verwendet. Diese Formatierung entspricht dann den Empfehlungen der Normen ISO 8601:2004 und EN 28601.

Erlaubte Schlüsselwörter des zweiten Parameters [*Trennzeichen*]

Wird `[def]` dem Makro `\TodayShort` als zweiter Parameter übergeben, dann bewirkt dies die Verwendung des Standardtrennzeichens für Datumsangaben. Dabei erfolgt die Auswahl des Trennzeichens in Abhängigkeit von der Formatangabe des ersten Parameters und ist

- ein Punkt für den Fall einer DIN-gerechten Formatierung und
- ein Gedankenstrich im Falle der ISO-konformen Formatierung!

Wird dem Makro `\TodayShort` hingegen ein `[⟨Freitext⟩]` als zweiter Parameter übergeben, dann wird das angegebene Zeichen beziehungsweise die verwendete Zeichenkette zur Trennung der einzelnen Bestandteile des Datums verwendet. Die Angabe eines „leeren“ Zeichens ist ebenfalls möglich.

Die verschiedenen Möglichkeiten der Kombination der Parameter `[⟨Format⟩]` und `[⟨Trennzeichen⟩]` zeigen die Beispiel in Tabelle 3.

Weitere Makros der Datei `fomisc.sty`

Autoren wissenschaftlicher Arbeiten beziehen sich häufig auf Produkte und Marken von Herstellern, die mit Vermerken wie „Registered“, „Copyright“ oder auch „Trademark“ ausgestattet sind. Hierfür werden in der Regel spezielle Symbole verwendet, die von `fomisc` ebenfalls bereitgestellt werden. Beispiele zur Verwendung der Makros `\Registered`, `\Copyright` und `\Trademark` sind

- Google™,
- Microsoft® und
- Copyright © 2007.

Das Makro `\Axe1`, das eine Ausgabe der Form Ax_E1 erzeugt, soll den Anwendern des Pakets `fomsdt` die Möglichkeiten des \LaTeX -Systems näher bringen. Es ist als Beispiel zur Formatierung von Texten gedacht, deren einzelne Buchstaben „aus der Reihe tanzen“. Andere Beispiele derartiger Texte sind die Eigennamen \TeX und \LaTeX . Hierdurch soll auch erreicht werden, dass die Anwender des Pakets `fomsdt` inspiriert werden, eigene Versuche zur Textgestaltung zu unternehmen und spielerisch mit dem \LaTeX -System umzugehen.

3.2.7 Indexkonfiguration

Abweichend zu allen zuvor behandelten Paketelementen verfügt die *FOM Index Style Definition*, implementiert in der Datei `fomidx.ist`, nicht über Makros im klassischen Sinne. Denn sie dient, wie der Dateiname bereits vermuten lässt, ausschließlich der Konfiguration des Layouts des Indexes. Daher wird sich dieser Abschnitt darauf beschränken, einen Überblick über die vorgenommenen Layoutanpassungen des Indexes zu verschaffen.

Basis für die Konfiguration des Indexes bildet die Datei `din.ist`, deren Aufgabe in erster Linie darin besteht, die DIN-gerechte Sortierreihenfolge für Wörter mit deutschen Umlauten festzulegen. Jedoch stellte sich während der Umsetzung heraus, dass diese Sortierung nicht in allen Fällen einwandfrei funktioniert! Somit wurde die Programmierung eines Patches erforderlich, bei der eine

Tabelle 3: Mögliche Verwendungsformen des Makros `\TodayShort`

| Aufrufparameter | Ausgabe | Bemerkung |
|------------------------------------|------------|-------------------------------------|
| <code>\TodayShort</code> | 01.08.2007 | DIN-gerecht, Standardtrennung |
| <code>\TodayShort[din]</code> | 01.08.2007 | DIN-gerecht, Standardtrennung |
| <code>\TodayShort[din][def]</code> | 01.08.2007 | DIN-gerecht, Standardtrennung |
| <code>\TodayShort[din][]</code> | 01082007 | DIN-gerecht, ohne Trennung |
| <code>\TodayShort[din][/]</code> | 01/08/2007 | DIN-gerecht, amerikanische Trennung |
| <code>\TodayShort[iso]</code> | 2007-08-01 | ISO-konform, Standardtrennung |
| <code>\TodayShort[iso][def]</code> | 2007-08-01 | ISO-konform, Standardtrennung |
| <code>\TodayShort[iso][]</code> | 20070801 | ISO-konform, ohne Trennung |
| <code>\TodayShort[iso][/]</code> | 2007/08/01 | ISO-konform, amerikanische Trennung |

Korrektur des L^AT_EX-Makros `\@wrindex` erfolgte. Da die Datei `fomidx.ist`, wie eingangs bereits erwähnt, keine Makros enthält, wurde die durchgeführte Fehlerbehebung mit in der Datei `fomdes.sty` aufgenommen. Dennoch gehört dieser Patch zur Datei `fomidx.ist` und ist somit der Indexkonfiguration zuzurechnen.

Wie Abbildung 28 erkennen lässt, wurden am vordefinierten Aussehen des Stichwortverzeichnisses verschiedentliche Einstellungen und Anpassungen vorgenommen. Dabei orientieren sich alle Änderungen an einer im deutschen Sprachraum oft verwendeten gestalterischen Ausprägung des Indexlayouts, auf die nun näher eingegangen werden soll.

Gruppierung nach Anfangsbuchstaben

Als erste maßgebliche Änderung war eine Gliederung der einzelnen Einträge des Indexes erforderlich. Hierfür wurde als Überschrift der Anfangsbuchstabe der jeweiligen Gruppe gewählt und so formatiert, dass deren Zugehörigkeit deutlich erkennbar wird. Im Einzelnen besteht die Formatierung der Gruppenüberschrift aus einem links ausgerichteten Großbuchstaben, der fett und ohne Serifen ausgegeben wird. Weiterhin wird eine größere Distanz zur vorhergehenden Gruppe eingestellt, als das für den Abstand zu den eigenen Einträgen der Fall ist.

Abtrennung der Seitenangabe

Im nächsten Schritt wurde die Seitenreferenz eines jeden Indexeintrages rechtsbündig ausgerichtet und der hierdurch entstandene Zwischenraum mit Punkten aufgefüllt. Außerdem wurde die Folgeseitenreferenzierung hinter der Seitenzahl

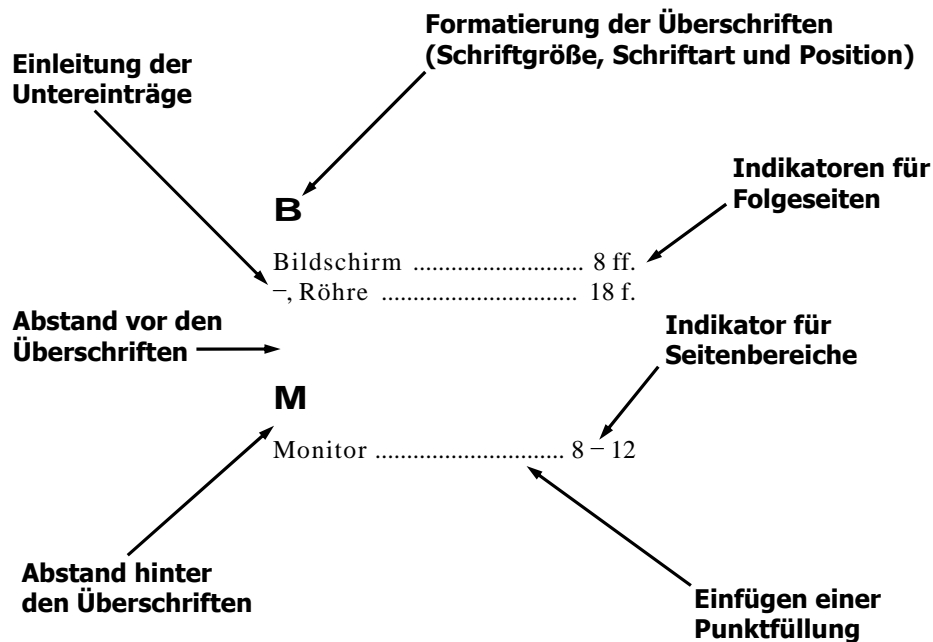


Abbildung 28: Anpassungen und Änderungen des Layouts des Indexes

an die im deutschsprachigen Raum typischerweise zu diesem Zweck verwendeten Abkürzungen angepasst. Dabei besteht für die jeweiligen Seitenzahlformatierungen der hier dargelegte Zusammenhang.

- ... $\langle n \rangle$ f. Indexeinträge, die auf den Seiten $\langle n \rangle$ und $\langle n+1 \rangle$ vorkommen, wird die Abkürzung *f.* für „folgende“ angehängt.
- ... $\langle n \rangle$ ff. Indexeinträge, die auf den Seiten zwischen $\langle n \rangle$ und $\langle n+2 \rangle$ vorkommen, wird die Abkürzung *ff.* für „fortfolgende“ angehängt.
- ... $\langle n \rangle - \langle m \rangle$ Indexeinträge, die sich über mehr als drei Seiten erstrecken, verwenden einen Gedankenstrich, der durch zwei Leerzeichen eingefasst wird.

Einrückung der Untereinträge

Die Standardeinstellungen des Indexes rücken alle Untereinträge eines Haupteintrages um 20 Punkt und alle Untereinträge eines Untereintrages um insgesamt 30 Punkt ein [3]. Dieser Einzug vermittelt nicht selten den Eindruck einer gewissen Unübersichtlichkeit. Daher wurde die Einzugformatierung für

Untereinträge angepasst und etwas übersichtlicher gestaltet. Hierfür wird dann die erste Unterebene nicht mehr eingezogen und zu dessen Hervorhebung ein Gedankenstrich vorangestellt, der von einem Komma und einem Leerzeichen gefolgt wird. Die zweite Ebene hingegen wird nur noch um 15 Punkt eingezogen und ebenfalls durch einen Gedankenstrich, gefolgt von Komma und Leerzeichen, eingeleitet. Weitere Unterebenen sind nicht zu berücksichtigen, da sowohl \LaTeX als auch das Programm `makeindex` nur bis zu drei Ebenen im Index unterstützt [17, Seite 3].

Abschließend soll noch ein besonderes Merkmal der Datei `fomidx.ist` hervorgehoben werden. Es besteht die Möglichkeit, die Indexkonfiguration auch völlig losgelöst vom Paket `fomsdt` zu verwenden! Das bedeutet, dass die Anwender, die sich eventuell dazu entscheiden, das Paket `fomsdt` nicht zu verwenden, dennoch die Chance haben, den Index mit einem ansprechenderen Layout zu erzeugen. Der einzige Nachteil entsteht dadurch, dass in diesem Fall der oben angesprochene Patch nicht aktiv ist und somit alle mit einem Umlaut beginnenden Wörter im Abschnitt „Symbole“ des generierten Indexes einsortiert werden. Dies wäre dann weniger schön.

3.3 Benutzerdokumentation

Ein weiterer wichtiger Bestandteil der Implementierung des Pakets `fomsdt` war die Bereitstellung einer entsprechenden Benutzerdokumentation, auf die in diesem Abschnitt näher eingegangen werden soll.

Im Fokus des zu erstellenden Benutzerhandbuchs stand, dass es lediglich den Anspruch erheben sollte, dem Endanwender den Umgang mit dem Paket `fomsdt` zu erläutern. Daher erfolgte die Aufteilung beziehungsweise der Aufbau der einzelnen Handbuchkapitel auf Basis des pragmatischen Ansatzes: *Was interessiert einen Benutzer zu welchem Zeitpunkt.*

Somit wurde während des Verfassens der Paketbeschreibung besonderes Augenmerk auf das Vorhandensein von Anleitungen und Beispielen zur Verwendung der einzelnen Komponenten des Pakets gelegt, da diese die Autoren wissenschaftlicher Arbeiten am ehesten interessieren würden.

Im Gegenzug wurde auf eine \LaTeX -konforme Dokumentation aller Quellcodes verzichtet. Dies bedeutet aber nicht, dass auf die Quelltextkommentierung gänzlich verzichtet wurde. Denn für diejenigen Endbenutzer, die selbst Änderungen oder Anpassungen am Paket `fomsdt` vornehmen möchten, befindet sich eine vollständige Kommentierung des Quelltextes in der zur Auslieferung verwendeten Paketbasis.

Neben einer Beschreibung der Funktionsweisen der einzelnen bereitgestellten Kommandos ist eine ausführliche Anleitung zur Installation des Pakets `fomsdt` ebenfalls Bestandteil der Benutzerdokumentation.

Die Entscheidung, alle Anweisungen zur Installation in der Dokumentation zu integrieren, wurde unter anderem auf Grund der eingeschränkten Möglichkeiten zur Hervorhebung und Formatierung von Texten innerhalb von ASCII-Dateien getroffen.

Hieraus resultiert der durchaus als positiv zu bewertende Nebeneffekt, dass die Paketdokumentation durch den Endbenutzer erst erzeugt werden muss, bevor mit der eigentlichen Installation begonnen werden kann. Dies gewährleistet, zumindest gilt das für den Durchschnittsbenutzer, dass die Endbenutzerdokumentation auch auf dem Zielsystem vorhanden ist. Folglich kann der ebenfalls zur Dokumentation zu zählende Inhalt der Datei `readme.txt` bis auf eine Kurzanleitung zur Erstellung des Benutzerhandbuches reduziert werden.

3.4 Installationsmethode

Ungeklärt war ebenfalls die Frage nach der Methode der Installation des Pakets `fomsdt` auf dem Zielsystem. Diese Entscheidung fiel jedoch leicht, da in \LaTeX allgemeine Gepflogenheiten [23, 26] im Bezug auf eine Paketinstallation gelten.

Diese Empfehlungen raten an, alle Paketdateien zu einer einzelnen Datei, einer so genannten Documented- \TeX -Datei, zusammenzufassen. Weiterhin wird das Beifügen eines Installationsskripts sowie einer Installationsanweisung in Form einer Textdatei befürwortet. Somit besteht die Auslieferungsversion des Pakets `fomsdt` nur noch aus den Dateien

- `fomsdt.dtx` – das zusammengefasste Paket inklusive Dokumentation,
- `fomsdt.ins` – das zur Installation benötigte Skript und
- `readme.txt` – die Installationsanweisung für den Endbenutzer.

Die Erzeugung der Dateien `fomsdt.dtx` und `fomsdt.ins` erfolgte unter Verwendung des Pakets `makedtx`, das in der aktuellen Version 0.94b vorlag. Dieses Paket ermöglicht die automatisierte Erstellung der Documented- \TeX -Datei `fomsdt.dtx`. Weiterhin kann das benötigte Installationsskript `fomsdt.ins` ebenfalls automatisch durch dieses Paket generiert werden.

Hingegen wurde die Datei `readme.txt`, also die Anweisung zur Installation des Pakets `fomsdt`, händisch erzeugt und dem Distributionspaket durch einfaches

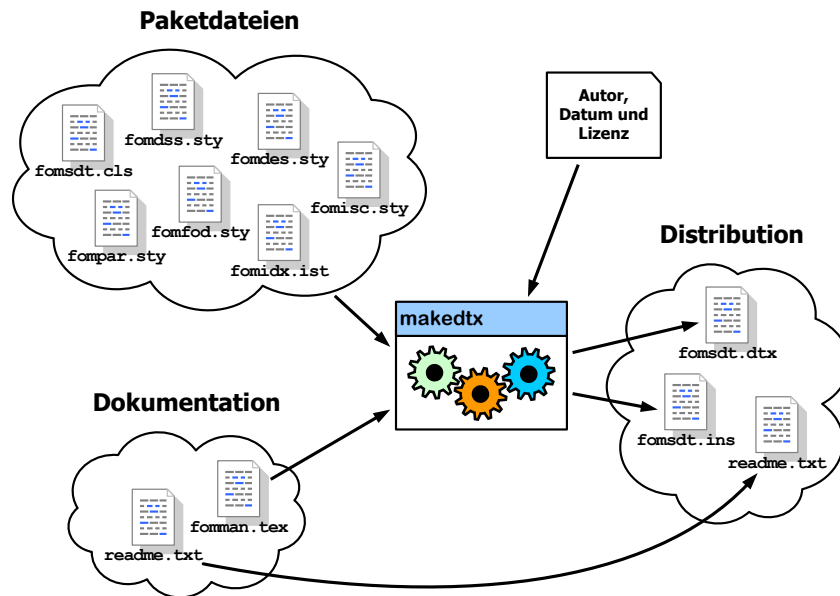


Abbildung 29: Erstellprozess der Paketdateien zur Auslieferung

Kopieren beigelegt. Diese Datei beinhaltet jedoch lediglich die Anweisungen zur Erzeugung des Benutzerhandbuchs.

Neben der Reduzierung der Anzahl der Dateien, die für eine Distribution zur Verfügung gestellt werden müssen, bietet diese Form der Paketbereitstellung einen weiteren großen Vorteil. Es ist nicht nur möglich, sondern auch sehr empfehlenswert, das Benutzerhandbuch in der Documented- \TeX -Datei, also in der zusammengefassten Paketdatei, unterzubringen.

Dieses für den reinen Endanwender des Pakets `fomsdt` bestimmte Handbuch, das in der in Abbildung 29 gezeigten Datei `fomman.tex` enthalten ist, wird ebenfalls automatisch in der Datei `fomsdt.dtx` eingefügt.

Die zuvor angeführten Vorteile bergen jedoch einen inhärenten und nicht zu vernachlässigenden Nachteil! Sowohl die Wahl einer eigenständigen Dokumentenklasse als auch die verwendete Art der Paketdistribution erhöhen die an den Endbenutzer gestellten Anforderungen. Denn zur Durchführung einer Paketinstallation sind nunmehr mehrere Schritte unumgänglich, bei denen es sich im Einzelnen um

- die Erzeugung der Paketdokumentation durch den Endbenutzer,
- das daran anschließende Entpacken des gesamten Pakets,

- das Kopieren der zuvor entpackten Dateien an ihren Bestimmungsort und
- die händische Aktualisierung des L^AT_EX-Grundsystems

handelt. Dennoch überwiegen die Vorteile, wie beispielsweise das Verbergen der inhärenten Komplexität, die durch die gewählte Paketstruktur erreicht werden.

3.5 Verträglichkeit

Nach Abschluss der Implementierungsphase wurde das Paket `fomsdt` durchgängig getestet. Hierbei wurde besonderer Wert auf die Verträglichkeit des Pakets zu den anderen Klassen und Paketen des L^AT_EX-Systems gelegt. Alle Fehler, die während der Tests auftraten, wurden behoben und daran anschließend der geänderte Code erneut getestet. Probleme, die nicht umgangen oder behoben werden konnten, werden in diesem Abschnitt dokumentiert.

Test mit anderen Dokumentenklassen

Während der durchgeführten Testphase wurden insbesondere die Wechselwirkungen des Pakets `fomsdt` zu den gängigen Dokumentenklassen `article`, `book`, `report`, `letter` und `beamer` untersucht. Hierbei konnte festgestellt werden, dass die einzelnen Paketkomponenten problemlos mit den meisten L^AT_EX-eigenen Dokumentenklassen zusammenarbeiten. Im Einzelnen konnten dabei die im Folgenden zusammengefassten Zusammenhänge ermittelt werden.

| | |
|----------------------|---|
| <code>article</code> | Die Dokumentenklasse <code>article</code> bildet die Basis der neu erstellten Klasse <code>fomsdt</code> . Somit wurden alle zugehörigen Komponenten im Umfeld dieser Basisklasse entwickelt. Ein unabhängig durchgeführter Test bestätigte ebenfalls ein problemloses Zusammenwirken der einzelnen Bestandteile des Pakets mit der Klasse <code>article</code> . |
| <code>book</code> | Die einzelnen Komponenten des Pakets <code>fomsdt</code> arbeiten problemlos mit der Dokumentenklasse <code>book</code> zusammen. |
| <code>report</code> | Ebenfalls problemlos arbeiten die einzelnen Komponenten des Pakets <code>fomsdt</code> mit der Dokumentenklasse <code>report</code> zusammen. |
| <code>letter</code> | Die Klasse <code>letter</code> verursachte zusammen mit der Datei <code>fomfod.sty</code> diverse leichte bis schwere Fehlermeldungen! Speziell trat hierbei der Fehler <code>LaTeX Error: Environment figure undefined</code> auf, der als durchaus ernst zunehmendes Problem zu betrachten ist. |

Dieser Fehler tritt auch dann auf, wenn die Klasse `letter` zusammen mit der Umgebung `figure` verwendet wird, ohne dass die Datei `fomfod.sty` darin involviert ist.

Auf Grund dieses Problems wurde entschieden, die Verwendung der Datei `fomfod.sty` zusammen mit der Klasse `letter` gänzlich zu unterdrücken und stattdessen den Fehler `Package fomfod Error: This package cannot be used with document class 'letter'!` auszugeben und daran anschließend einen Abbruch zu initiieren.

beamer Auch die Dokumentenklasse `beamer` verursachte diverse Fehlermeldungen, wenn diese zusammen mit der Datei `fomfod.sty` verwendet wurde. Als Ursache dieser Fehler konnte das Paket `subfig`, welches durch das Makro `\DoublePicture` benötigt wird, ausgemacht werden. Da die aufgetretenen Fehler umgangen werden konnten, ist die Verwendung der Klasse `beamer` zusammen mit der Datei `fomfod.sty` auch weiterhin möglich. Leider kann in diesem Fall das Makro `\DoublePicture` nicht verwendet werden!

Auch wenn grundsätzlich die Möglichkeit einer Verwendung der einzelnen Komponenten des Pakets `fomsdt` gemeinsam mit anderen L^AT_EX-Klassen besteht, wird dennoch von einer derartigen Anwendung abgeraten! Nicht zuletzt deshalb, weil in diesem Fall die allgemeinen Grundeinstellungen nicht verfügbar sind und die Benutzung der Titelseite, des Sperrvermerks und der ehrenwörtlichen Erklärung ausgeschlossen ist.

Test mit anderen Paketen

Bedingt durch den großen Umfang an verfügbaren L^AT_EX-Paketen kann sich der geneigte Leser sicherlich vorstellen, dass ein allumfassender Test des Pakets `fomsdt` in der Tat unrealistisch ist. Daher konnten die einzelnen Tests nur für ausgewählte Pakete durchgeführt werden! Tabelle 4 zeigt die Übersicht der Pakete, mit denen `fomsdt` problemlos verwendet werden kann.

Zusammenfassend kann an dieser Stelle festgehalten werden, dass alle durchgeführten Tests ein hohes Maß an Robustheit des Pakets `fomsdt` bestätigt haben. Es sind daher nur wenige Ausnahmen bekannt, in denen die Komponenten des Pakets `fomsdt` anders als vermutlich zu erwarten wäre, agieren. Diese Ausnahmen sollen Gegenstand der nun anschließenden Betrachtungen sein.

Formatierung der Folgeseiten im Index

Wie bereits aus Abbildung 28 ersichtlich wurde, werden im Stichwortverzeichnis die Angaben der Folgeseiten gemäß der Schemata $\langle n \rangle f.$, $\langle n \rangle ff.$ sowie $\langle n \rangle - \langle m \rangle$ formatiert. Diese Art der Darstellung birgt den Nachteil, dass die durch

Tabelle 4: Übersicht der getesteten Pakete

| | | | |
|----------|----------|----------|----------|
| a4wide | gloss | listings | rotating |
| caption | graphicx | makeidx | subfig |
| color | hyperref | multicol | tabularx |
| fancyhdr | ifpdf | natbib | textfit |
| fontenc | ifthen | ngerman | textpos |
| footmisc | inputenc | nomencl | ulem |

das Paket `hyperref` erzeugten Querverweise innerhalb eines PDF-Dokuments nicht wie erwartet angesprungen werden können.

In anderen Worten, ein Leser, der beispielsweise im Index einer Abhandlung den Verweis *11 ff.* findet, kann durch Klicken der angegebenen Zahl nicht wie erwartet auf *Seite 11* der Abhandlung wechseln. Stattdessen wird zum Anfang der Abhandlung gesprungen. Dieser Umstand ist sicherlich bedauerlich, kann aber auch nicht umgangen werden.

Skalierung von Abbildungen

Versucht der Anwender, die Größe einer mit Hilfe des Makros `\SinglePicture` oder unter Verwendung der Umgebung `PlainFigure` erstellten Abbildung zu manipulieren, dann kann durch die Angabe eines zu klein gewählten Skalierungsfaktors unter Umständen die Platzierung der Elemente Abbildung, Quellenangabe und Beschriftung durcheinander geraten. Dieser Effekt tritt unabhängig von der Textlänge für Quellenangabe und Beschriftung auf und ist somit nur vom Wert des Skalierungsfaktors der Abbildung abhängig. Die genauen Grenzen zeigt die Übersicht in Tabelle 5.

Tabelle 5: Empfohlene Skalierungsuntergrenzen für Abbildungen

| Kommando | Skalierungsfaktor | Bemerkung |
|----------------|-------------------|--------------------------|
| PlainFigure | 0,50 | mit umgebenden Rahmen |
| | 0,50 | ohne zusätzlichen Rahmen |
| \SinglePicture | 0,49 | mit umgebenden Rahmen |
| | 0,50 | ohne zusätzlichen Rahmen |

Dieser Effekt kann freilich vernachlässigt werden, da die Darstellung von zu klein skalierten Abbildungen ohnehin nicht empfehlenswert ist. Denn hierdurch wird die Lesbarkeit des Abbildungsinhaltes negativ beeinflusst.

Doppelbenennung des Makros `\PrintIndex`

Das in der Datei `fomdes.sty` definierte Makro `\PrintIndex` wird in der Datei `doc.sty` unter gleichem Namen definiert. Somit ist dieses Makro nach der Installation des Pakets `fomsdt` im \LaTeX -System an zwei unterschiedlichen Stellen vorhanden!

Durch diese Unverträglichkeit sind dennoch kaum Beeinträchtigungen zu erwarten, da die Datei `doc.sty` in der Regel nicht zur Erstellung wissenschaftlicher Arbeiten verwendet wird!

Kapitelüberschriften und die Option `nopagebreak`

Die Empfehlungen zur Erstellung wissenschaftlicher Arbeiten sagen aus, dass Hauptkapitel auf einer neuen Seite beginnen sollten. Kommt ein Anwender zum Beispiel auf die Idee, hierfür die Definition des Makros `\section`, wie in Quelltext 22 gezeigt, abzuändern, dann kann dies in den Dateien `fomdss.sty` und `fomdes.sty` zu einem Konflikt während der Verarbeitung der Option `nopagebreak` führen!

Quelltext 22: Konflikt mit der Option `nopagebreak`

```
% Speichere Originaldefinition
\let\tempsection\section
% Erzwingt Seitenvorschub vor jedem ...
...Hauptkapitel
\renewcommand\section{\clearpage\tempsection}
```

Dieser Konflikt tritt besonders dadurch hervor, dass trotz der Verwendung der Option `nopagebreak` jedes der erstellten Verzeichnisse auf einer neuen Seite beginnt. Das kann für den Benutzer durchaus ärgerlich sein, da dieses Problem nicht offensichtlich zu erkennen ist!

Falls ein Anwender dennoch nicht auf die Umdekoration des Makros `\section` verzichten will, dann zeigt Quelltext 23 die hierfür anwendbare Umgehung des hier beschriebenen Konflikts.

Quelltext 23: Umgehung des Konflikts der Option `nopagebreak`

```
\documentclass{fomsdt}
\usepackage[... ,nopagebreak]{fomdss}
\usepackage[... ,nopagebreak]{fomdes}
\begin{document}
% Verzeichnisse ausgeben
\PrintContents
\PrintAbbrev
\PrintFigures
\PrintTables
% Kapitelüberschriften anpassen.
\let\tempsection\section
\renewcommand\section{\clearpage\tempsection}
```

```
% Abhandlungsinhalt
\section{Einleitung}
\section{Grundlagen}
\section{...}
\section{Fazit}
% Kapitelüberschriften zurücksetzen.
\let\section\tempsection
% Verzeichnisse ausgeben
\PrintListings
\PrintGlossary{Glossar}
\PrintBibliography{Literatur}
\PrintIndex
\end{document}
```

Verwendung des Makros `\AtEndDocument`

Verwendet der Anwender das \LaTeX -Makro `\AtEndDocument` erst im Anschluss an eine Bekanntgabe der Dokumentenklasse via Makro `\documentclass`, dann tritt bei gleichzeitiger Aktivierung der Eidesstattlichen Versicherung der Effekt auf, dass selbige nicht mehr wie vorgeschrieben auf der letzten Abhandlungsseite zu finden ist!

Diese Phänomen lässt sich dadurch erklären, dass die einzelnen Dateien eines \LaTeX -Dokuments sequenziell eingelesen und verarbeitet werden! Da die Klassendatei `fomsdt.cls` das Makro `\AtEndDocument` bereits zur Erzeugung der Eidesstattlichen Versicherung verwendet, entsteht durch dessen erneute Anwendung ein Konflikt in der Reihenfolge der zuletzt ausgegebenen Seiten. Somit kann dieses Problem nur dann umgangen werden, wenn das Makro `\AtEndDocument` noch vor der Definition der Dokumentenklasse, also vor einem Aufruf des Makros `\documentclass`, erfolgt!

3.6 Kritikpunkte

Dieser Abschnitt enthält Informationen und Anmerkungen zu Punkten, die während der Umsetzung des Pakets `fomsdt` nicht berücksichtigt werden konnten, aber dennoch die Leser interessieren könnten. Primär soll hierin diskutiert werden, warum einige Implementierungsbestandteile von allgemeingültigen Standards abweichen oder auch warum auf eine Realisierung bestimmter Bereiche gänzlich verzichtet wurde.

Kommandobenennung

Die Erfahrung lehrt, dass die Namensfindung für Befehle und Funktionen, im Fall von \LaTeX also die Benennung der Makros und Umgebungen, eine durchaus anspruchsvolle Aufgabe darstellt. Aus diesem Grund verfügt zum Beispiel die Programmiersprache C++ über einen als „namespace“ bezeichneten Bereichsoperator, der vorrangig zur Gruppierung von Klassen und Methoden verwendet

wird. Das hehre Ziel derartigen Unterteilungen ist die Definition einzigartiger Kommandonamen, um mögliche Konflikte durch bereits vorhandene Bezeichner abzuschwächen beziehungsweise sie ganz zu unterdrücken.

Leider ist eine so ausgeprägte Gruppierung in L^AT_EX unmöglich. Daher behilft man sich in der Praxis durch Voranstellen eines Kürzels vor dem eigentlichen Bezeichner. Dies geschieht in der Hoffnung, dass niemand jemals auf die Idee kommt, denselben Bezeichner für seine eigenen Makros zu verwenden. Dieser Ansatz ist grundsätzlich auch sinnvoll, da er das Kernproblem eines Namenskonflikts deutlich reduziert!

In diesem Punkt bietet das Paket `fomsdt` Anlass zur Kritik, da keinem der veröffentlichten Makros und Umgebungen ein entsprechendes Präfix vorangestellt wurde! Dass dies ein unglücklicher Umstand ist, zeigt sich daran, dass bereits heute ein Namenskonflikt bekannt ist. Dieser Konflikt besteht in der Benennung des Makros `\PrintIndex`, weil die Datei `doc.sty` ebenfalls ein Makro unter diesem Namen definiert. Jedoch ist zu erwarten, dass diese Doppelbenennung nicht weiter problematisch ist, da die Datei `doc.sty` in der Regel nur im Zusammenhang mit der Erstellung einer Paketbeschreibung Verwendung findet. Somit ist die Wahrscheinlichkeit gering, dass ein Benutzer des Pakets `fomsdt` jemals auf dieses Problem stoßen wird.

Dessen ungeachtet wurde gleichwohl auf die Verwendung eines Kürzels verzichtet, weil zur Verbesserung der Lesbarkeit der einzelnen `fomsdt`-Bezeichner die Namensgebung durchgängig Groß- und Kleinbuchstaben kombiniert. Dies weicht von der in L^AT_EX üblichen Benennung ab und bietet somit zumindest einen minimalen Schutz vor möglichen Namensüberschneidungen.

Mehrsprachigkeit

Nach Durchführung einer intensiven Analyse aller notwendigen Aufwände sowie des sich daraus ergebenden Mehrwerts für den Endanwender wurde letztendlich der Entschluss gefasst, im Paket `fomsdt` auf die Unterstützung verschiedener Sprachen gänzlich zu verzichten! Einer der Gründe, die schlussendlich zu dieser Entscheidung beitrugen, ist der hierfür erforderliche Implementierungsaufwand.

Zudem richtet sich das Paket `fomsdt` in erster Linie an die Studierenden, die im deutschen Sprachraum ansässig sind. Daher wird davon ausgegangen, dass mit Hilfe des *Pakets zur Erstellung wissenschaftlicher Arbeiten* vorwiegend deutschsprachige Abhandlungen verfasst werden. Dies lässt dann auch einen überdurchschnittlichen Implementierungsaufwand als ungerechtfertigt erscheinen.

Sollte `fomsdt` dennoch einmal dazu verwendet werden, um eine Arbeit in einer von Deutsch verschiedenen Sprache zu erstellen, dann bleibt einem Anwender

immer noch die Möglichkeit, unter Verwendung der jeweiligen Makroparameter alle vordefinierten deutschsprachigen Texte anzupassen.

Quellcodedokumentation

In \LaTeX existieren gewisse Gepflogenheiten bezüglich der Dokumentation der Quelltexte. Diese allgemein anerkannten Regeln zielen grundsätzlich darauf ab, den verschiedenen Intensionen der Betrachtung eines Codefragments gerecht zu werden. Dabei wird in erster Linie darauf Wert gelegt, die Implementierung selbst einem Dritten anschaulich und verständlich darzubieten. Das bedeutet auch, dass eine typische \LaTeX -Quellcodedokumentation in der Regel die einzelnen Bestandteile der Umsetzung erläutert und einem Anwender somit wenig hilfreiche Informationen zur Verwendung des Codes bietet.

Die `fomsdt`-Dokumentation weicht von diesem Grundsatz ab! Der hierfür vornehmliche Grund bestand darin, den Endanwendern ein ausführliches und leicht verständliches Benutzerhandbuch mit möglichst vielen Beispielen sowie diversen Tipps und Tricks zur Verfügung zu stellen und ihnen nicht die Beschreibung aller Implementierungsdetails zuzumuten.

Wenn ein Anwender dennoch den Wunsch verspürt, sich die Funktionsweise der einzelnen Makros näher anschauen zu wollen, dann steht ihm immer noch die Option offen, in der Datei `fomsdt.dtx` den vollständig dokumentierten Quellcode einzusehen.

Mathematischer Satz

Das Setzen mathematischer Formeln bleibt innerhalb des Pakets `fomsdt` gänzlich unbeachtet! Denn gegen die entsprechende Unterstützung sprach vor allem eine nahezu unmöglich durchzuführende Abstraktion der mannigfaltigen Varianten mathematischer Zusammenhänge zu einer allgemeingültigen Struktur.

Auf der anderen Seite ist die Einbeziehung mathematischer Formeln in einem \LaTeX -Dokument derart einfach, dass jeglicher hierfür zu leistender Implementierungsaufwand zuzüglich der notwendigen Tests sowie einer Dokumentation des Verhaltens den daraus erwachsenden Nutzen deutlich übersteigen würde!

Zur Demonstration dieser Einfachheit der Verwendung mathematischer Formeln in \LaTeX -Dokumenten soll an dieser Stelle das Beispiel in Quelltext 24 dienlich sein. Dieses Beispiel zeigt unter anderem die wenigen notwendigen Handgriffe, um alle im Text verwendeten Formeln einheitlich zu definieren und diese dann auch in einem entsprechenden Verzeichnis am Dokumentenbeginn auszugeben.

Quelltext 24: Beispiel zur Demonstration der Einfachheit der Einbeziehung mathematischer Formeln in ein \LaTeX -Dokument

```

% Die Formelnummerierung am linken Rand verwenden!
\documentclass[leqno]{fomsdt}
% Präambel
\makeatletter
\def\equationshead{Formelverzeichnis}% Text der Überschrift
\def\listofequations{% Verzeichniserzeugung
  \section*{\equationshead}% Kapitelüberschrift ausgeben
  % sowie dessen Eintragung im Inhaltsverzeichnis vornehmen
  \addcontentsline{toc}{section}{\equationshead}%
  \@starttoc{loe}}
\makeatother

\newcommand{\formula}[3]{%      Kommando zur einfachen
  \begin{equation}%            Formelverwaltung definieren
    % Übergebenen Beschriftung im Formelverzeichnis eintragen
    \addcontentsline{loe}{subsection}{\theequation\quad#1}%
    \label{#2}#3%              Label definieren und Formel ausgeben
  \end{equation}}

% Text
\begin{document}
\listofequations% Formelverzeichnis ausgeben
% Formeln definieren
\formula{Die erste Binomische ...
  ...Formel}{binom1}{(a+b)^2=a^2+2ab+b^2}
\formula{Die Äquivalenz von Masse und Energie}{emc2}{E=m\cdot ...
  ...c^2}
% Formeln referenzieren
Die erste Binomische Formel in Formel~\ref{binom1} hat nichts mit
der Massen"~Energie"~Äquivalenz aus Formel~\ref{emc2} gemeinsam!
\end{document}

```

4 Schlussbetrachtung

Die vorliegende Diplomarbeit zeigt, dass das Verfassen einer wissenschaftlichen Abhandlung bisweilen recht komplizierte Züge annehmen kann. Und dies ist nicht zuletzt dadurch bedingt, dass von vielen Studierenden die Einhaltung diverser Regularien mit zum Teil widersinnigen und auch widersprüchlichen Vorschriften abverlangt wird. Das soll aber nicht heißen, dass Vorschriften und Regeln grundsätzlich schlecht oder gar falsch sind! Nur wäre ein „gesundes“ Maß an Auflagen, unter Beachtung allgemeiner Grundsätze zur Verbesserung der Lesbarkeit für alle Autoren von Seminar- und Abschlussarbeiten ebenso wünschenswert wie hilfreich. Jedoch konnten die vorherrschenden Verhältnisse im Rahmen dieser Abhandlung nur festgestellt werden.

Andererseits zeigt diese Abschlussarbeit auch, dass eine Verwendung von \LaTeX den Autoren während der Erstellung ihrer wissenschaftlichen Arbeiten in verschiedenen Bereichen helfen kann. Sei es beispielsweise durch die automatisierte Erzeugung von Verzeichnissen oder auch durch eine vorlagengestützte Erstellung des Titelblattes. Dennoch sollte nicht vergessen werden, dass trotz großer praktischer Erfahrung das Schreiben eines Dokuments in \LaTeX mehr Zeit in Anspruch nimmt, als dies vergleichsweise durch den Einsatz eines der gängigen Textverarbeitungssysteme der Fall ist. Gleichwohl rechtfertigt das ausgegebene Endergebnis diesen Mehraufwand.

Beide zuvor reflektierten Aspekte, also die Gestaltung der Schriftstücke in Bezug auf eine verbesserte Lesbarkeit sowie die vereinfachte Anwendung von \LaTeX , versucht das Paket `fomsdt` weitestgehend miteinander zu vereinen. Aus diesem Grund wurde auch auf eine Implementierung abstruser Regelungen so gut es ging verzichtet; einzige Ausnahme bildet die Option `toctotoc`, deren Aufgabe darin besteht, die Überschrift des Inhaltsverzeichnisses ebenfalls mit in das Inhaltsverzeichnis aufzunehmen. Zugleich bietet das *Paket zur Erstellung wissenschaftlicher Arbeiten* allen Anwendern genügend Flexibilität, um auch zukünftig das Anfertigen von Seminar- und Abschlussarbeiten zu ermöglichen.

Ein Benutzerhandbuch mit vielen Anwendungsbeispielen ist ebenfalls Bestandteil des Pakets `fomsdt`. Erst dieses Handbuch erlaubt dem Anwender die einfache Handhabung der einzelnen Komponenten des Pakets. Dabei ist der Aufbau der Dokumentation in weiten Teilen einem „Kochbuch“ nachempfunden und enthält daher vorwiegend die Erklärungen, *wie* ein bestimmtes Kommando verwendet werden sollte. Im Gegenzug wurde bewusst auf die funktionale Beschreibung der einzelnen Kommandos verzichtet, da ausladende Details der Implementierung in einem Handbuch mit Fokus auf eine Verwendung in der Regel eher hinderlich sind und somit nicht tatsächlich zur Übersichtlichkeit beitragen.

Bei der Wahl der Installationsmethode wurde besonderes Augenmerk darauf gelegt, dass sich das *Paket zur Erstellung wissenschaftlicher Arbeiten* auf möglichst vielen Plattformen gleichermaßen einrichten lässt. Diese Randbedingung hat nun aber zur Folge, dass die Integration des Pakets `fomsdt` in einer vorhandenen \LaTeX -Umgebung verschiedene Aktionen des Anwenders unumgänglich werden lässt. Sicherlich wäre eine Optimierung des Verfahrens zur Installation ebenso denkbar wie wünschenswert, jedoch stellt eine Rücksichtnahme auf alle Details der verfügbaren \LaTeX -Plattformen ein nicht zu vernachlässigendes Mehr an Aufwand dar. Somit bietet die aktuell unterstützte Installationsvariante ein ausgewogenes Maß an Komfort und Flexibilität.

Technologiebetrachtungen

Eine der herausragenden Eigenschaften von \LaTeX ist dessen Verfügbarkeit für diverse Betriebssysteme. Dieser Umstand ermöglicht allen \LaTeX -Anwendern eine plattformübergreifende Erstellung von Schriftsätzen. Dies stellt einen großen Vorteil dar, wenn sich beispielsweise ein Autor veranlasst sieht, eine Abhandlung auf verschiedenen Rechnern mit unterschiedlichen Betriebssystemen verfassen zu müssen. In diesem Fall können die Quellen des Dokuments dann einfach von einem System auf ein anderes kopiert und anschließend das Ausgabedokument problemlos erstellt werden. Das vom jeweiligen \LaTeX -Kern produzierte Ergebnis wird in beiden Fällen eine identische Brillanz aufweisen.

Eine ebenfalls bestechende Eigenschaft des \LaTeX -Systems ist die Tatsache, dass alle Dokumentquellen aus einfachen ASCII-Dateien bestehen und somit zu deren Bearbeitung ein simpler Texteditor völlig ausreicht. Dies ist als gravierender Vorteil gegenüber gängigen Textverarbeitungsprogrammen zu werten, da diese nicht selten zu Programmabstürzen neigen, je umfangreicher und größer die damit erstellten Dokumente werden.

An dieser Stelle darf auch der Hinweis nicht fehlen, dass sämtliche zur Erstellung von \LaTeX -Dokumenten notwendigen Programme als kostenlose Downloads im Internet verfügbar sind. Das gilt nicht nur für den \LaTeX -Kern sondern auch für alle Pakete. Auch existieren viele kostenlose Programme, die zur Bearbeitung einfacher Textdateien hervorragend geeignet sind.

Selbst wenn das Textsatzsystem \LaTeX den Anwendern ein im Vergleich zu üblichen Textverarbeitungen überragendes Ausgabeergebnis liefert, sollten freilich einige unschöne Perspektiven nicht unterschlagen werden.

Vom Standpunkt eines Paketentwicklers aus betrachtet stellt das Fehlen eines Verfahrens zur Privatisierung von Kommandos unter Umständen ein größeres Problem dar. Denn Pakete enthalten nicht selten Algorithmen, die nur für den internen Einsatz gedacht sind. Da nun die Möglichkeit besteht, auch intern verwendete Variablen oder Makros und Umgebungen von außerhalb eines Pakets zu manipulieren, ohne hierfür den Originalcode ändern zu müssen, wird

hiervon auch des Öfteren Gebrauch gemacht. Dies kann zwar zur Lösung eines speziellen Problems oder zur Behebung eines Fehlers, wie im Falle der Korrektur des Makros `\@wrindex` geschehen, sehr nützlich sein, ist dessen ungeachtet aber als Schwachstelle des Systems zu bezeichnen, da zum Beispiel durch die fehlerhafte Umdekoration eines Kommandos ein unvorhersagbares Ausgabeergebnis entstehen kann.

Eine weitere Schwäche von \LaTeX ist die durchaus als kompliziert zu bezeichnende Art der Implementierung mehrerer optionaler Kommandoparameter. Hier wäre sicherlich eine prozedurbasierte oder gar klassenorientierte Möglichkeit der Kommandodefinition, wie beispielsweise in der Programmiersprache JavaTM verwendet, besser geeignet. Jedoch wird dieser Ansatz wohl auch in ferner Zukunft eher nicht verfolgt werden, da hierfür die Definition einer gänzlich neuen Sprache und somit die komplette Überarbeitung des \LaTeX -Kernsystems nötig wäre.

Auch stellt das Fehlen einer Möglichkeit zur Definition des Gültigkeitsbereichs von Kommandos und Variablen für Paketentwickler eine gewisse Schwierigkeit dar. Denn jeder Bezeichner, der in einem Paket definiert wird, dabei spielt eine interne oder externe Verwendung keine Rolle, sollte im gesamten \LaTeX -System nur einmal vergeben sein. Anderenfalls besteht die Möglichkeit von Konflikten durch konkurrierende Bezeichner! Diese Problematik spielt in der Praxis aber nur eine untergeordnete Rolle, da in der Regel die Paketentwickler auf diese Einschränkung Rücksicht nehmen. Sollte, wie im Fall des Makros `\PrintIndex`, dennoch ein Konfliktpotenzial vorhanden sein, dann kann dies meistens vernachlässigt werden. Jedoch ist es stets Aufgabe des Entwicklers, die Möglichkeit des jeweiligen Konflikts zu dokumentieren und eventuell auch eine entsprechende Problemumgehung bereitzustellen.

Die Quintessenz des Pakets fomsdt

Einer der Hauptaspekte, die durch das Paket `fomsdt` positiv ins Gewicht fallen, ist die Zusammenfassung vieler verschiedener Vorschriften zu einer sinnvollen Regeluntermenge, die zur Erstellung wissenschaftlicher Abhandlung wirklich benötigt werden. Hierzu zählen unter anderem die verwendete Schriftgröße, der vorgegebene Zeilenabstand und auch der voreingestellte Seitenrand, der eine lesefreundliche Zeilenbreite zur Folge hat. Alle implementierten Regeln dienen einzig und allein der Verbesserung der Lesbarkeit von Dokumenten, die mit Hilfe des Pakets `fomsdt` erstellt werden. Nicht mehr aber auch nicht weniger.

Auch sollen durch das Paket `fomsdt` wenig geübte Endanwender befähigt werden, ihre Abhandlungen in \LaTeX zu setzen. Daher wurden typische Elemente wissenschaftlicher Ausarbeitungen soweit abstrahiert, dass die Definition allgemeingültiger Kommandos möglich wurde. Diese Kommandos erlauben dem

Anwender des Pakets `fomsdt` Abbildungen, Tabellen, Codelistings und auch alle Verzeichnisse einheitlich und vor allem regelkonform zur gestalten.

Ebenfalls besteht die Möglichkeit, charakteristische Absatzlayouts auf einfache Art und Weise zu erzeugen. Zu diesem Zweck stellt `fomsdt` entsprechende Kommandos zur Verfügung, mit deren Hilfe die Gestaltung spezieller Layouts erst praktikabel wird.

Der nächste große Pluspunkt des Pakets `fomsdt` ist die automatisierte Erstellung des Sperrvermerks sowie der Eidesstattlichen Versicherung. Beide für Abschlussarbeiten unverzichtbaren Elemente werden durch `fomsdt` im Ausgabedokument an den „richtigen“ Stellen platziert. Ein Anwender braucht sich also hierüber keinerlei Gedanken zu machen.

Unter Einbeziehung aller in dieser Abschlussarbeit aufgeführten positiven und negativen Aspekte ist nunmehr deutlich erkennbar, dass \LaTeX mit all seinen technischen Möglichkeiten die perfekte Grundlage zur Erstellung von Seminar- und Abschlussarbeiten bietet. Durch Verwendung des Pakets `fomsdt` können die bereits vorhandenen Mittel dann noch effektiver eingesetzt werden, somit ist es sehr empfehlenswert, wissenschaftliche Ausarbeitungen mit \LaTeX und unter Einsatz des Pakets `fomsdt` zu erstellen.

A Quellcode des Pakets fomsdt

A.1 Dokumentenklasse (Quelldatei fomsdt.cls)

```
\plainlisting{../../Paket/fomsdt/src/fomsdtsrc.cls}
```

A.2 Dokumentenbeginn (Quelldatei fomdss.sty)

```
\plainlisting{../../Paket/fomsdt/src/fomdsssrc.sty}
```

A.3 Dokumentenende (Quelldatei fomdes.sty)

```
\plainlisting{../../Paket/fomsdt/src/fomdessrc.sty}
```

A.4 Absatzdefinition (Quelldatei fompar.sty)

```
\plainlisting{../../Paket/fomsdt/src/fomparsrc.sty}
```

A.5 Gleitobjektdefinition (Quelldatei fomfod.sty)

```
\plainlisting{../../Paket/fomsdt/src/fomfodsrc.sty}
```

A.6 Vermischtes (Quelldatei fomisc.sty)

```
\plainlisting{../../Paket/fomsdt/src/fomiscsrc.sty}
```

A.7 Indexkonfiguration (Quelldatei fomidx.ist)

```
\plainlisting{../../Paket/fomsdt/src/fomidxsrc.ist}
```

A.8 Installationsanweisung (Quelldatei readme.txt)

```
\plainlisting{../../Paket/fomsdt/doc/readme.txt}
```

A.9 Benutzerdokumentation (Quelldatei fomman.tex)

```
\plainlisting{../../Paket/fomsdt/doc/fomman.tex}
```

Glossar

Durchschuss Der Begriff *Durchschuss* stammt aus der Zeit, als literarische Werke noch von Hand und aus einzelnen in Blei gegossenen Buchstabenstempeln, den so genannten Lettern, gesetzt wurden. Dabei bezeichnet der Durchschuss vereinfacht ausgedrückt einen zusätzlichen, vertikalen Abstand zwischen zwei Textzeilen, also den Abstand, der sich genau zwischen der Stempelunterkante und der Stempeloberkante zweier untereinander liegender Lettern befindet. Zur Erzeugung eines zusätzlichen Abstandes legte der Schriftsetzer so genanntes Blindmaterial zwischen die einzelnen Buchstabenreihen.

In diesem Zusammenhang sollte auch darauf hingewiesen werden, dass der Durchschuss nicht mit dem Zeilenabstand zu verwechseln ist. Denn als Zeilenabstand wird die Distanz bezeichnet, die sich zwischen den Grundlinien zweier aufeinander folgender Textzeilen befindet, wohingegen der Durchschuss einen zusätzlichen Zwischenraum zwischen zwei Textzeilen bezeichnet [? 21?]. (Seite 8)

Gleitobjekt Ein *Gleitobjekt* stellt ein Objekt dar, das im Fließtext eingefügt wird und zumeist der Darstellung eines bestimmten Zusammenhangs dient. Abbildungen und Tabellen sind gängige Beispiele für derartige Objekte. Sie werden deshalb als Gleitobjekt bezeichnet, weil ein Schriftsetzer sie im Fließtext beliebig verschieben kann, bis der Gesamteindruck des Textes „stimmig“ wirkt. Daher haben Gleitobjekte den großen Nachteil, dass ihre genaue Position innerhalb des Textes nicht vorherbestimmt werden kann! (Seite 17)

Hurenkind Die letzte Zeile eines Absatzes, welche zugleich die erste Zeile einer neuen Seite ist, wird im Druckerhandwerk allgemein mit dem Begriff *Hurenkind* bezeichnet. Derartig zerrissene Absätze gelten im Druckgewerbe als besonders schwere handwerkliche Fehler, da sie sich auf das gesamte Schriftbild und somit auf den Lesefluss besonders negativ auswirken [21?]. (Seite 12)

Java™ Hierbei handelt es sich um eine von der Firma SUN MICROSYSTEMS entwickelte objektorientierte Programmiersprache, die ursprünglich zur Ansteuerung von zum Beispiel Küchengeräten gedacht war. Java-Programme sind im Gegensatz zu Anwendungen in anderen Programmiersprachen dadurch gekennzeichnet, dass der einmal generiert Binärcode *ohne* erneute Kompilierung auf unterschiedlichen → *Plattformen* lauffähig ist. Dieses Verhalten wird unter anderem dadurch erreicht, dass zur Ausführung des jeweiligen Java-Codes ein plattformabhängiger Interpreter, eine so genannte Virtual Machine, verwendet wird. Dieser Umstand macht Java-Programme besonders interessant für jegliche Art von Internetanwendungen. (Seite 99)

Kapitälchen Hierbei handelt es sich um eine spezielle Form der Darstellung einzelner Buchstaben einer bestimmten Schriftart. Dabei werden alle Groß-

buchstaben wie sonst auch dargestellt. Alle Kleinbuchstaben werden ebenfalls als Großbuchstaben, jedoch mit nur zirka Dreiviertel der Höhe der Großbuchstaben, dargestellt. (Seite 9)

Lesevorgang Während des Lesens führt das Auge ruckartige Sprünge – so genannte Sakkaden – durch, ohne hierbei die einzelnen Buchstaben vollständig wahrzunehmen. Dabei hält die Augenbewegung für den Bruchteil einer Sekunde inne. Diese als Fixationspunkte bezeichneten Stopps sind unwillkürlich und dienen der Erfassung der nächsten bis zu zehn Buchstaben in Leserichtung. Eine Verarbeitung des Gelesenen erfolgt innerhalb des Gehirns durch den Vergleich visueller Muster bereits bekannter Wörter [21, 10, 19]. (Seite 7)

Methodenüberladung Das Überladen von Methoden ist ein in der objekt-orientierten Programmierung gängiges Verfahren zur Bereitstellung gleichnamiger Funktionen, die über unterschiedliche Parameterlisten verfügen. Dieses Verfahren wird häufig dort eingesetzt, wo ein einziger Algorithmus auf unterschiedliche Datentypen angewendet werden kann. (Seite 33)

Office Ein Softwarepaket, das die elektronische Verarbeitung von Texten, die Erstellung tabellarischer Kalkulationen, die Erarbeitung von Präsentationen, die Manipulation von Graphiken und vieles mehr ermöglicht, wird im Allgemeinen unter dem Begriff *Office* zusammengefasst. Wie der englische Fachbegriff bereits vermuten lässt, kommen derartige Pakete zumeist in Büros zum Einsatz. Andererseits erfreuen sich auch viele private Anwender der reichhaltigen Möglichkeiten, die Office-Produkte in der Regel bieten. (Seite 3)

Patch Als *Patch* wird in der Sprache der Softwareentwickler die Korrektur eines Programmierfehlers bezeichnet. Häufig werden auch kleinere Softwareupdates unter diesem Begriff zusammengefasst. Dabei handelt es sich aber nur um eine saloppe Umschreibung für ein Paket zu Softwareaktualisierung. (Seite 83)

Plattform Im Umfeld der Informationstechnologie wird unter einer *Plattform* zuerst einmal nur die Hardware und im Speziellen sogar nur die Zentraleinheit eines bestimmten Computersystems verstanden. Im weiteren Sinne kann durch diesen Begriff aber auch das auf einem Computer installierte Betriebssystem gemeint sein. Diese etwas allgemeiner gefasste Definition des Begriffs *Plattform* wird beispielsweise immer dann verwendet, wenn Softwareproduzenten ihre Produkte als \rightarrow *plattformenübergreifend* anpreisen. (Seite 98)

plattformenübergreifend In der Terminologie der Softwareentwicklung wird ein Quellcode genau dann als *plattformenübergreifend* bezeichnet, wenn dieser ohne (größere) Änderungen für unterschiedliche \rightarrow *Plattformen* kompiliert werden kann. Bezogen auf bereits fertig kompilierte Anwendungen kann auch dann von *plattformenübergreifender* Software gesprochen werden, wenn diese

auf verschiedenen Betriebssystemen, zum Beispiel unter Windows[®] und unter Linux, ein identisches Verhalten aufweisen.

An dieser Stelle sollte aber auch festgestellt werden, dass beispielsweise die Firma MICROSOFT[®] unter einer *plattformübergreifenden* Software exakt die Programme versteht, die auf den verschiedenen Windows[®]-Versionen lauffähig sind. Dabei können nicht einmal 16-Bit Programme unter der 64-Bit-Variante des Betriebssystems Windows[®] XP gestartet werden. (Seite 98)

Präambel Unter dem Begriff *Präambel* wird der Teil eines L^AT_EX-Dokuments verstanden, der sich zwischen den beiden Kommandos `\documentclass{}` und `\begin{document}` befindet. Hier werden in der Regel globale Einstellungen für das entsprechende L^AT_EX-Dokument hinterlegt. (Seite 32)

Schusterjunge Die erste Zeile eines Absatzes, welche zugleich die letzte Zeile der vorherigen Seite ist, wird im Druckerhandwerk allgemein mit dem Begriff *Schusterjunge* bezeichnet. Schusterjungen gelten gegenüber \rightarrow *Hurenkindern* als weniger schwerwiegender Fehler [21?]. (Seite 12)

Textverarbeitung Unter einer *Textverarbeitung* wird allgemein ein Computerprogramm verstanden, dessen Aufgabe darin besteht, Texte durch einen Benutzer anlegen, öffnen, ändern, speichern und drucken zu lassen. Eine grundlegende Eigenschaft solcher Programme ist die Möglichkeit zur Texteingabe über die Tastatur des Computers. (Seite 3)

What you see is what you get (WYSIWYG) Ein Computerprogramm zur Erstellung von Textdokumenten, dessen besondere Eigenschaft darin besteht, den eingegebenen Text „druckfertig“ anzuzeigen, wird im Allgemeinen mit dem Begriff *WYSIWYG* beschrieben. Derartige Computerprogramme werden häufig auch als \rightarrow *Textverarbeitung* beziehungsweise als Textverarbeitungsprogramm bezeichnet [24]. (Seite 4)

Literatur

- [1] BEINERT, Wolfgang: *Das Lexikon der westeuropäischen Typographie*. 2007. – URL <http://www.typolexikon.de/s/satzbreite.html>. – Zugriffsdatum: 2007-11-11
- [2] BÖTTCHER, Martin ; INSTITUT FÜR INFORMATIK (Hrsg.): *Einführung in das wissenschaftliche Arbeiten*. Universität Leipzig. 2007. – URL http://wacko.informatik.uni-leipzig.de/de/Lehre/0506/SS/SemASKE/files?get=einfuehrung_in_das_wiss_arbeiten.pdf. – Zugriffsdatum: 2007-12-15
- [3] CHEN, Pehong ; HARRISON, Michael A.: *Index Preparation and Processing*. Juli 1991. – URL <http://www.ctan.org/get/indexing/makeindex/paper/ind.tex>. – Zugriffsdatum: 2008-01-08
- [4] DORLOFF, F.-D.: *Hinweise zur Anfertigung wissenschaftlicher Arbeiten*. März 2006. – URL <http://www.bli.uni-essen.de/lehre/Anfertigung.pdf>. – Zugriffsdatum: 2007-11-19
- [5] ENERGIEWIRTSCHAFTLICHES INSTITUT (Hrsg.): *Leitfaden zur Erstellung von Seminararbeiten*. Universität zu Köln. Juni 2007. – URL http://www.energie.uni-koeln.de/fileadmin/user/PDFs/Leitfaden_Seminararbeiten_Stand_0707.pdf. – Zugriffsdatum: 2007-12-02
- [6] FACHBEREICH WIRTSCHAFT (Hrsg.): *Richtlinien für die Erstellung einer schriftlichen Abschlussarbeit*. Hochschule Darmstadt. Juni 2006. – URL <http://www.fbw.fh-darmstadt.de/Data/BA2D4935-235D-45DE-9C27-91F3E5CDE6CE/ccFiles/FF06FC81-4A95-4ADE-8B03-9ED729E0FA18.pdf>. – Zugriffsdatum: 2007-12-15
- [7] FACHBEREICH WIRTSCHAFTSWISSENSCHAFTEN (Hrsg.): *Formelle Anforderungen und Hinweise zur Bearbeitung von Diplomarbeiten/Seminararbeiten im Fach „Management kleiner und mittlerer Unternehmen“ (PRO KMU)*. Universität Siegen. 2007. – URL http://www.prokmu.de/lehre/pdf_allgemein/bearbeitungsrichtlinien_seminar_diplom.pdf. – Zugriffsdatum: 2007-12-15
- [8] HARVARD BUSINESS SCHOOL (Hrsg.): *Citation Guide*. Oktober 2007. – URL <http://www.library.hbs.edu/guides/citationguide.pdf>. – Zugriffsdatum: 2007-11-19

- [9] HOCHULI, Jost: *Das Detail in der Typografie. Buchstabe, Buchstabenabstand, Wort, Wortabstand, Zeile, Zeilenabstand, Kolumne*. Neuauflage. Niggli Verlag AG, Zürich, November 2005. – ISBN 3-7212-0547-2
- [10] HUESTEGGE, Lynn ; INSTITUT FÜR PSYCHOLOGIE (Hrsg.): *Visuelles Abtasten während des Lesens*. Rheinisch-Westfälische Technische Hochschule Aachen. Mai 2007. – URL <http://www.psych.rwth-aachen.de/ifp-zentral/upload/huestegge/Handout3a.pdf>. – Zugriffsdatum: 2007-11-11
- [11] HUMANWISSENSCHAFTLICHE FAKULTÄT (Hrsg.): *Hinweise zum Zitieren fremder Quellen in sozialwissenschaftlichen Arbeiten*. Universität zu Köln. 2007. – URL <http://www.uni-koeln.de/ew-fak/seminar/sowi/politik/Hinweise%20zum%20Zitieren%20fremder%20Quellen%20in%20sozialwissenschaft%85.pdf>. – Zugriffsdatum: 2007-11-16
- [12] INSTITUT FÜR BETRIEBSWIRTSCHAFTLICHE GELDWIRTSCHAFT (Hrsg.): *Empfehlungen für Studierende zur Anfertigung wissenschaftlicher Arbeiten*. Georg-August-Universität Göttingen. Oktober 2003. – URL http://ifbg.wiwi.uni-goettingen.de/downloads/allgemeines/Empfehlungen-zum-Anfertigen_Okt03.pdf. – Zugriffsdatum: 2007-12-08
- [13] KÜMPEL, Thomas ; SENG, Anja: *Leitfaden zur formalen Gestaltung von Seminar- und Diplomarbeiten*. Juni 2007. – URL <http://campus.bildungszentrum.de/nfcampus/UserFiles/File/formalesarbeiten.pdf>. – Zugriffsdatum: 2007-12-02
- [14] KÖNIG, Anne Rose: *Alles Buch; Studien der Erlanger Buchwissenschaft VII*. 2004. – URL <http://www.buchwiss.uni-erlangen.de/AllesBuch/Koenig/Koenig.pdf>. – Zugriffsdatum: 2007-11-11
- [15] KOWALSKI, Susann: *Leitfaden zur Erstellung von Seminararbeiten*. Juni 2007. – URL <http://studieren.wi.fh-koeln.de/kurse/homepage/ftp/Anleitung%20Diplomarbeit.pdf>. – Zugriffsdatum: 2007-12-02
- [16] KRAWINKEL, Michael: *Leitfaden zur Anfertigung einer Studien-, Master-, Diplom- oder Doktorarbeit*. April 2006. – URL <http://www.uni-giessen.de/fbr09/int-nutr/Diplom-und%20Doktorarbeiten%20im%20AK/Leitfaden%20Pruefungsarbeiten.pdf>. – Zugriffsdatum: 2008-01-19
- [17] LAMPORT, Leslie: *MakeIndex: An Index Processor For L^AT_EX*. Februar 1987. – URL <ftp://tug.ctan.org/pub/tex-archive/indexing/makeindex/doc/makeindex.pdf>. – Zugriffsdatum: 2008-02-14

- [18] LJUBENOVA, Martina: *Auswirkung von Schriftart und Satz auf die Lesbarkeit und Lesegeschwindigkeit (Aufsatz im Studiengang: Medieninformatik)*. 2004. – URL http://www.medien.ifi.lmu.de/fileadmin/mimuc/mmi_ws0304/exercise/aufsaetze/Martina_Ljubenova.html. – Zugriffsdatum: 2007-11-11
- [19] PLETL, Heinz: *Aufsatz über den Leseprozess*. 2007. – URL <http://www.typoguide.de/downloads/leseprozess.pdf>. – Zugriffsdatum: 2007-11-11
- [20] ROTERMUND, Hermann: *Gute Typographie auf Papier und im Web*. 2007. – URL <http://www.typoguide.de/downloads/aufsatzrotermund.pdf>. – Zugriffsdatum: 2007-11-11
- [21] RUNK, Claudia: *Grundkurs Typografie und Layout*. 1. Auflage. Galileo Press GmbH, Bonn, Februar 2006. – ISBN 3-89842-406-5
- [22] SCHOOR, Cornelia: *Richtlinien zur formalen Gestaltung von Hausarbeiten, Magister-, Bachelor- und Masterarbeiten*. 2005. – URL http://www.tu-chemnitz.de/phil/elearning/files/Richtlinien_Manuskriptgestaltung.pdf. – Zugriffsdatum: 2008-01-19
- [23] TALBOT, Nicola: *makedtx v0.94b: a Perl script to help create a DTX file from source code*. August 2007. – URL <http://theoval.cmp.uea.ac.uk/~nlct/latex/packages/makedtx/makedtx.pdf>. – Zugriffsdatum: 2007-11-06
- [24] TESCHNER, Helmut: *Druck- & Medien-Technik. Informationen, gestalten, produzieren, verarbeiten*. 11. Auflage. Fachschriften-Verlag, Fellbach, 2003. – ISBN 3-931436-88-8
- [25] TEUBER, Mathias: *Kriterien für die Lesbarkeit von Texten*. 2007. – URL http://www.medien.ifi.lmu.de/fileadmin/mimuc/mmi_ws0304/exercise/aufsaetze/Mathias_Teuber.html. – Zugriffsdatum: 2007-11-11
- [26] THE L^AT_EX3 PROJECT (Hrsg.): *L^AT_EX 2_ε for class and package writers*. Februar 2006. – URL <http://www.dante.de/help/latex/guide/clsguide.pdf>. – Zugriffsdatum: 2007-11-06
- [27] WILHELM, Andreas: *Leitfaden zum Anfertigen wissenschaftlicher Hausarbeiten*. Mai 2006. – URL http://www.uni-kiel.de/sport/test/media/arbeitsbereiche/sportwissenschaft1/steckbriefe/wilhelm/Materialien/methoden/Leitfaden5_040506_Kiel.pdf. – Zugriffsdatum: 2008-01-19

- [28] WINKER, Gabriele ; DERBOVEN, Wibke: *Einführung in wissenschaftliches Arbeiten*. 2007. – URL http://www.tu-harburg.de/agentec/lehre/ast/AST_Wissenschaft.pdf. – Zugriffsdatum: 2007-11-16

Index

A

a4wide (*Paket*) 91
Abbildung → *Gleitobjekt* 17
 –, Beschriftung 17
 –, Quellenangabe 18
Abbildungsverzeichnis 20, 23, 36, 54
\Abbrev (*Makro*) 54 f., 57
abbreviation (*Option*) 52 ff.
abbrevrefpage (*Option*) 53
Abkürzungen 11
 –, Erläuterung 11
Abkürzungsverzeichnis 11, 20, 23, 36, 52, 54
 –, Seitenreferenz 53
 –, überflüssig 11
affidavit (*Option*) 39, 48
\AffidavitContent (*Makro*) . 48 f., 51
\AffidavitDate (*Makro*) ... 49, 51
\AffidavitHeader (*Makro*) 49
\AffidavitSigLeft (*Makro*) ... 49
\AffidavitSigRight (*Makro*) .. 49
Anhang 26, 28
 –, Sonderstellung 26
article (*Paket*) 37, 89
\AtEndDocument (*Makro*) ... 48, 93
\author (*Makro*) 39
\Axel (*Makro*) 83

B

beamer (*Paket*) 89 f.
Benutzerdokumentation 86
Benutzerhandbuch .. 86 ff., 95, 97
Beschriftung 17 f.
 –, Abbildung 17
 –, Codelisting 18
 –, oberhalb 18

–, Objekt 17
 –, Platzierung 17
 –, Regel 19
 –, Tabelle 18
 –, unterhalb 17
\bibliographystyle (*Makro*) .. 61
bibtex (*Datei*) 58
book (*Paket*) 89
box (*Option*) 72, 76

C

caption (*Paket*) 91
\chapter (*Makro*) 69
Citation (*Umgebung*) 63
classiccaption (*Option*) 71
classictitle (*Option*) 39 f., 43
CodeListing (*Umgebung*) 78 f.
\CodeListingFile (*Makro*) .. 78 ff.
Codelisting → *Gleitobjekt* 18, 57
 –, Beschriftung 18
 –, Quellenangabe 18
color (*Paket*) 91
coloredlinks (*Option*) 38
\Copyright (*Makro*) 83
\Course (*Makro*) 41

D

Danksagung 22
\Date (*Makro*) 41
Dateien
 –, **bibtex** 58
 –, **din.ist** 83
 –, **dinat.bst** 61
 –, **doc.sty** 92, 94
 –, **fomdes.sty** 36, 57, 84, 92
 –, **fomdss.sty** 36, 52, 92
 –, **fomfod.sty** 36, 70, 72, 77, 89 f.

-
- , fomidx.ist 37, 83 f., 86
 - , fomisc.sty 37, 80, 83
 - , fomman.tex 88
 - , fompar.sty 36, 63, 67
 - , fomsdt.cls .. 36 ff., 40, 48, 93
 - , fomsdt.dtx 87 f., 95
 - , fomsdt.ins 87
 - , makeindex 37, 53, 59, 86
 - , pdfLaTeX 51
 - , readme.txt 87
 - \Description (Makro) 69 f.
 - def (Option) 82
 - \DefaultItemIndent (Makro) .. 68
 - \DefaultItemOffset (Makro) .. 68
 - \DefaultRemarkIndent (Makro) 68
 - \DefaultRemarkOffset (Makro) 68
 - din (Option) 82
 - din.ist (Datei) 83
 - dinat.bst (Datei) 61
 - doc.sty (Datei) 92, 94
 - \documentclass (Makro) ... 43, 93
 - Dokumentenbeginn 36, 52
 - , Optionen 52
 - Dokumentenende 36, 57
 - , Optionen 57
 - Dokumentenklasse 32, 35 ff.
 - , Optionen 38
 - double (Option) 62
 - \DoublePicture (Makro) 74, 76, 90
 - \DoubleSubPicture (Makro) 74, 76

 - E**
 - Erklärung (ehrenwörtlich) .. 28 f.,
39, 48 ff.

 - F**
 - fancyhdr (Paket) 91
 - Fehlermeldung ... 52, 58, 60, 89 f.
 - figure (Umgebung) 90
 - fom (Option) 44, 47, 49
 - fomdes (Paket) 58 – 61
 - fomdes.sty (Datei) . 36, 57, 84, 92
 - fomdss (Paket) 53 f.
 - fomdss.sty (Datei) 36, 52, 92
 - fomfod (Paket) 70 f., 78
 - fomfod.sty (Datei) . 36, 70, 72, 77,
89 f.
 - fomidx.ist (Datei) ... 37, 83 f., 86
 - fomisc (Paket) 81, 83
 - fomisc.sty (Datei) 37, 80, 83
 - fomman.tex (Datei) 88
 - fompar (Paket) 69
 - fompar.sty (Datei) 36, 63, 67
 - fomsdt (Paket) 32, 35, 37 – 40,
42 – 45, 47 – 51, 56, 63, 81,
83, 86 – 90, 92 – 95, 97 – 100
 - fomsdt.cls (Datei) 36 ff., 40, 48, 93
 - fomsdt.dtx (Datei) 87 f., 95
 - fomsdt.ins (Datei) 87
 - fontenc (Paket) 91
 - footmisc (Paket) 91
 - \footnote (Makro) 55
 - Formatierung 36
 - , Abbildung 36
 - , Absatz 36, 38
 - , Blattrand 36
 - , Codelisting 36
 - , Fußnote 36
 - , Fußzeilen 36
 - , Index 37
 - , Kopfzeile 36
 - , Tabelle 36
 - Fremdliteratur 27
 - Fußnote 11
 - , Informationsgehalt 12
 - , überlesen 12
 - , Vermehrung 16

 - G**
 - \General (Makro) ... 41, 47, 49, 51

Gleitobjekt 17, 70
 –, Beschriftung 17, 19
 –, Optionen 71
 –, Quellenangabe 19
 –, Referenzierung 17
 Gliederungsebene 10, 24
 Gliederungstiefe 10
 gloss *(Paket)* 58, 60 f., 91
 glossrefpage *(Option)* 58
 Glossar 27 f., 57
 –, Seitenreferenz 58
 glossary *(Option)* 57 ff.
 \graphicspath *(Makro)* 73
 graphicx *(Paket)* 91

H

Harvard-Notation 14, 16
 –, abgewandelt 15 f.
 Hervorhebung 9, 38
 –, Eigenname 9
 –, Fettdruck 10 f.
 –, Kapitälchen 9
 –, Kopfzeile 10
 –, Linie 10
 –, Linien 10
 –, Schriftart 9
 –, Tabelle 10
 –, Textpassage 10
 –, Unterstreichung 10
 hyperref *(Paket)* 91

I

ifpdf *(Paket)* 91
 ifthen *(Paket)* 91
 Implementierung 32
 index *(Option)* 59
 Index 27, 29, 37, 59
 –, Formatierung 37
 –, Manipulation 37

Indexkonfiguration 37
 Inhaltsverzeichnis .. 11, 20, 22, 36,
 52 f., 69, 97
 inputenc *(Paket)* 91
 Installation 87, 92, 98
 Installationsmethode 87, 98
 \Institute *(Makro)* 41
 iso *(Option)* 82
 \Item *(Makro)* 64 – 69
 \ItemIndent *(Makro)* 68
 \ItemOffset *(Makro)* 68

K

Kapitelnummerierung 26
 Klassenoption 40, 43 f.
 Kommando → *Makro* 32 ff.
 Konfiguration .. 32, 36 ff., 48, 52,
 57, 67, 71, 78, 83
 Kopfzeile 10, 24, 38 f.
 Kurzbeleg 13 – 17

L

Lesbarkeit 7, 69
 –, Abkürzungen 11
 –, Bildschirmtext 24
 –, Fußnote 12, 15 f.
 –, Gliederungstiefe 10
 –, Hervorhebung 9
 –, Linien 10
 –, Querverweise 12
 –, Randbereich 12
 –, Schriftbild 9
 –, Schriftgröße 9
 –, Serifenschrift 9 f.
 –, Tabelle 10
 –, Unterstreichung 10
 –, Zeilenabstand 8 f.
 –, Zeilenlänge 7 f.
 Lesefluss → *Lesbarkeit* ... 7, 15 f.

-
- , beeinträchtigen 17
 - , Gleitobjekt 18
 - , unterbrechen 7, 11
 - Lesevorgang 7 f.
 - letter *(Paket)* 89 f.
 - listingcolored *(Option)* 71
 - listings *(Paket)* 91
 - listings *(Option)* 57, 59 f.
 - Listingverzeichnis 60
 - Literaturquelle 25
 - , Auswertung 25
 - , Bewertung 25
 - Literaturverzeichnis 13 ff., 17,
27 f., 36, 58, 61
 - lockflag *(Option)* 39 f., 43 f.
 - \LockflagCompany *(Makro)* 45, 47 f.
 - \LockflagContent *(Makro)* 44,
47 ff.
 - \LockflagDate *(Makro)* 45, 48
 - \LockflagHeader *(Makro)* 45
 - \LockflagSigLeft *(Makro)* 44
 - \LockflagSigRight *(Makro)* ... 44
- M**
- \makeatletter *(Makro)* 34
 - makeindex *(Datei)* ... 37, 53, 59, 86
 - \maketitle *(Makro)* 40, 42 f.
 - makedtx *(Paket)* 87
 - makeidx *(Paket)* 91
 - Makro 33, 35 ff.
 - , lokal 34
 - Makros
 - , \Abbrev 54 f., 57
 - , \AffidavitContent .. 48 f., 51
 - , \AffidavitDate 49, 51
 - , \AffidavitHeader 49
 - , \AffidavitSigLeft 49
 - , \AffidavitSigRight 49
 - , \AtEndDocument 48, 93
 - , \author 39
 - , \Axel 83
 - , \bibliographystyle 61
 - , \chapter 69
 - , \CodeListingFile 78 ff.
 - , \Copyright 83
 - , \Course 41
 - , \Date 41
 - , \Description 69 f.
 - , \DefaultItemIndent 68
 - , \DefaultItemOffset 68
 - , \DefaultRemarkIndent ... 68
 - , \DefaultRemarkOffset ... 68
 - , \documentclass 43, 93
 - , \DoublePicture ... 74, 76, 90
 - , \DoubleSubPicture ... 74, 76
 - , \footnote 55
 - , \General 41, 47, 49, 51
 - , \graphicspath 73
 - , \Institute 41
 - , \Item 64 – 69
 - , \ItemIndent 68
 - , \ItemOffset 68
 - , \LockflagCompany ... 45, 47 f.
 - , \LockflagContent .. 44, 47 ff.
 - , \LockflagDate 45, 48
 - , \LockflagHeader 45
 - , \LockflagSigLeft 44
 - , \LockflagSigRight 44
 - , \makeatletter 34
 - , \maketitle 40, 42 f.
 - , \Mark 55 ff.
 - , \nomlabel 56
 - , \paragraph 69
 - , \part 69
 - , \PDFAuthor 51
 - , \PDFProducer 51
 - , \PDFSubject 51
 - , \PDFTitle 51
 - , \PrintAbbrev 54 f., 57
 - , \PrintBibliography 61 f.
 - , \PrintContents 53 f.
 - , \PrintFigures 54
 - , \PrintGlossary 60, 62
 - , \PrintIndex . 61 f., 92, 94, 99

-
- , \PrintListings 60, 62
 - , \PrintTables 54
 - , \Professor 42
 - , \Raise 81
 - , \Registered 83
 - , \Remark 64 f., 67 ff.
 - , \RemarkIndent 68
 - , \RemarkOffset 68
 - , \section 69, 92
 - , \SeeAlso 62
 - , \Semester 41
 - , \SinglePicture . 72 ff., 76, 91
 - , \Student 42
 - , \Subject 41
 - , \subparagraph 69
 - , \textwidth 72
 - , \title 39
 - , \Title 42, 47
 - , \today 41, 81
 - , \TodayShort 81 – 84
 - , \Trademark 83
 - , \@wrindex 84, 99
 - Makrosprache 32
 - \Mark *(Makro)* 55 ff.
 - Meldung 53, 58 f.
 - Methodenüberladung 33
 - moderntitle *(Option)* 39 f., 43
 - multicol *(Paket)* 91
- N**

- natbib *(Paket)* 91
- ngerman *(Paket)* 91
- nobibliography *(Option)* 58 f.
- nopagebreak *(Option)* ... 53, 59, 92
- nochapter *(Option)* 39
- noheadrule *(Option)* 38 f.
- nomenc1 *(Paket)* 52 f., 55 ff., 91
- \nomlabel *(Makro)* 56

O

Optionen

- , abbreviation 52 ff.
- , abbrevrefpage 53
- , affidavit 39, 48
- , box 72, 76
- , classiccaption 71
- , classictitle 39 f., 43
- , coloredlinks 38
- , def 82
- , din 82
- , double 62
- , fom 44, 47, 49
- , glossrefpage 58
- , glossary 57 ff.
- , index 59
- , iso 82
- , listingcolored 71
- , listings 57, 59 f.
- , lockflag 39 f., 43 f.
- , moderntitle 39 f., 43
- , nobibliography 58 f.
- , nopagebreak 53, 59, 92
- , nochapter 39
- , noheadrule 38 f.
- , secnumdepth 69
- , sig 44, 48, 51
- , single 62
- , std 44, 47, 51
- , tocdepth 69
- , toctotoc 52, 97
- , triple 62

P

Pakete

- , a4wide 91
- , article 37, 89
- , beamer 89 f.
- , book 89
- , caption 91
- , color 91
- , fancyhdr 91
- , fomdes 58 – 61
- , fomdss 53 f.

-
- , fomfod 70 f., 78
 - , fomisc 81, 83
 - , fompar 69
 - , fomsdt 32, 35, 37 – 40,
42 – 45, 47 – 51, 56, 63, 81,
83, 86 – 90, 92 – 95, 97 – 100
 - , fontenc 91
 - , footmisc 91
 - , gloss 58, 60 f., 91
 - , graphicx 91
 - , hyperref 91
 - , ifpdf 91
 - , ifthen 91
 - , inputenc 91
 - , letter 89 f.
 - , listings 91
 - , makedtx 87
 - , makeidx 91
 - , multicol 91
 - , natbib 91
 - , ngerman 91
 - , nomencl 52 f., 55 ff., 91
 - , report 89
 - , rotating 91
 - , subfig 90 f.
 - , tabularx 91
 - , textfit 91
 - , textpos 91
 - , ulem 91
 - Paketelemente 37
 - Paketstruktur 35
 - , Dateien 87
 - , Unterteilung 35
 - \paragraph (Makro) 69
 - Parameter 32
 - , obligatorisch 32
 - , optional 32 ff.
 - \part (Makro) 69
 - \PDFAuthor (Makro) 51
 - pdfLaTeX (Datei) 51
 - \PDFProducer (Makro) 51
 - \PDFSubject (Makro) 51
 - \PDFTitle (Makro) 51
 - Pflichtparameter 33 f.
 - PlainFigure (Umgebung) . . . 74, 76, 91
 - Präambel 32, 35
 - \PrintAbbrev (Makro) 54 f., 57
 - \PrintBibliography (Makro) . . 61 f.
 - \PrintContents (Makro) 53 f.
 - \PrintFigures (Makro) 54
 - \PrintGlossary (Makro) ... 60, 62
 - \PrintIndex (Makro) . . 61 f., 92, 94,
99
 - \PrintListings (Makro) ... 60, 62
 - \PrintTables (Makro) 54
 - \Professor (Makro) 42
- ## Q
- Quellenangabe 13 – 16
 - , überlesen 16
 - Quellenverweis 13
 - Quellenzitat 12
 - , bevorzugt 17
 - , Fußnote 13 – 16
 - , Harvard 14, 16
 - , abgewandelt 15 f.
 - , Kurzbeleg 13, 15
 - , Schlüssel 14, 16
 - , traditionell 13
 - , Vollbeleg 13, 15
 - Querverweis ... 12, 38, 62, 79, 91
- ## R
- \Raise (Makro) 81
 - readme.txt (Datei) 87
 - \Registered (Makro) 83
 - \Remark (Makro) 64 f., 67 ff.
 - \RemarkIndent (Makro) 68
 - \RemarkOffset (Makro) 68
 - report (Paket) 89
 - Richtlinie 2 f., 5
 - rotating (Paket) 91

S

Sachwortverzeichnis ... 27, 57, 61
 Schriftbild 9
 secnumdepth (Option) 69
 \section (Makro) 69, 92
 \SeeAlso (Makro) 62
 Seitennummerierung 29
 –, Dokumentbeginn 30
 –, Dokumentende 30
 –, Textteil 30
 Seitenvorschub 24
 –, Kapitelende 24
 –, Verzeichnisse 23, 29
 \Semester (Makro) 41
 sig (Option) 44, 48, 51
 Signatur 44 ff., 48 f., 51
 single (Option) 62
 \SinglePicture (Makro) 72 ff., 76,
 91
 Sperrvermerk 22, 39 f., 43
 –, Abhängigkeiten 47
 –, FOM-Sperrvermerk 46
 –, Inhalt 44
 –, Signatur 44
 –, Standardsperrvermerk 45
 –, Überschrift 45
 –, Unternehmen 45
 std (Option) 44, 47, 51
 Stichwortverzeichnis 27, 59, 61 f.,
 84, 90
 Stildatei 35
 \Student (Makro) 42
 subfig (Paket) 90 f.
 \Subject (Makro) 41
 \subparagraph (Makro) 69

T

Tabelle → *Gleitobjekt* 18
 –, Beschriftung 18
 –, Quellenangabe 18

Tabellenverzeichnis 20, 23, 36, 54
 Table (Umgebung) 77 f.
 tabularx (Paket) 91
 textfit (Paket) 91
 Textformatierung 32
 textpos (Paket) 91
 Textteil 23
 –, Abhandlungskern 25
 –, Aufbau 24
 –, Einleitung 25
 –, Grundlagen 25
 –, Schlussbetrachtung 25
 Textverarbeitung 24
 \textwidth (Makro) 72
 Textzwischenraum 23
 Titelblatt 36, 39, 42, 46
 Titelseite 22, 39
 –, Angaben 41
 –, Benutzug 42
 –, klassisch 39, 43
 –, modern 39, 43
 \title (Makro) 39
 \Title (Makro) 42, 47
 tocdepth (Option) 69
 toctotoc (Option) 52, 97
 \today (Makro) 41, 81
 \TodayShort (Makro) 81 – 84
 \Trademark (Makro) 83
 triple (Option) 62
 Trivialabkürzungen 11

U

ulem (Paket) 91
 Umgebung 34
 Umgebungen
 –, Citation 63
 –, CodeListing 78 f.
 –, figure 90
 –, PlainFigure 74, 76, 91
 –, Table 77 f.

V

Versicherung (eidesstattlich) 28 f.,
39, 48 ff.

Verzeichnis 19
–, Platzierung 20, 28
–, Reihenfolge 20, 22, 28
–, Zweck 19
Vollbeleg 13, 15, 27
Vorwort 22

W

Warnmeldung 39, 60, 62

\@wrindex *(Makro)* 84, 99

WYSIWYG 4

Z

Zeilenabstand 8, 23 f.

–, optimal 9

Zeilenlänge 7

Zitat → *Quellenzitat* 12, 15

Zitatquelle 12

Zitierweise 12, 15

–, Bewertung 15

–, Überblick 13

Ehrenwörtliche Erklärung

Hiermit versichere ich, dass die vorliegende Diplomarbeit von mir selbstständig und ohne unerlaubte Hilfe angefertigt worden ist, insbesondere, dass ich alle Stellen, die wörtlich oder annähernd wörtlich aus Veröffentlichungen entnommen sind, durch Zitate als solche gekennzeichnet habe. Ich versichere auch, dass die von mir eingereichte schriftliche Version mit der digitalen Version übereinstimmt. Weiterhin erkläre ich, dass die Diplomarbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat.

Kaarst, den 28.01.2015
(Ort, Datum)

(Eigenhändige Unterschrift)