



Otto-Friedrich-Universität Bamberg
Lehrstuhl für Praktische Informatik



Ausarbeitung

Im Rahmen des Bachelor|Master-Seminars

Thema des Seminars

Zum Thema:

Thema Alles LaTeX Oder Was ?

Vorgelegt von:

Autor

Betreuer: Prof. Dr. Guido Wirtz

Bamberg, Wintersemester 2013

Inhaltsverzeichnis

1	Einleitung	1
2	Genereller Aufbau	1
2.1	Erstellung einer Titelseite	1
2.2	Einbindung der einzelnen Kapitel	2
2.3	Einbindung des Literaturverzeichnisses	2
3	Einbindung von Grafiken	2
3.1	Grafiken zwischen Absätzen	3
3.2	Von Text umflossene Grafiken	3
4	Einbindung von Java-Quellcode	4
5	Nutzung des Abkürzungsverzeichnisses	5
	Literaturverzeichnis	6

Abbildungsverzeichnis

1	DSG-Logo	3
2	DSG-Logo	3

Tabellenverzeichnis

Listings

1	Das ist das erste Listing	4
2	Das ist das zweite Listing	5

Abkürzungsverzeichnis

DSG Distributed Systems Group

1 Einleitung

Zur Verfassung von Seminar-, Bachelor- und Masterarbeiten bietet die Distributed Systems Group neben den Word-Vorlagen auch Vorlagen für Latex. In diesem Dokument soll kurz gezeigt, wie die einzelnen Umgebungen und Befehle der Vorlage sinnvoll zur Erstellung von Arbeiten genutzt werden können. Es erfolgt aber keine allgemeine Anleitung zur Erstellung von Dokumenten mit L^AT_EX. Als Einstiegsliteratur, die aber für die allgemeine Arbeit mit L^AT_EX durchaus ausreichend ist, eignet sich das Buch von Kopka [1]. Eines der älteren Werke des Autors reicht ebenfalls aus. Eine sehr gute Übersicht über alle wichtigen Szenarien gibt das Wikibook „L^AT_EX“ [2]. Weitere hilfreiche Lektüre findet sich unter [3], [4], [5], [6] und allen verfügbaren Suchmaschinen.

Die Erstellung eines L^AT_EX-Dokumentes erfolgt z.B. über die Anweisung `pdflatex seminar` oder mit einer entsprechenden IDE, wie *TeXnicCenter*¹ oder *TeXlipse*². Hier empfiehlt sich auch das Zusammenspiel mit SumatraPDF³, das eine Vorwärts- und Rückwärtssuche in den Dokumenten ermöglicht.

2 Genereller Aufbau

Die zentrale Datei der Vorlage ist die Datei `seminar.tex`, die vom Autor für jeweilige Arbeit anzupassen ist. In dieser Datei können die Einstellungen für die Titelseite vorgenommen werden. Des Weiteren müssen in dieser Datei die einzelnen Kapitel der Arbeit, sowie die `.bib`-Datei mit den *bibtex*-Einträgen eingebunden werden.

2.1 Erstellung einer Titelseite

Die Erstellung der Titelseite erfolgt über den Befehl `\maketitle`. Die Anzahl der Parameter, mit der dieser Befehl aufgerufen wird, hängt davon ab, ob eine Seminararbeit oder eine Bachelor-/Masterarbeit erstellt werden soll. Zur Erstellung einer Titelseite für eine Seminararbeit muss der Befehl `\maketitle` mit sechs Parametern aufgerufen werden.

```
\maketitle{Thema des Seminars}{Thema der Ausarbeitung}%  
{Autor}{Betreuer}{Semester}{Bachelor|Master}
```

Die einzelnen Parameter sind selbsterklärend. Falls die Seminararbeit von mehreren Autoren verfasst wird, sind diese durch `,\` zu trennen. Bei drei Autoren ist also z.B. die Zeichenkette `Hans Meier,\ Peter Müller und\ Hans Müller` als dritten Parameter einzusetzen. Als Semesterangabe sollte ein Kürzel wie `SoSe13` übergeben werden.

Um die Titelseite einer Abschlussarbeit zu erstellen, ist der Befehl `\maketitle` mit drei Parametern aufzurufen.

¹<http://www.texniccenter.org>

²<http://texlipse.sourceforge.net>

³<http://blog.kowalczyk.info/software/sumatrapdf/free-pdf-reader-de.html>

```
\maketitle{Bachelor-/Masterarbeit}{Studiengang}{Thema der Arbeit}%  
{Autor}{Abgabedatum}
```

Auch in diesem Fall sind die Parameter selbsterklärend.

2.2 Einbindung der einzelnen Kapitel

Um die Übersichtlichkeit der erstellten Arbeit zu gewährleisten, ist es sinnvoll, für jedes Kapitel eine einzelne *.tex*-Datei zu erstellen. Damit unter Verwendung dieser Dateien ein einzelnes Dokument erstellt wird, sind die einzelnen Kapitel in der Datei `seminar.tex` einzubinden. Dies erfolgt über den Befehl `\input`. Soll z.B. ein Kapitel eingefügt werden, dass in der Datei `kapitel-1.tex` enthalten ist, so geschieht dies durch die folgende Anweisung.

```
\input{kapitel-1}
```

Diese Anweisung ist an der in der Datei `seminar.tex` markierten Stelle einzufügen.

2.3 Einbindung des Literaturverzeichnisses

Die einzelnen Literaturquellen sind in einer *.bib*-Datei aufzuführen. Die *bibtex*-Datei `example.bib` enthält einige Beispiele, wie verschiedenen Quellen wie Buch, Artikel usw. in dieser Datei aufzuführen sind. Jeder der Einträge benötigt zur Referenzierung ein eindeutiges Label, das sich aus den ersten drei Buchstaben des Nachnamen des Autors und den letzten beiden Ziffern der Jahreszahl der Veröffentlichung ergeben. Sind mehrere Autoren bei einer Literaturquelle aufgeführt, so ergibt sich der Label aus den Anfangsbuchstaben der Nachnamen der ersten drei Autoren und den letzten beiden Ziffern der Jahreszahl der Veröffentlichung. Falls ein Autor in einem Jahr mehrere Werke veröffentlicht hat, so sind die weiteren Label mit kleinen Buchstaben, beginnend bei *b*, zu ergänzen.

Eine *.bib*-Datei ist mit Hilfe der Anweisung `\bibliography` in der Datei `seminar.tex` einzubinden. Die hier verwendete Datei `references.bib` wurde also durch folgende Anweisung in dieses Dokument eingebunden.

```
\bibliography{references}
```

3 Einbindung von Grafiken

Im Allgemeinen werden Grafiken zwischen Absätzen eingefügt. Eine andere Möglichkeit ist es, die Grafiken von Text umfließen zu lassen. Generell gilt aber, dass die Grafiken als *pdf*- oder *eps*-Dateien vorliegen sollten. Dies hängt davon ab, ob *latex* oder *pdflatex* zur Erzeugung des Dokumentes verwendet wird. Wird *latex* zur Erzeugung genutzt, so können nur *eps*-Dateien eingebunden werden. Wird das Dokument mit *pdflatex* erstellt, so sind *pdf*-Dateien zu verwenden.

Grafiken müssen generell im *images*-Verzeichnis vorhanden sein, um sie erfolgreich einbinden zu können. Selbst erstellte Grafiken müssen also in das *images*-Verzeichnis kopiert werden.

3.1 Grafiken zwischen Absätzen

Eine Einbindung von Grafiken zwischen Absätzen erfolgt mit dem Befehl `\asfigure`.

```
\asfigure{bild1}{vs-logo}{DSG-Logo}{5}
```

Der erste Parameter `bild1` dient als Label für Referenzen auf die Grafik. Der zweite Parameter ist der Name der Grafikdatei ohne Endung. Als dritten Parameter ist dem Befehl `\asfigure` die Beschriftung der Abbildung zu übergeben. Der letzte Parameter gibt die Breite der Grafik in cm an. Die Größenangabe darf aber nur als Zahl erfolgen. Ist eine Grafik breiter als der übergebene Parameter, so wird sie verkleinert, kleinere Grafiken werden entsprechend vergrößert. Der obige Befehl führt zu der folgenden Abbildung 1.

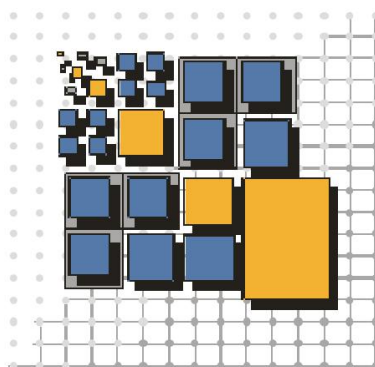
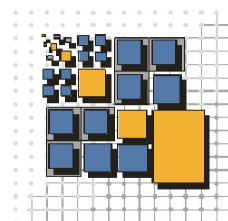


Abbildung 1: DSG-Logo

3.2 Von Text umflossene Grafiken

Sollen Grafiken von Text umflossen werden, so ist der Befehl `\textflow` zu benutzen. Dieser Befehl basiert auf dem Paket *wrapfig*, dass u.U. nachinstalliert werden muss. Dieses Paket ist, wie alle anderen Umgebungen mit dieser Funktionalität, etwas beschränkt und zum Teil fehlerhaft, deshalb ist die Nutzung nur unter Vorbehalt empfohlen. Eine Grafik kann wie folgt eingebunden werden.

```
\textflow{r}{3}{vs-logo}{DSG-Logo}{bild2}
```



Nach diesem Befehl folgt der Text, der das eingebundene Bild umfließen soll. Für den ersten Parameter kommen als einzig relevante Möglichkeiten die kleinen Buchstaben *l* und *r* in Frage. Diese sorgen für eine Anordnung der Grafik auf der linken, bzw. rechten Seite. Mit Hilfe des zweiten Parameters wird die gewünschte Breite der Grafik in cm angegeben, die entsprechend skaliert wird. Der dritte Parameter ist der Name der Grafikdatei ohne

Abbildung 2:
DSG-Logo

Endung. Der vierte Parameter bestimmt die Bezeichnung der Grafik. Durch den fünften Parameter kann ein Label zur Referenzierung angegeben werden.

Eine Eigenheit dieses Pakets ist es, dass Grafiken immer am Anfang eines Absatzes eingefügt werden. Soll eine Grafik von einem Absatz komplett umschlossen werden, wie dies bei der Abbildung 2 der Fall ist, so ist der `\fließtext`-Befehl nicht vor dem Absatz aufzuführen, sondern in den Text zu integrieren. Der obige `\fließtext`-Befehl ist demnach hinter dem Wort „Dieses“ in den Quelltext eingefügt worden.

4 Einbindung von Java-Quellcode

Zur Einbindung von Java-Quellcode ist das Paket *listings* notwendig, dass u.U. nachinstalliert werden muss. Dieses Paket ermöglicht es, dass Quellcode von verschiedenen Sprachen auf eine übersichtliche Weise in das Dokument eingebunden werden kann. Die Einbindung von Java-Quellcode kann auf zwei Arten erfolgen: zum einen kann der Java-Quellcode im Quelltext des Dokumentes aufgelistet werden, zum anderen kann der Quellcode aus einer Datei eingelesen werden.

Die Einbindung von Quellcode, der im Quelltext aufgeführt ist, muss über die Nutzung einer Umgebung geschehen. Dies soll an einem kleinen Beispiel gezeigt werden.

```
\begin{javacode}{Das ist das erste Listing}{listing1}
public class Test{

    public Test(){
        // mach was
    }

}
\end{javacode}
```

Zunächst wird eine Umgebung mit der Bezeichnung `javacode` erstellt. Als Parameter wird die Überschrift, sowie das Label übergeben. In der Umgebung wird der Java-Quellcode in den Quelltext des Dokumentes eingefügt und anschließend die Umgebung wieder beendet. Der obige Code erzeugt das folgende Listing 1.

Listing 1: Das ist das erste Listing

```
1 public class Test {
2
3     public Test() {
4         // mach was
5     }
6
7 }
```

Alternativ kann Quellcode auch aus einer Datei eingebunden werden. Dies ist vorteilhaft, weil die Übersichtlichkeit des Quelltextes verbessert wird. Die Einbindung einer Quellcode-Datei erfolgt über den Befehl `\javafile`. Damit die Datei erfolgreich eingebunden werden kann, muss sie im Verzeichnis `src` liegen. Der Befehl kann wie folgt genutzt werden.

```
\javafile{src.java}{Das ist das zweite Listing}{listing2}
```

Als erster Parameter muss der Name der Datei (mit Endung) im *src*-Verzeichnis übergeben werden. Der zweite Parameter bestimmt die Überschrift des Listings. Der dritte Parameter dient zur Referenzierung. Es wird der gleiche Quelltext erstellt, wie in Listing 2 zu sehen ist.

Listing 2: Das ist das zweite Listing

```
1 public class Test{  
2  
3     public Test(){  
4         // mach was  
5     }  
6  
7 }
```

5 Nutzung des Abkürzungsverzeichnisses

Zur Einbindung des Abkürzungsverzeichnisses wird das `acronym`-Package⁴ verwendet. Der Eintrag aller verwendeten Abkürzungen erfolgt in der Datei `abbreviations.tex`, die bspw. den folgenden Inhalt hat:

```
\begin{acronym}[LSPI]  
  \acro{DSG}{Distributed Systems Group}  
  \acro{LSPI}{Lehrstuhl für Praktische Informatik}  
\end{acronym}
```

Die geklammerte Abkürzung [LSPI] sollte dabei durch die längste vorhandene Abkürzung ersetzt werden. Eine Abkürzung wird mit dem Befehl `\acro` innerhalb der `acronym`-Umgebung definiert. Standardmäßig werden nur die im Text tatsächlich verwendeten Abkürzungen auch im Verzeichnis ausgegeben. Um eine Abkürzung innerhalb eines Textes einzufügen genügt der folgende Ausdruck:

```
\ac{<acronym>}
```

Bei der ersten Verwendung innerhalb des Textes bewirkt die Anweisung, dass der vollständige Name gefolgt von der Abkürzung in Klammern ausgegeben wird. Bei jeder weiteren Verwendung wird lediglich die Kurzform ausgegeben. Die Anweisung `\ac{DSG}` erzeugt also zunächst die Langform „Distributed Systems Group (DSG)“. Anschließend nur noch die Abkürzung „DSG“.

⁴<http://www.ctan.org/tex-archive/macros/latex/contrib/acronym>

Literatur

- [1] H. Kopka, *Latex I. Einführung*. Pearson Studium, 2002.
- [2] “LaTeX - Wikibooks,” <http://en.wikibooks.org/wiki/LaTeX>, 2012.
- [3] A. Jürgens, “LaTeX - eine Einführung und ein bißchen mehr...” <ftp://ftp.fernuni-hagen.de/pub/pdf/urz-broschueren/broschueren/a0260911.pdf>, 2000.
- [4] M. Kühner, “LaTeX-Einführung für Doktoranden bzw. Ingenieure,” http://bipede.de/Downloads/LaTeX-Kurs/Kurs-Folien_2011-11-17.pdf, 2011.
- [5] A. Jürgens, “LaTeX - Fortgeschrittene Anwendungen,” <ftp://ftp.fernuni-hagen.de/pub/pdf/urz-broschueren/broschueren/a0279510.pdf>, 1995.
- [6] T. Erbsland and A. Nitsch, “Diplomarbeit mit LaTeX,” <http://drzoom.ch/diplomarbeit-mit-latex.html>, 2009.