

- und `theorempostskipamount` auf 0 pt gesetzt werden. Diese Option ist natürlich nur sinnvoll, wenn die Umgebung `mdframed` mit einer `theorem`-Umgebung kombiniert wird.
- defaultunit:** Die Option `defaultunit` erwartet als Eingabe eine L^AT_EX-spezifische Längeneinheit. Auf die Anwendung werde ich im nachstehenden Abschnitt *Besonderheiten* eingehen. Vorab sei die deutsche Bedeutung angegeben: Basiseinheit.
- xcolor:** Wie bereits angedeutet, wird standardmäßig das Paket `color` verwendet. Viele Anwender nutzen sinnvollerweise das Paket `xcolor`, um bspw. zusätzliche Farbanweisungen nutzen zu können. Bei Angabe der Option `xcolor` werden die Argumente direkt an das Paket `xcolor` übergeben, das gleichzeitig geladen wird.
- style:** Seit der Version 0.4 kann der Anwender zwischen verschiedenen Erstellungsmethoden wählen. Derzeit sind zwei implementiert, die mittels `style=0` und `style=1` ausgewählt werden können. Im Abschnitt *Beispiele* werden die Unterschiede verdeutlicht. Vorab sei angemerkt, dass mit der Standardeinstellung `style=0` die Rahmen mittels `\vrule` und `\rule` gezeichnet werden, wohingegen `style=1` das Paket `tikz` lädt, das dann zum Erstellen der Rahmen verwendet wird.

Die nachstehenden Optionen funktionieren nur bei der Auswahl `style=1`. Wichtig zu wissen ist, dass die Grundeinstellung eines `tikz`-Rahmens auf einer doppelten Umrahmung beruht.

- roundcorner:** Rundungsstärke der Ecken mit der Standardeinstellung 0 pt.
- innerlinewidth:** Linienstärke des inneren Rahmens mit der Standardeinstellung 0 pt.
- middlelinewidth:** Linienstärke des mittleren Rahmens mit der Standardeinstellung `linewidth`.
- outerlinewidth:** Linienstärke des äußeren Rahmens mit der Standardeinstellung 0 pt.
- innerlinecolor:** Linienfarbe des inneren Rahmens mit der Standardeinstellung `linecolor`.
- middlelinecolor:** Linienfarbe des mittleren Rahmens mit der Standardeinstellung `backgroundcolor`.
- outerlinecolor:** Linienfarbe des äußeren Rahmens mit der Standardeinstellung `linecolor`.

Besonderheiten

Die Optionen `xcolor` und `style` sind nur global beim Laden des Paketes anzugeben. Werden diese Optionen lokal gesetzt, erzeugt dies Fehler im \LaTeX -Lauf.

Die Optionen, die eine Länge als Eingabe erwarten, besitzen die Besonderheit, dass die Einheit weggelassen werden kann. Wird bspw. `linewidth=2` als Option angegeben, so ergänzt das Paket die Einheit mit der aktuellen Einheit. Das bedeutet, dass im Standardfall die Eingabe `linewidth=2` zu `linewidth=2cm` ergänzt wird. Wird hingegen `linewidth=2mm` verwendet, so wird `defaultunit` ignoriert.

Beispiele

Ausgangspunkt ist in allen Fällen das Laden des Paketes mit der Option `style=1`. Für die Umsetzung der Idee in `tikz` hat Elke Schubert wesentliche Beiträge geliefert.

Beispiel 1

Das erste Beispiel wird wie folgt erzeugt:

```
\begin{mdframed}[linewidth=2pt]
Zur Verdeutlichung der Paketoptionen soll dieses Beispiel dienen.
\end{mdframed}
```

Zur Verdeutlichung der Paketoptionen soll dieses Beispiel dienen.

Beispiel 2

Das zweite Beispiel soll viele Möglichkeiten des Paketes auf einmal demonstrieren. Hierbei soll die Typographie nicht im Vordergrund stehen.

```
\begin{mdframed}[leftmargin=1cm,rightmargin=1cm,fontcolor=black!70,%
innerlinewidth=1.5pt,middlelinewidth=3pt,outerlinewidth=4.5pt←
,%
innerlinecolor=black!40,middlelinecolor=black!70,%
outerlinecolor=black,backgroundcolor=black!10,%
roundcorner=10pt,margin=3cm]
Zur Verdeutlichung der Paketoptionen soll dieses Beispiel dienen.
\end{mdframed}
```

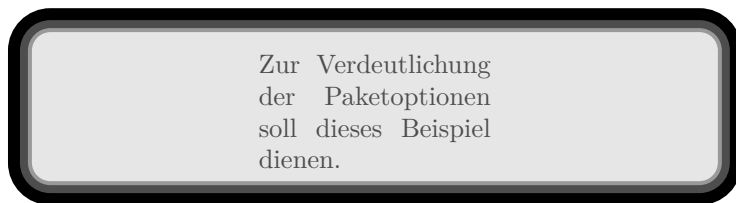


Abbildung 1: Ausgabe des vorstehenden Beispiels.

An den gezeigten Beispielen wird deutlich, welche Möglichkeiten dem Anwender zur Verfügung stehen. Allerdings sind die Möglichkeiten in einigen Bereichen beschränkt, was mich zum nächsten Abschnitt führt.

To-do-Liste

Derzeit ist es nicht möglich, `mdframed` innerhalb einer `minipage`-Umgebung anzuwenden. Die Nutzung des Paketes bei der Option `twoside`, sowie die Kombination mit `multicolumn` sind fehlerhaft. Außerdem ist das Paket zur Zeit inkompatibel mit dem Paket `gmverb`.

In Tabellen rechnen mit `spreadtab`

Uwe Ziegenhagen

Mit einer zu gängigen Tabellenkalkulationen ähnlicher Syntax bietet das `spreadtab` [2] Paket von Christian Tellechea die Möglichkeit, in Tabellen einfache Berechnungen auszuführen. In diesem Artikel wird das Paket kurz vorgestellt und anhand eines Beispiels gezeigt, wie man in \LaTeX einfache Rechnungen erstellen kann.

Einleitung

Eine `spreadtab`-Umgebung benötigt als Parameter in der geschweiften Klammer den Tabellenkopf so, wie er auch in der »normalen« Tabelle gesetzt werden würde, in unserem Beispiel also `{{tabular}{rr|r}}`. Laut Handbuch ist `spreadtab` dabei zu *allen* tabellenartigen Umgebungen kompatibel, in diesem Artikel beschränke ich mich aber auf die `tabular`-Umgebung. In die

einzelnen Zellen schreibt man die Zahlen, wobei man über das auch in Tabellenkalkulationen gängige Adressierungsformat auf einzelne Zellen zugreifen kann. So errechnet beispielsweise **a1+b1** die Summe der Zellen in der ersten und zweiten Spalte der ersten Zeile.

11	22	33
33	44	77
44	66	

```
\begin{spreadtab}%
{{tabular}{rr|r}}
11 & 22 & a1+b1 \\
33 & 44 & a2+b2 \\ \hline
a1+a2 & b1+b2 & \\
\end{spreadtab}
```

Eine Rechnung mit *spreadtab*

Ausgehend von unserem Minimalbeispiel werden wir jetzt schrittweise ein komplexes Beispiel aufbauen, das auch genutzt werden könnte, um aus einem Warenwirtschaftssystem oder einer Datenbank heraus Dokumente zu erzeugen.

Relative Adressierung

Im ersten Schritt wird die absolute Adressierung der einzelnen Zellen aufgelöst und durch eine relative Adressierung ersetzt. Dies hat den Vorteil, dass nur noch die eigentlichen Zahlen in das Dokument eingetragen werden müssen. Die Syntax lautet [**<X-Verschiebung>**,**<Y-Verschiebung>**], wobei die Verschiebung immer von der aktuellen Zelle aus gilt. Beispielsweise addiert die Funktion **[-1,-1]+[1,2]**, die Zelle, die sich links oberhalb der aktuellen Zelle befindet zu der Zelle, die sich eine Zelle rechts und zwei Zellen unter der aktuellen Zelle befindet. Unser Beispiel ändert sich daher zu

11	22	33
33	44	77
44	66	

```
\begin{spreadtab}%
{{tabular}{rr|r}}
11 & 22 & [-2,0]+[-1,0] \\
33 & 44 & [-2,0]+[-1,0] \\ \hline
[0,-2]+[0,-1] & [0,-2]+[0,-1] & \\
\end{spreadtab}
```

Funktionen

Da es insbesondere für die vertikalen Summen schnell unübersichtlich werden würde, wenn mehr Zeilen dazu kommen, ist es an der Zeit, die Summenfunktion **sum(<Zellbereich>)** von *spreadtab* zu nutzen. Der Parameter **<Zellbereich>** kann dabei sowohl absolut als auch relativ oder als Mischung der beiden Adressierungsarten angegeben werden. Das folgende Beispiel zeigt die verschiedenen Arten:

11	22	33
33	44	77
55	66	121
99	132	231

```
\begin{spreadtab}%
{{tabular}{rr|r}}
11 & 22 & [-2,0]+[-1,0] \\
33 & 44 & [-2,0]+[-1,0] \\
55 & 66 & [-2,0]+[-1,0] \\ \hline
sum([0,-3]:[0,-1]) & sum(b1:b3)
& sum(c1:[0,-1]) \\
\end{spreadtab}
```

Der Vorteil der gemischten Adressierung ist, dass nicht bekannt sein muss, über wieviele Zeilen oder Spalten hinweg summiert wird.

Export von Zellwerten

Oftmals ist es auch erforderlich, das Ergebnis einer Rechnung, wie z. B. die Gesamtsumme außerhalb der Tabelle auszugeben. Zu diesem Zweck kann man als optionaler Parameter der Umgebung der Befehl `[\STsavecell <Befehlsname> <absoluter Bezug>]` genutzt werden. `<Befehlsname>` ist dabei eine beliebige Befehlssequenz: Es sollten aber keine existierenden L^AT_EX/T_EX-Kommandos überschrieben werden.

11	22	33
33	44	77
55	66	121
99	132	231

Die Gesamtsumme lautet 231.

```
\begin{spreadtab}%
[\STsavecell{\myresult}{c4}]%
{{tabular}{rr|r}}
11 & 22 & [-2,0]+[-1,0] \\
33 & 44 & [-2,0]+[-1,0] \\
55 & 66 & [-2,0]+[-1,0] \\ \hline
sum([0,-3]:[0,-1]) & sum(b1:b3)
& sum(c1:[0,-1]) \\
\end{spreadtab}
```

Die Gesamtsumme lautet \myresult.

Reine Textzellen und gemischte Zellinhalte

Spaltenüberschriften und andere Zellen, die nicht in die Berechnungen von *spreadtab* einfließen sollen, können durch ein führendes @ markiert werden.

11	22	33
33	44	77
55	66	121
Summe	132	

Die Gesamtsumme lautet 132.

```
\begin{spreadtab}%
[\STsavecell{\myresult}{c4}]%
{{tabular}{rr|r}}
11 & 22 & [-2,0]+[-1,0] \\
33 & 44 & [-2,0]+[-1,0] \\
55 & 66 & [-2,0]+[-1,0] \\ \hline
\multicolumn{2}{r|}{@Summe}
& sum(b1:b3)
\end{spreadtab}
```

Die Gesamtsumme lautet \myresult.

Für Zellen mit gemischten Inhalten stellt das **spreadtab**-Paket die folgende Syntax bereit:

<Text1> :=<Berechnung><Text2>

<Text1> und **<Text2>** sind dabei beliebige Textteile oder Währungssymbole, innerhalb von **:=<Berechnung>** werden die Formeln angegeben.

Foo	Bar	
11	22	33
33	44	77
55	66	121
Summe:		66 \$

```
\begin{spreadtab}%
{{tabular}{rr|r}}
@Foo & @Bar \\ \hline
11 & 22 & [-2,0]+[-1,0] \\
33 & 44 & [-2,0]+[-1,0] \\
55 & 66 & [-2,0]+[-1,0] \\ \hline
\multicolumn{3}{r}%
{Summe: :=\sum(b1:b3)} \$}
\end{spreadtab}
```

Dezimalzeichen und Rundung

Bislang haben wir in den Beispielen nur ganze Zahlen genutzt. Beim Gebrauch von Dezimalzahlen fällt jedoch auf, dass Nachkomma-Nullen nicht ausgegeben werden und als Trennzeichen standardmäßig der Punkt genutzt wird.

1	3,3	4,3
2,2	4,4	6,6
3,2	7,7	

```
\begin{spreadtab}%
{{tabular}{rr|r}}
1.0 & 3.3 & a1+b1 \\
2.2 & 4.4 & a2+b2 \\ \hline
a1+a2 & b1+b2 & \\
\end{spreadtab}
```

Um diese Unschönheiten zu beheben, nutzen wir das Paket **numprint** [1] und die **spreadtab**-Anweisung **\STsetdecimalsep**,. Der Unterschied zwischen den neuen Spaltentypen **N** und **n** besteht darin, dass **N** den Textmodus nutzt, während **n** die Spalte im mathematischen Modus ausgibt.

1,00	3,30	4,30
2,20	4,40	6,60
3,20	7,70	

```
\STsetdecimalsep{,}
\begin{spreadtab}%
{{tabular}{N{2}{2}N{2}{2}n{2}{2}}}
1.0 & 3.3 & a1+b1 \\
2.2 & 4.4 & a2+b2 \\ \hline
a1+a2 & b1+b2 & \\
\end{spreadtab}
```

Finales Beispiel

In den vorangegangenen Abschnitten haben wir alle notwendigen Tricks und Kniffe gelernt, die für die Erstellung einer komplexen **spreadtab**-Tabelle notwendig sind. Im abschließenden Beispiel sind nochmals alle Funktionen des Pakets, die wir für die Erstellung einer Rechnung benötigen, versammelt:

```

\documentclass{scrartcl}
\usepackage{spreadtab}
\usepackage{eurosym,numprint}
\begin{document}
\setdecimalsep{,} \nprounddigits{2} % Rundungsgenauigkeit
\renewcommand\arraystretch{1.3}
\begin{spreadtab}[\STsavecell{\myresult}{d6}]%
{{tabular}{@{}rcN{2}{2}N{2}{2}@{}}}
@ Artikel& @Anzahl& {@ Preis}& {@ Summe} \\ \hline
@Brot & 1.00 & := {2.00} \, \euro & := {[-1,0]*[-2,0]} \, \euro \\
@Butter & 2.00 & := {1.67} \, \euro & := {[-1,0]*[-2,0]} \, \euro \\
@Milch & 3.00 & := {0.55} \, \euro & := {[-1,0]*[-2,0]} \, \euro \\
@Wein & 4.00 & := {7.50} \, \euro & := {[-1,0]*[-2,0]} \, \euro \\
\hline
& & {@Gesamt:} & := {sum(d1:[0,-1])} \, \euro \\
\end{spreadtab}

\medskip
Die Gesamtsumme ist \numprint{\myresult} \, \euro.
\end{document}

```

Artikel	Anzahl	Preis	Summe
Brot	1	2,00 €	2,00 €
Butter	2	1,67 €	3,34 €
Milch	3	0,55 €	1,65 €
Wein	4	7,50 €	30,00 €
Gesamt:			36,99 €

Die Gesamtsumme ist 36,99 €.

Zusammenfassung

Die vorgestellten Beispiele sollen das Interesse wecken, sich mit diesem interessanten Paket zu beschäftigen. Das sehr gute Handbuch beschreibt noch weitere Befehle und Funktionen und sei jedem ans Herz gelegt, der mit dem Paket arbeiten möchte. Eine umfassende Darstellung der Version 0.2 findet sich auch in der zweiten Auflage des Tabellensatz-Buches von Herbert Voß [3]. Für Anregungen und Beispiele bin ich dankbar, die ich dann über mein Blog auf www.uweziegenhagen.de veröffentlichen werde.

Literatur

- [1] Harald Harders: *The numprint package*; Paketbeschreibung; Febr. 2008; CTAN:macros/latex/contrib/numprint.
- [2] Christian Tellechea: *spreadtab v0.3b*; Paketbeschreibung; Juni 2010; CTAN:macros/latex/contrib/spreadtab.
- [3] Herbert Voß: *Tabellen mit \LaTeX* ; Lehmanns Media, Berlin; 2. Aufl.; 2010; ISBN: 9-783-86541-370-3.

PocketMods mit \LaTeX erstellen

Uwe Ziegenhagen

PocketMods sind kleine Papierheftchen, die aus *einem* kunstvoll gefalteten und eingeschnittenen Blatt Papier bestehen. In diesem Artikel möchte ich verschiedene Wege vorstellen, zu einem eigenen PocketMod zu kommen.

Einleitung

Ein PocketMod (Notizbuch) besteht aus einem einseitig bedruckten Blatt Papier, das in insgesamt acht gleich große Teile unterteilt wird. Bei einem DIN-A4-Blatt entsprechen also die einzelnen Stücke jeweils DIN-A7. Schneidet man das Blatt dann ein und faltet die einzelnen Stücke gemäß Vorgabe, so entsteht eine kleine Broschüre, die auf vielfältige Art nützlich sein kann. PocketMod.com bietet von Sudoku-Vorlagen und Kalendern über Einkaufs- und Aufgabenlisten und RSS-Feeds alles an, was sich sinnvollerweise auf ein so kleines Format drucken lässt.

Faltanleitungen findet man am einfachsten über das Internet, gute Anleitungen sind zum Beispiel [1] oder [3]. Es gibt auch diverse Flash-Anwendungen im Netz, die die komplette Erstellung übernehmen [2].

Die Schwierigkeit, die den Einsatz eines n -auf-eine-Seite-Druckertreibers für die Erstellung von PocketMods quasi unmöglich macht, liegt in der Ausrichtung und Anordnung der Seiten, wie sie in Abbildung 1 veranschaulicht wird. Links oben liegen die Seiten 6 bis 8, gefolgt von 1, die restlichen Seiten sind gedreht. Der Schnitt erfolgt in der Mitte zwischen den Seiten 4/7 und 3/8.