

```
\addtocontents{lol}{\bcom Text \ecom}
```

soll den Text `\bcom Text \ecom` in die Datei schreiben. Trotz Einhaltung aller Syntaxregeln quittiert L^AT_EX den Versuch mit einer großen Anzahl an Fehlermeldungen. Dies liegt daran, daß L^AT_EX versucht, die beiden Befehl *auszuwerten* und diese Ersatztexte in die Datei zu schreiben, was zu großen Mengen an in diesem Kontext unerlaubten Sonderzeichen führt. Die Befehle fallen sozusagen in eine große Zahl von Bruchstücken auseinander.

Der Befehl `\protect` verhindert die Auswertung der Befehle und schreibt stattdessen die Befehlswoorte literal in die Datei. Dazu muß der Befehl `\protect` vor den fraglichen Kommandowörtern stehen:

```
\protect
```

```
\addtocontents{lol}{\protect\bcom Text \protect\ecom}
```

Auch wenn Sie zum Beispiel einen Eintrag ins Inhaltsverzeichnis mit einem zusätzlichen `\numberline` planen, muß dieser Befehl geschützt werden:

```
\addtocontents{toc}
  {\protect\numberline{\thesection} Abschnitt}
```

2.4 Pakete

L^AT_EX stellt Mechanismen und Befehle bereit, die speziell auf die Bedürfnisse der Anwender, die Pakete schreiben oder modifizieren wollen, zugeschnitten sind (synonym zu Paket ist in der Literatur häufig der Begriff Stildatei zu finden). Lesenswert sind auf jeden Fall die Dokumentationen [15, 16], die wertvolle Hinweise für Autoren enthalten!

2.4.1 Die Identifizierung der Pakete

Wie oft geschieht es beim Einsatz von Paketen, daß diese nicht das gewünschte Ergebnis erbringen. Eine genaue Untersuchung bringt an den Tag, daß ein Paket für eine mittlerweile veraltete L^AT_EX 2_ε- (alle sechs Monate wird ein Update veröffentlicht) oder gar L^AT_EX-2.09-Version geschrieben war. L^AT_EX 2_ε vergibt deshalb von vornherein Befehle, aus denen ersichtlich wird, für welche Konfiguration oder Version das Paket geschrieben wurde. Zunächst können Sie angeben, welches T_EX-Format Sie voraussetzen:

```
\NeedsTeXFormat{<Format>}[<Datum>]
```

```
\NeedsTeXFormat
```

Die Angabe `<Format>` informiert darüber, mit welchem Format die L^AT_EX-Version ausgestattet sein muß, um als Grundlage für Ihr Paket dienen zu können. Sie setzen hier in der Regel die Zeichenfolge `LaTeX2e` ein. Die freigestellte Angabe von `<Datum>` ermöglicht es

dem System, eine Meldung zu generieren, wenn das Paket oder die Klasse mit einer L^AT_EX-Version benutzt wird, die vor dem angegebenen Datum verteilt wurde und daher eventuell nicht die notwendige Funktionalität bietet. Das Datum muß in der ISO-Norm geschrieben werden, für den 13. September 1998 wird 1998/09/13 angegeben.

Die drei folgenden Kommandos dienen der Selbstidentifikation des Paketes, der Klassendatei oder einer Hilfsdatei:

```
\ProvidesPackage \ProvidesPackage{<Name>}[<Datum> [<Info>]]
\ProvidesClass  \ProvidesClass{<Name>}[<Datum> [<Info>]]
\ProvidesFile   \ProvidesFile{<Name>}[<Datum> [<Info>]]
```

<Name> ist dabei der Name des Paketes oder der Klassendatei, <Datum> ein optionales Datum zur Sicherstellung einer hinreichend neuen L^AT_EX-Version. Auf dieses Datum kann weiterer Kommentar folgen. Im Falle von `\ProvidesFile` muß <Name> der komplette Name der Datei einschließlich der Dateiergung sein.

2.4.2 Nachladen von Paketen und Klassen

Eine große Anzahl von Problemen kann durch leichte Modifikationen an bereits vorhandenen Paketen oder Klassen gelöst werden. Sind nur kleine Änderungen erforderlich, ist es noch günstiger, nur diese in eine eigene Paketdatei zu schreiben und am Beginn oder Ende dieser Datei die unveränderte Originaldatei nachzuladen. Bei Bedarf können mit den beiden Kommandos

```
\RequirePackage \RequirePackage[<Optionen>]{<Name>}[<Datum> [<Info>]]
...WithOptions \RequirePackageWithOptions{<Name>}[<Datum> [<Info>]]
```

noch weitere Pakete geladen werden, wobei beim ersten Befehl die in der Liste aufgeführten (oder mit einem `\PassOptionsToPackage`-Befehl bezeichneten) Optionen übergeben werden, die aus der Sicht des Paketes <Name> *lokal* sind. Der Befehl ist mit dem Aufruf `\usepackage` identisch. Dagegen werden die Optionen, mit denen das Paket oder die Klasse selbst aufgerufen wurde, bei dem Befehl `\RequirePackageWithOptions` weitergereicht. Sie können als *globale* Optionen betrachtet werden und werden auch dann vom Paket <Name> benutzt, wenn das von Ihnen geschriebene Paket, in dem der Befehl `\RequirePackageWithOptions` steht, die Optionen selbst auswertet. Die Optionen werden in solchen Fällen also zweimal behandelt, von jedem Paket einmal.

Möchten Sie auf einer bereits existierenden Klasse <Name> aufbauen, verwenden Sie stattdessen einen der Befehle

```
\LoadClass \LoadClass[<Optionen>]{<Name>}[<Datum> [<Info>]]
...WithOptions \LoadClassWithOptions{<Name>}[<Datum> [<Info>]]
```

Die Kommandos dürfen nur in Klassen eingesetzt werden! `\LoadClass` ist mit dem Aufruf `\documentclass` identisch und gibt die in der Liste oder mit einem `\PassOptionsToClass`-Befehl angeführten Optionen weiter. `\LoadClassWithOptions` hingegen gibt die Optionen weiter, mit denen die aufrufende Klasse selbst aufgerufen wurde. Im nächsten Abschnitt werden Sie hierzu einige weitere Erläuterungen finden.

2.4.3 Optionen

Für jede Option `<Name>`, die das Paket oder die Klasse unterstützen soll, ist der Befehl

```
\DeclareOption{<Name>}{<Befehle>}
```

\DeclareOption

zu verwenden. Er teilt L^AT_EX mit, daß beim Auftreten von `<Name>` die Befehlsfolge `<Befehle>` auszuführen ist. Schreiben Sie eine neue Klasse, so reicht diese von Ihnen angegebene Optionen, die nicht mit `\DeclareOption` einem Programmfragment in der Klasse entsprechen, an vorhandene Pakete weiter. Schreiben Sie ein neues Paket, so kann es ebenfalls vorkommen, daß eine Option angefordert wird, für die kein Code vorhanden ist. Pakete generieren in solchen Fällen eine Fehlermeldung, da sie nicht mehr auf weitere Pakete zurückgreifen können, die diese Option bearbeiten könnten. Solche Fälle können Sie mit dem Befehl `\DeclareOption*{<Befehle>}` abfangen. Hiermit wird eine Befehlsfolge angegeben, die eine Voreinstellung darstellt und immer ausgeführt wird, wenn eine nicht definierte Option angefordert wurde.

*\DeclareOption**

Innerhalb der Befehlsargumente von `\DeclareOption` sind zwei nützliche Kommandos definiert. `\CurrentOption` wird zum Namen der augenblicklich aktuellen Option expandiert und eignet sich daher hervorragend zum Einsatz im Defaultzweig nach `\DeclareOption*`, um zu erfahren, welche nicht definierte Option aufgerufen wurde. Die Folge

\CurrentOption

```
\DeclareOption*{Nicht definierte Option:
  \CurrentOption\par}
```

erzeugt eine Liste aller Optionen, die angegeben wurden, aber nicht definiert sind. Der Befehl `\OptionNotUsed` fügt die Option den »unused options« hinzu und kann dazu dienen, vorläufige Platzhalteroptionen einzurichten.

\OptionNotUsed

Innerhalb der Befehlsposition der `\DeclareOption`-Kommandos (aber nicht nur dort) können die Befehle

```
\PassOptionsToPackage{<Optionen>}{<Paket>}
\PassOptionsToClass{<Optionen>}{<Klasse>}
```

\PassOptionsToPackage
\PassOptionsToClass

eingesetzt werden. Sie fügen die gegebenen `<Optionen>` der Liste für das Paket `<Paket>` oder die Klasse `<Klasse>` hinzu. Diese Optionen werden beim nächsten Aufruf des Paketes oder der Klasse mit einem der Befehle `\RequirePackage`, `\usepackage` oder `\LoadClass` wirksam. Sie können so die Behandlung einzelner Optionen selbst übernehmen, andere wiederum an spezielle Pakete weiterreichen.

`\ProcessOptions`

Die tatsächliche Auswertung der Optionen und die Ausführung des Codes erfolgt erst mit dem Befehl `\ProcessOptions\relax`. Das Kommando `\relax` wird als Abschluß des Befehlswortes empfohlen, da eine Sternform `\ProcessOptions*` existiert. Verzichten Sie auf `\relax` und folgt – aus welchen Gründen auch immer – ein Stern im Eingabestrom, so würde dieser ungewollt das `\ProcessOptions`-Kommando in die Sternform konvertieren.

Der Befehl `\ProcessOptions` sucht zunächst für jede globale oder lokale Option den mit `\DeclareOption` definierten Code. Ist ein solcher vorhanden, wird er in der Reihenfolge der `\DeclareOption`-Befehle ausgeführt. Alle Optionen, die mit einem der Befehle `\RequirePackage`, `\PassOptionsToPackage` oder `\usepackage` explizit an das Paket (und nur an dieses) übergeben wurden, sind lokal.

Ist kein Code für die Option vorhanden, so wird versucht, eine innerhalb von `\DeclareOption*` definierte Befehlsfolge zu finden. Bleibt auch diese Suche ohne Ergebnis und wird `\ProcessOptions` in einem Paket verwendet, wird eine Fehlermeldung ausgegeben. Klassen liefern an dieser Stelle keine Meldung, da sie unbenutzte Optionen weiterreichen können.

`\ProcessOptions*`

Eine ähnliche Wirkung besitzt `\ProcessOptions*`. Der Unterschied zur Form ohne Stern besteht darin, daß die Optionen in der Reihenfolge ihres Auftretens in der Optionenliste (nicht in der ihrer Definition) ausgeführt werden.

Defaultoptionen

`\ExecuteOptions`

Möchten Sie in einer Klasse eine Liste von voreingestellten Optionen einführen, die ohne explizite Nennung gültig sind, können Sie das Kommando `\ExecuteOptions{<Optionen>}` unmittelbar vor `\ProcessOptions` benutzen. Es sucht für jede Option der Liste den entsprechenden Code und führt ihn aus. Kann er nicht gefunden werden, wird die Option stillschweigend übergangen.

Betrachten wir das Problem der Weitergabe von Optionen an Pakete und Klassen genauer. Stellen Sie sich eine Klasse `artikel.cls` vor, die einige Optionen selbst verarbeitet und auf einer Klasse `base.cls` aufbaut. `artikel.cls` wird in einer Testdatei aufgerufen, die fünf Optionen spezifiziert:

```
\documentclass[a,b,c,d,e]{artikel}
```

Die Klasse `base.cls` soll zunächst ausgeben, welche Optionen ihr übergeben wurden:

```
% base.cls
\DeclareOption*{\typeout{Option: \CurrentOption}}
```

Die Datei `artikel.cls` können Sie in verschiedener Weise programmieren. Eine erste Variante

```
% artikel.cls erste Variante
\DeclareOption{a}{\typeout{artikel: a}}
\DeclareOption{b}{\typeout{artikel: b}}
\LoadClass{base}
```

liefert uns die Erkenntnis, daß `artikel.cls` die Optionen `a` und `b` erhalten und verarbeitet hat. Die Optionen `c` bis `e` wurden nicht verwendet. Die Änderung

```
% artikel.cls zweite Variante
\DeclareOption{a}{\typeout{artikel: a}}
\DeclareOption{b}{\typeout{artikel: b}}
\LoadClassWithOptions{base}
```

bewirkt die Auswertung der Optionen `a` und `b` durch `artikel.cls` und die Abarbeitung aller (!) Optionen durch die Klasse `base.cls`, da sie mit genau denselben Optionen (eben allen) wie `artikel.cls` aufgerufen wird. Möchten Sie, daß nur die Optionen weitergereicht werden, die `artikel.cls` nicht auswertet, so führt folgende Variante zum Ziel:

```
% artikel.cls dritte Variante
\DeclareOption{a}{\typeout{artikel: a}}
\DeclareOption{b}{\typeout{artikel: b}}
\DeclareOption*{%
  \PassOptionsToClass{\CurrentOption}{base}}
\LoadClass{base}
```

Sie gibt mit dem `\PassOptionsToClass`-Befehl genau die nicht von `artikel.cls` bearbeiteten Optionen an `base.cls` weiter, im Beispiel also `c` bis `e`. Ersetzen Sie in der letzten Variante `\LoadClass{base}` durch `\LoadClassWithOptions{base}`, so führt dies wiederum zur doppelten Auswertung der Optionen `a` und `b`.

2.4.4 Nützliche Einsprungstellen

$\text{\LaTeX} 2_{\epsilon}$ kennt vier Kommandos, die die übergebenen Befehle speichern und sie zu Beginn oder am Ende des Dokumentes oder am

Ende des Paketes oder der Klassendatei ausführen. Für das Schreiben von Paketen sind folgende Befehle innerhalb von Optionszweigen nützlich:

```
\AtEndOfPackage \AtEndOfPackage{<Befehle>}
\AtEndOfClass   \AtEndOfClass{<Befehle>}
```

Die gesammelten Befehle werden am Ende der Paket- oder Klassendatei ausgeführt. Die Befehle

```
\AtBeginDocument \AtBeginDocument{<Befehle>}
\AtEndDocument   \AtEndDocument{<Befehle>}
```

sammeln ebenfalls die Befehlsfolgen, führen sie jedoch bei dem Kommando `\begin{document}` bzw. `\end{document}` aus. Im letzteren Falle wird der Code ausgeführt, bevor die letzte Seite gesetzt ist. Wegen noch nicht gesetzter Fließobjekte kann daher ein `\clearpage`-Befehl erforderlich sein, falls Ihre Aktionen wirklich zuallerletzt stattfinden müssen.

2.4.5 Fehlermeldungen

Eigene Pakete und Klassen können mit Hilfe der Befehle

```
\ClassError \ClassError{<Klassenname>}{<Fehler>}{<Hilfe>}
\PackageError \PackageError{<Paketname>}{<Fehler>}{<Hilfe>}
```

Fehlermeldungen in L^AT_EX-konformer Weise erzeugen. Sie geben den Fehlertext `<Fehler>` aus und erzeugen den bekannten Errorprompt mit Fragezeichen. Geben Sie nunmehr `h` für »weitere Hilfe« ein, so wird zusätzlich der informative Text `<Hilfe>` ausgegeben, bevor der Errorprompt wieder erscheint.

Möchten Sie keine Fehlermeldung, sondern nur eine informative Mitteilung oder eine Warnung ohne Errorprompt erzeugen, so greifen Sie auf die Befehle

```
\...Warning \ClassWarning{<Name>}{<Warnung>}
\PackageWarning{<Name>}{<Warnung>}
\ClassWarningNoLine{<Name>}{<Warnung>}
\PackageWarningNoLine{<Name>}{<Warnung>}
\ClassInfo{<Name>}{<Info>}
\PackageInfo{<Name>}{<Info>}
```

zurück. In allen Fällen ist `<Name>` der Name der Klasse oder des Paketes. Sie können mit diesen Befehlen den Text `<Warnung>` oder `<Info>` ausgeben. Die beiden `NoLine`-Versionen der Warnbefehle unterdrücken die Ausgabe der Zeilennummer, die bei Benutzung der

ersten beiden Kommandos ausgegeben wird und helfen soll, den fehlerhaften Befehl im Text aufzuspüren. Die Info-Befehle erzeugen Mitteilungen an den Benutzer ohne Angabe der Zeilennummern.

Innerhalb der ausgegebenen Mitteilungen können die folgenden Kommandos zur Textformatierung eingesetzt werden:

- `\Messagebreak` erzeugt einen Zeilenumbruch,
- `\space` druckt ein Leerzeichen,
- `\protect` verhindert die Ausführung des nachfolgenden Befehlswortes und druckt es stattdessen.

2.4.6 Laden von Dateien

Neben dem Laden von fertigen Paketen oder Klassen kann es sinnvoll sein, Hilfs- oder Ergänzungsdateien in den \LaTeX -Quelltext einzufügen. Die erweiterten Befehle

```
\IfFileExists{<Datei>}{<Ja>}{<Nein>}
\InputIfFileExists{<Datei>}{<Ja>}{<Nein>}
```

```
\IfFileExists
\InputIfFileExists
```

überprüfen zunächst, ob die Datei mit dem Namen `<Datei>` vorhanden ist. Wenn dies nicht der Fall ist, werden die Befehle des `<Nein>`-Zweiges ausgeführt, ansonsten die Befehle aus dem `<Ja>`-Zweig. Beim `\InputIfFileExists`-Befehl wird zusätzlich nach Ausführung die angefragte Datei geladen.

2.4.7 Eine Buchklasse als Beispiel

Eine Beispielklasse soll alles bisher im Zusammenhang mit der Verwendung von Klassen Gesagte illustrieren. `buch.cls` soll es gestatten, Bücher in verschiedenen Stilrichtungen zu schreiben, wobei innerhalb der Klassendatei nur einige elementare Einstellungen vorgenommen werden und die eigentliche Layoutgestaltung in separaten Optionsdateien erfolgt. Das Verfahren bietet den Vorteil, daß neue Layouts einfach durch Bereitstellung der entsprechenden Optionsdatei eingebunden werden können, ohne daß die Klassendatei ständig geändert werden muß (\LaTeX verfährt ähnlich, indem es für jeden Hauptstil Optionsdateien bereitstellt, zum Beispiel `bk10.clo` bis `bk12.clo` als Optionen für Schriftgrößen des Stils »book«). Die Optionsdatei soll den Namen der Option tragen und ebenfalls durch die Endung `.clo` gekennzeichnet werden.

Beispielklasse

Innerhalb einer Optionsdatei können Sie alle Elemente modifizieren, die für die betreffende Stilrichtung relevant sind. Dazu zählt unter anderem die Gestaltung der Titelseite, des Inhaltsverzeichnisses, der Gliederungsüberschriften, des Seitenlayouts sowie spezieller

Makros und Umgebungen. Um auch im Stichwortverzeichnis ein einheitliches Erscheinungsbild zu gewährleisten, sollten Sie daran denken, eine entsprechende Stildatei für Ihr bevorzugtes Stichworterstellungsprogramm zu schreiben. Mehr zu dieser Thematik erfahren Sie in Abschnitt 3.10.

Funktionsumfang

Zunächst wird angesprochen, was die Klasse leisten soll, danach wird das Programm vorgestellt, gefolgt von einer Optionsdatei. Das Layout des Inhaltsverzeichnisses sowie der Gliederungsüberschriften wurde bis zur Stufe des `\subsection`-Befehls durchgestaltet und eine Titelseite entworfen, die neben Titel, Datum und Autor auch die Angabe eines Untertitels, eines Verlages und eines Herausgebers ermöglicht. Auf der Rückseite dieser Titelseite können Sie Copyright-Vermerke, ein Impressum und andere Angaben unterbringen. Weiterhin sind noch passende Seitenlayouts enthalten. Das Layout wurde für das Format DIN A4 entworfen, eine Änderung der Parameter `\textheight` oder `\paperwidth` sind möglich, können aber im Einzelfall zu unschönen Resultaten führen. In solchen Fällen müssen Sie gegebenenfalls eine entsprechend bearbeitete Version der Optionsdateien für Ihre Papiergröße erstellen. Beachten Sie, daß die Option `a4paper` nicht automatisch geladen wird, sondern Sie diese Papiergröße explizit angeben müssen – vielleicht möchten Sie ein anderes Papierformat laden. Nicht festlegen müssen Sie dagegen die deutsche Anpassung durch das `german.sty`-Paket, dieses wird automatisch geladen. (Zum Einsatz der neuen deutschen Rechtschreibung müssen Sie im Quelltext in Zeile 37 das Paket `ngerman.sty` laden.) Die Klasse `buch.cls` setzt auf dem Original `book.cls` auf, so daß Ihnen alle Möglichkeiten dieses Stils offen stehen. Das Stichwortverzeichnis wird ohne Angabe entsprechender Optionen oder Befehle automatisch am Ende des Buches erzeugt. Für Werke, an denen mehrere Autoren beteiligt sind, ist bei jedem mit dem `\part`-Befehl erzeugten Werkteil eine Autorenangabe möglich.

Anwendung

Wie schaut nun der Textrahmen für ein Buch aus? Das Beispiel geht davon aus, daß die Beispielloption `celg` wie »colour elegant« heißt. Die Datei stellt Elemente für ein farbig gestaltetes Buch bereit. Für eigene Layouts geben Sie den Namen Ihres Paketes an. Ein typisches Gerüst für ein Buch schaut wie folgt aus:

```
buch.tex
```

```
\documentclass[celg,multi]{buch}
\usepackage[latin1]{inputenc}
\begin{document}

\frontmatter % Angaben für die Titelseite
\author{M. Wurle, H. Durlinger}
\title{Deutschsprachige Lyrik}
```

```

\subtitle{von 1800 bis heute}
\editor{I. Klöckl}
\edition{Erste Auf"|lage}
\publisher{Selbst-Verlag Wien--Budapest}
% v-- zuweilen nützlich --v
\renewcommand{\editorname}{Zusammenstellung}
\maketitle

% die Impressumseite
\vspace*{.5\textheight}
(c) 1996\\
Dieses Buch wurde gesetzt mit \LaTeX{} unter
Verwendung der Option \texttt{celg}. Gedruckt auf
chlorfreiem Papier.

% Vorworte, Widmungen, Danksagungen etc.
\preface[Danksagung]{Danksagung} ...
\preface{Vorwort zur ersten Auf"|lage} ...
\tableofcontents

\mainmatter % los geht's

\partauthor{M. Wurl} % wer hat diesen Teil geschrieben?
\part{Autoren des letzten Jahrhunderts}
...
\partauthor{H. Durlinger} % noch ein Autor
\part{Autoren der Gegenwart}
...
\begin{thebibliography}{9}
...
\end{thebibliography}

\backmatter ...
\end{document}

```

Wie Sie sehen, sind folgende Befehle zur Erzeugung einer Titel- und einer Impressumseite erforderlich:

- ❑ `\author{<Text>}` legt den Namen des Autors fest.
- ❑ `\date{<Text>}` erklärt das Erscheinungsdatum und ist optional. In diesem Falle wird das aktuelle Datum in geeignetem Format gedruckt.
- ❑ `\title{<Text>}`, `\subtitle{<Text>}` bestimmen Titel und (optional) Untertitel, der in kleinerem Schriftgrad erscheint.

- `\editor{<Text>}` (optional) kann benutzt werden, um den Namen des Herausgebers eines Buches oder einer Reihe kenntlich zu machen. Er wird vom Leser durch den Zusatz »Hrsg.« erkannt. Soll der Name eines Übersetzers oder eines Redakteurs genannt werden, so können Sie mit

```
\renewcommand{\editorname}{Übersetzer}
\renewcommand{\editorname}{Red.}
```

dementsprechend Texte einfließen lassen. Die ursprüngliche Definition lautet `\newcommand{\editorname}{Hrsg.}`.

- `\publisher{<Text>}` ist eine optionale Angabe des Verlages nebst Ort.
- Der `\preface`-Befehl wird auf Seite 115 erläutert; die dort vorgestellte Version ist auch in der Buchklasse verfügbar.

Die der Titelseite folgende Rückseite, das Impressum, können Sie frei gestalten. Danksagungen, Widmungen und das Vorwort können mit einem `\chapter*`-Befehl erzeugt werden. Zur Generierung des Stichwortverzeichnisses müssen Sie neben den `\index`-Befehlen keine weiteren Kommandos hinzufügen; es erscheint automatisch in kleinerer Schrift am Ende des Buches.

Im Beispiel wurde neben der Option `celg` noch `multi` angegeben. Diese Option ermöglicht es Ihnen, Bücher zu erstellen, in denen jeder Teil von einem anderen Autor stammt. Vor jedem `\part`-Befehl müssen Sie mit `\partauthor{<Autor>}` den jeweiligen Autor benennen. Er wird dann in der linken (gerade zahligen) Kopfzeile zusammen mit der Kurzfassung der `\part`-Überschrift aufgeführt, während die rechten Kopfzeilen unverändert bleiben. Weiterhin werden die einzelnen Autoren im Inhaltsverzeichnis aufgeführt. Hat ein Autor mehrere Teile geschrieben, können die folgenden `\partauthor`-Kommandos entfallen, bis eine Änderung eintritt.

Die Gestaltung des Stichwortverzeichnisses wird mit der Make-Index-Stildatei `elegant.ist` vorgenommen. Bearbeiten Sie Ihr Buch nach dem bekannten Schema:

```
latex buch          % z.B. mit "elegant"-Option
makeindex -s elegant.ist buch
latex buch
```

Programmierung

Die meisten Funktionen werden aus der Klasse `book.cls` übernommen, auf die `buch.cls` aufsetzt. Es werden lediglich einige von allen

`\partauthor`

Layouts gemeinsam benutzte Makros zentral definiert und dafür sorgt, daß `german.sty` automatisch geladen wird. Die von `buch.cls` bearbeiteten Optionen ersetzen den Inhalt des Makros `\filename` durch den Namen einer Datei, in der die Einzelheiten des betreffenden Layouts enthalten sind. Dieser Dateiname wird am Ende der Klasse benutzt, um die Datei zu laden. Möchten Sie eigene Layouts hinzufügen, müssen Sie eine neue Option mit der zugehörigen Datei bereitstellen. Alle nicht von `buch.cls` bearbeiteten Optionen werden an `book.cls` weitergereicht.

Da durch das Laden der Klasse `book.cls`, die die standardmäßigen Einstellungen festlegt, die eigenen gleich wieder überschrieben würden, muß dafür Sorge getragen werden, daß die Optionsdateien erst später geladen werden. Hierzu läßt sich der `\AtEndOfClass`-Befehl hervorragend einsetzen, der die in `\filename` gespeicherte Datei lädt.

Reihenfolge beachten!

```

1 \NeedsTeXFormat{LaTeX2e}
2 \ProvidesClass{buch}[1996/06/25 I.Kloeckl]
3 \typeout{BUCH.CLS 25/6/96 I. Kloeckl}
4 \RequirePackage{ifthen}
5 %
6 % multi muß VOR allen anderen Optionen deklariert
7 % sein, damit diese das Flag auswerten können!
8 %
9 \newboolean{multi} \setboolean{multi}{false}
10 \DeclareOption{multi}{\setboolean{multi}{true}}
11 \newcommand{\filename}{}
12 % unbekannte Optionen an book.cls weiterreichen
13 \DeclareOption{bel}
14   {\renewcommand{\filename}{bel.clo}}
15 \DeclareOption{elegant}
16   {\renewcommand{\filename}{elegant.clo}}
17 \DeclareOption{tech}
18   {\renewcommand{\filename}{tech.clo}}
19 \DeclareOption{modern}
20   {\renewcommand{\filename}{modern.clo}}
21 \DeclareOption{celg}
22   {\renewcommand{\filename}{celg.clo}}
23 \DeclareOption*
24   {\PassOptionsToClass{\CurrentOption}{book}}
25 %
26 \AtEndOfClass{\input{\filename}}
27 %
28 % Hier Defaultoptionen für BUCH.CLS eintragen

```

buch.cls

```

29 \ExecuteOptions{elegant}
30 \ProcessOptions\relax
31 %
32 % Hier Defaultoptionen für BOOK.CLS eintragen,
33 % dann Klasse laden: \LoadClass[...]{book}
34 \LoadClass{book}
35 %
36 % Hier nun die Defaultpackages laden
37 \RequirePackage{german,makeidx,calc}
38 %
39 % Noch ein paar gemeinsame Dinge tun...
40 \AtEndDocument{\renewcommand{\baselinestretch}{1.0}
41 \addcontentsline{toc}{chapter}{Index}
42 {\small\printindex}}
43 %
44 \newcommand{\@subtitle}{}
45 \newcommand{\subtitle}[1]{\gdef\@subtitle{#1}}
46 \newcommand{\@dateob}{}
47 \newcommand{\dateofbook}[1]{\gdef\@dateob{#1}}
48 \newcommand{\@edition}{}
49 \newcommand{\edition}[1]{\gdef\@edition{#1}}
50 \newcommand{\@editor}{}
51 \newcommand{\editor}[1]{\gdef\@editor{#1}}
52 \newcommand{\@verlag}{}
53 \newcommand{\publisher}[1]{\gdef\@verlag{#1}}
54 \newcommand{\@mauthor}{}
55 \newcommand{\partauthor}[1]{\gdef\@mauthor{#1}}
56 \newcommand{\editorname}{Hrsg.}
57 %
58 \providecommand{\prefacename}{Vorwort}
59 \newcommand{\preface}[2][\prefacename]
60 {\chapter*{#2}\markboth{#1}{#1}}
61 %
62 \makeindex

```

Die einzelnen Optionsdateien `*.clo` enthalten Modifikationen interner L^AT_EX-Strukturen, die ein einheitliches Layout aller betroffenen Gestaltungselemente sicherstellen und die in Kapitel 3 im einzelnen erläutert werden. Sie finden nach Funktionen gegliedert und eventuell an ein spezielles Layout angepaßt, Beispielprogramme zur Änderung des Inhaltsverzeichnisses, der Gliederungsüberschriften und der Titelseite. Da die Option `celg` die Erzeugung farbiger Bücher gestattet, muß das `color`-Paket, das in Abschnitt 4.8.2 näher vorgestellt wird, geladen werden.

```

1 % CELG.CLO
2 \typeout{BUCH class option CELG.CLO}
3 \RequirePackage[dvips]{color}
4 %
5 % ----- Titelseite -----
6 \renewcommand{\maketitle}
7  {{\thispagestyle{empty}\setlength{\parindent}{0pt}
8   \vspace*{1cm}%
9   \@author
10  \ifthenelse{\equal{\@editor}{}}
11   {}
12   {\newline\@editor{}} (\@editorname)}
13  \par
14  \vspace*{0.2\textheight}
15  \hrule height 1pt\vspace{\baselineskip}
16  {\Huge\bfseries\@title\par}%
17  \ifthenelse{\equal{\@subtitle}{}}
18   {}%
19   {\Large\@subtitle\par}
20  \vspace*{0.5\baselineskip}\hrule height 1pt
21  \vfill
22  \ifthenelse{\equal{\@edition}{}}
23   {}
24   {\@edition{}} }
25  \ifthenelse{\equal{\@dateob}{}}
26   {\the\year}
27   {\@dateob}
28  \\
29  \ifthenelse{\equal{\@verlag}{}}
30   {}
31   {\@verlag}
32  \newpage\thispagestyle{empty}%
33  }}
34 %
35 % ----- TOC -----
36 \newlength{\emlength}
37 \setlength{\emlength}{1em}
38 \newlength{\numwidth}
39 \setlength{\numwidth}{2\emlength}
40 \renewcommand{\numberline}[1]
41  {\makebox[\numwidth][l]{#1}\hspace{0.5\emlength}}
42
43 \renewcommand{\l@part}[2]{\pagebreak[3]%
44  \addvspace{3\baselineskip}%

```

celg.clo

```

45 \noindent\colorbox{YellowRed}
46 {\parbox[c][2cm][c]{\textwidth}
47   {\Large\bfseries\sffamily #1}}%
48 \par\nopagebreak
49 \addvspace{0.5\baselineskip}}
50 \renewcommand{\l@chapter}[2]
51 {\addvspace{2\baselineskip}\pagebreak[3]
52 \noindent
53 \colorbox{Yellow}{\makebox[\textwidth][1]{%
54   \large\rule[-1em]{0mm}{2.5em}%
55   \bfseries#1\quad#2}}
56 \par\nopagebreak\addvspace{0.5\baselineskip}}
57 \renewcommand{\l@section}[2]
58 {\addvspace{0.5\baselineskip}
59 \noindent\hspace{2.5\emlength}%
60 {\bfseries #1\quad#2}\par\nopagebreak[2]}
61 \renewcommand{\l@subsection}[2]
62 {{\setlength{\numwidth}{2.5\emlength}
63 \noindent\hspace{5\emlength}#1\quad#2\par}}
64 %
65 % ----- Gliederung -----
66 \def\@part[#1]#2
67 {\ifthenelse{\value{secnumdepth}>\m@ne}
68   {\refstepcounter{part}
69   \addcontentsline{toc}{part}{%
70     \protect\numberline{\thepart}#1}}
71   {\addcontentsline{toc}{part}{#1}}
72 \vspace*{0.2\textheight}
73 \noindent\colorbox{YellowRed}%
74 {\parbox[c][3cm][c]{\textwidth}
75   {\LARGE\bfseries\partname~\thepart\[\[1ex]
76   \Huge#2}
77 }
78 \ifthenelse{\boolean{multi}}
79   {\vspace*{\baselineskip}\noindent von \@mauthor
80   \addtocontents{toc}
81   {\vspace{\baselineskip}\noindent\hspace{2.5em}%
82   von \@mauthor
83   \protect\nopagebreak%
84   \vskip1\baselineskip\vskip1\baselineskip}}
85   {}%
86 \partmark{#1}%
87 \@endpart}
88 \renewcommand{\@spart}[1]

```

```

89   {\vspace*{0.2\textheight}
90    \noindent\colorbox{YellowRed}%
91    {\parbox[c][3cm][c]{\textwidth}{\Huge\bfseries#1}}
92    \ifthenelse{\boolean{multi}}
93     {\vspace*{1em}\noindent von \@author}
94     {}
95    \@endpart}
96 \renewcommand{\@makechapterhead}[1]
97   {\vspace*{3\baselineskip}
98    \noindent\colorbox{Yellow}
99    {\hbox to \textwidth{\Large%
100     \rule[-\baselineskip]{0mm}{2.5\baselineskip}%
101     \bfseries\thechapter\quad #1\hss}}
102    \vskip1\baselineskip}
103 \renewcommand{\@makeschapterhead}[1]
104   {\vspace*{3\baselineskip}
105    \noindent\colorbox{Yellow}
106    {\hbox to \textwidth{\Large%
107     \rule[-\baselineskip]{0mm}{2.5\baselineskip}%
108     \bfseries#1\hss}}\vskip1\baselineskip}
109 \renewcommand{\section}
110   {\@startsection{section}{1}
111    {\z@}{-3ex plus -1ex minus-.2ex}
112    {2ex plus.2ex}{\reset@font\normalsize\bfseries}}
113 \renewcommand{\subsection}
114   {\@startsection{subsection}{2}{\z@}
115    {-2ex plus -1ex minus-.2ex}{2ex plus.2ex}
116    {\reset@font\normalsize\itshape\bfseries}}
117 %
118 % ----- Seitenstile -----
119 \newcommand{\partmark}[1]{\}
120 \newcommand{\ps@headline}{%
121   \let\@mkboth\markboth
122   \ifthenelse{\boolean{multi}}
123    {\renewcommand{\partmark}[1]
124     {\markboth{\@author~--~#1}{}}
125     \renewcommand{\chaptermark}[1]{\}
126     }
127    {\renewcommand{\partmark}[1]{\markboth{}}
128     \renewcommand{\chaptermark}[1]
129     {\markboth{\@chapapp~\thechapter~--~#1}{}}
130     }
131 \renewcommand{\sectionmark}[1]
132   {\markright{\thesection~--~#1}}

```

```

133 \renewcommand{\subsectionmark}[1]{
134 \renewcommand{@oddfoot}{}
135 \renewcommand{@evenfoot}{}
136 \renewcommand{@oddhead}
137   {\normalfont{\small\sffamily\rightmark} \hrulefill%
138    \colorbox{LightRed}%
139    {\parbox{2cm}{\centering\thepage}}}
140 \renewcommand{@evenhead}
141   {\colorbox{LightRed}%
142    {\normalfont\parbox{2cm}{\centering\thepage}}}
143    \hrulefill{ \small\sffamily\leftmark}}
144 }
145 \pagestyle{headline}
146 %
147 % ----- Optionsspezifika -----
148 \definecolor{Yellow}{rgb}{1,1,0}
149 \definecolor{YellowRed}{rgb}{1,0.6,0}
150 \definecolor{LightRed}{rgb}{1,0.8,0.5}
151 % rosa unterlegte Legenden
152 \renewcommand{@makecaption}[2]
153   {\colorbox{LightRed}{\parbox{\linewidth}
154    {\textbf{#1 } #2}}}

```

Der Aufbau der Stichwort-Stildatei `elegant.ist` wird in Abschnitt 3.10 detailliert erläutert:

`elegant.ist`

```

1 % ELEGANT.IST
2 % Indexeintraege höherer Ordnung durch eine
3 % ---, -Kombination abgetrennt.
4 headings_flag 1
5 heading_prefix "\\hrulefill\\ "
6 heading_suffix "\\ \\hrulefill"
7
8 preamble "\\begin{theindex}\n
9           \\def\\subitem{\\par\\leavevmode
10            \\hangindent40pt
11             \\makebox[12pt][c]{\\hss---\\hss},
12             \\hspace{8pt} }
13           \\def\\subsubitem{\\par\\leavevmode
14            \\hangindent40pt
15             \\makebox[12pt][c]{\\hss---\\hss},
16             \\makebox[12pt][c]{\\hss---\\hss},
17             \\hspace{8pt} }"

```