

Installation Wohin mit `exercise.cls`?

Zur Installation der `exercise`-Klasse muss die Datei `exercise.cls` von L^AT_EX gefunden werden. Dazu muss

- 1) die Datei `exercise.cls` am besten im Verzeichnis `tex/latex/exercise` liegen. Unter MikTeX und Windows also zum Beispiel in folgendem Verzeichnis.

```
C:\Programme\MikTeX\tex\latex\exercise
```

Unter MacTeX und Mac OS X hingegen folgendes Verzeichnis.

```
/usr/local/texlive/texmf-local/tex/latex/exercise
```

- 2) L^AT_EX angewiesen werden, seine Dateinamen-Datenbank zu aktualisieren. Entweder mit der Anweisung `mktexlsr` oder unter MikTeX und Windows zum Beispiel über **Start - Programme - MikTeX Settings** auf der Registerkarte **Generals** über den Button **Refresh FNDB**.

Ab MikTeX 2.9 durchsucht `mktexlsr` ohne Parameter nicht mehr das Installationsverzeichnis (`common root directory`). Stattdessen wird das benutzerspezifische Verzeichnis (`user root directory`) durchsucht. Also entweder `mktexlsr` mit dem Parameter `-admin` starten, oder die Datei in ein benutzerspezifisches Verzeichnis kopieren. Diese werden angezeigt, wenn `mktexlsr` ausgeführt wird. Üblicherweise bietet sich zum Beispiel folgendes Verzeichnis an:

```
C:\Dokumente und Einstellungen\<<Benutzername>\Anwendungsdaten\  
MikTeX\2.9\tex\latex\exercise
```

Wer auf die richtige Installation keine Lust hat, kann die Klasse auch verwenden, indem sich die Datei `exercise.cls` im gleichen Verzeichnis wie das eigene L^AT_EX-Dokument befindet.

Vorraussetzungen Was der Autor alles angenommen hat

Die Klasse setzt automatisch den Eingabe-Zeichensatz auf UTF-8 und den Font auf `lmodern`. Das hat den Vorteil, dass Du Dich um nichts mehr kümmern musst, und den Nachteil, dass Du das so annehmen musst. Insbesondere muss Dein Quellcode als UTF-8 gespeichert werden, oder keine Sonderzeichen enthalten.

Die Klasse verwendet KOMA-Script [1] und setzt automatisch die Papiergröße auf A4, die Schriftgröße auf 10pt und die Absatzsteuerung auf einen halben Absatz ohne Einzug. Außerdem wird die Größe der Überschriften auf ein erträgliches Maß reduziert.

Auch die Fußzeile ist fest vorgegeben. Sie wird unten rechts im Randbereich gesetzt und besteht nur aus der aktuellen gefolgt von der Gesamt-Seitenzahl. Die Fußzeile wird nur gesetzt, wenn das Dokument mehr als eine Seite hat. Damit L^AT_EX die Gesamt-Seitenzahl kennt, muss das Dokument zweimal kompiliert werden. Nach dem ersten Durchlauf wird immer die aktuelle Seitenzahl in der Fußzeile ausgegeben.

Für den Satz von mathematischen Formeln wird die Umgebungen `amsmath` [2] eigentlich zwingend benötigt. Für verschiedene Symbole ist die Klasse `amssymb` sehr nützlich. Daher sind beide Pakete der American Mathematical Society bereits eingebunden. Zudem werden in mathematischen Umgebungen (`align`, `gather`, ...) alle Formeln automatisch linksbündig mit einem Einzug und ohne Nummerierung gesetzt. Die Nummerierung wird nur gesetzt, wenn auf die Formel bzw. die Zeile in der Formel referenziert wird.

Damit alle Verweise in der PDF-Ausgabe als Hyperlinks dargestellt werden, wird automatisch das Paket `hyperref` [6] eingebunden. Links werden weder durch eine Farbe noch durch einen Rahmen hervorgehoben, da das primäre Ausgabemedium für Übungsblätter Papier ist.¹ Die Anker für die Links werden durch den `\caption`-Befehl gesetzt. Daher zeigen die Links immer

¹Und auf Papier kann man leider noch immer keine Links anklicken...

auf das Ende der Float-Umgebung. Um diesen Umstand zu optimieren wird automatisch das Paket `hypcap` [7] mit der Option `all` geladen. Hier werden die Float-Umgebungen `table`, `table*`, `figure` und `figure*` so umdefiniert, dass der Anker am Anfang der Umgebung gesetzt wird. Dies verlangt aber, dass jede der genannten Umgebungen genau einen `\caption-`Befehl enthält.

Schließlich ist das Paket `xcolor` [4] eingebunden und kann überall verwendet werden, wo eine Farbangabe gebraucht wird.

Einbinden der Klasse Und wie benutze ich das jetzt?

Um die Klasse zu verwenden, muss das L^AT_EX-Dokument mit der Zeile

```
\documentclass [<Optionen>] {exercise}
```

beginnen. An dieser Stelle können auch die weiteren Optionen der Klasse angegeben werden.

In einer stillen Verneigung an die gute alte Zeit und vor allem die gute alte Schreibmaschine, neigen noch heute viele Menschen dazu, den 1.5-fachen für den einzig wahren Zeilenabstand zu halten. Um diesem Umstand gerecht zu werden – oder einfach um möglichst viele Seiten zu füllen – kann die Klasse angewiesen werden, 1.5-fachen Zeilenabstand zu setzen, indem bei der Einbindung die Option `biglinespread` ergänzt wird.

```
\documentclass [biglinespread] {exercise}
```

Die Option `biglinespread` ist eine boolesche Option. Die Variante ohne Wert steht für `biglinespread=true`. Alternativ können auch die Werte `true` oder `on` zum Einschalten der Option oder `false`, sowie `off` zum Abschalten der Option (was aber der Standard ist) verwendet werden.

Weitere Optionen werden in den Abschnitten *Kopfzeile*, *Aufgaben* und *Farbe* erläutert.

Kopfzeile Was der Übungsleiter alles auf dem Zettel haben will

Die Informationen in der Kopfzeile können mit drei Befehlen an die jeweilige Übung angepasst werden.

- In der Mitte der Kopfzeile befindet sich der Name der Übung oder ein anderer Dokumententitel (auf dieser Seite »exercise-Klasse«) und darunter die aktuelle Übung (z. B. »Übungszettel 42«) oder ein anderer Untertitel (auf dieser Seite »Anleitung«). Diese Informationen müssen der Klasse im Kopf deines Quellcodes (also noch vor dem Code `\begin{document}`) mitgeteilt werden durch den Befehl

```
\settitle [<Untertitel>] {<Titel>}
```

Dabei ist der Untertitel optional. Wird kein Untertitel angegeben und ist die Option `sheet` (siehe *Aufgabe*) gesetzt, so wird automatisch ein Untertitel der Form »Übungszettel 42« erzeugt. Damit die Option `sheet` einen Untertitel setzen kann, muss allerdings ein Titel gesetzt werden.

Mithilfe des Paktes `hyperref` [6] wird automatisch der Untertitel als Titel und der Titel als Thema in den Dateieigenschaften des PDF-Dokumentes gesetzt.

- Rechts oben in der Kopfzeile können die Namen der Studenten gesetzt werden. Der Befehl dazu lautet

```
\addstudent [<Nummer>] {<Name>}
```

Dieser Befehl fügt der Liste der Studenten einen hinzu, wobei die Matrikelnummer (optionaler Parameter) in Klammern hinter den Namen gesetzt wird. Durch mehrmaliges Aufrufen des Befehls können beliebig viele Studenten hinzugefügt werden, wobei das Layout bis zu drei Namen gut verkraftet.

Die Namen der Studenten ohne Matrikelnummern werden jeweils mit Komma getrennt als Autoren in den Eigenschaften des PDF-Dokumentes gesetzt.

- Links in der Kopfzeile kann eine zusätzliche Information von geringerer Bedeutung angegeben werden. Zum Beispiel die Übungsgruppe. Entsprechend schnell ist auch dieser Befehl erraten

```
\setgroup{<info>}
```

Aufgaben Überschriften vollautomatisch – naja fast

Überschriften können in dieser Klasse ganz normal durch `\section`, `\subsection`, ... gesetzt werden. Allerdings ist die Nummerierung ausgeschaltet, da die Überschrift normalerweise die Nummer der Aufgabe enthalten soll und eine doppelte Nummerierung der Form »1. Aufgabe 1« ist nicht sonderlich elegant. Da die meisten Übungszettel mit Aufgabe 1 beginnen und anschließend fortlaufend durchnummeriert sind, wurde wieder eine automatische Nummerierung eingeführt. Durch den Befehl `\task` wird eine Überschrift »Aufgabe« inklusive Nummer automatisch gesetzt.

Allerdings ist die eigentliche Funktion einer Überschrift, dass man sich in einem Dokument zurechtfindet. Durch eine reine Nummerierung der Form »Aufgabe 1« bis »Aufgabe 42« ist das allerdings noch nicht direkt gewährleistet. Daher kann der Überschrift eine Ergänzung hinzugefügt werden, die weniger kräftig und mit einem leichten Abstand neben die eigentliche Überschrift gesetzt wird.

```
\task[<Untertitel>]
```

Soll die Zählung der Aufgaben nicht bei 1 beginnen, kann mit der Klassenoption `task` die Nummer der ersten Aufgabe angegeben werden. Soll vor der Nummer der Aufgabe jedes Mal die Nummer des Aufgabenzettels gesetzt werden, muss der `exercise`-Klasse diese über die Klassenoption `sheet` mitgeteilt werden. Dadurch wird bei Verwendung von `\settitle` der Untertitel automatisch gesetzt (siehe Kopfzeile). Damit die Nummer dann auch zu sehen ist, muss zusätzlich die boolesche Klassenoption `prefix` angegeben werden. Sollen also beispielsweise die Aufgaben in der Form »Aufgabe 7.42«, »Aufgabe 7.43«, ... nummeriert werden, so muss die `exercise`-Klasse mit folgendem Befehl eingebunden werden:

```
\documentclass[sheet=7, prefix,  
task=42]{exercise}
```

Für ganz abstruse Aufgabennummerierungen können die Zähler `task` und `sheet` manuell gesetzt werden. Dabei ist `task` auf den Wert der *vorherigen* Aufgabe gesetzt werden. Soll die Nummerierung also mit 6 beginnen, muss der Zähler auf 5 gesetzt werden, da jeweils *vor* der Ausgabe der Überschrift der Zähler inkrementiert wird. Der Zähler `sheet` hingegen wird nie inkrementiert, sondern nur ausgegeben.

```
\setcounter{task}{12}  
\setcounter{sheet}{4}
```

Versagt die automatische Zählung komplett, da die Aufgaben nicht nummeriert sind, oder aus anderen Gründen eine eigene Überschrift gesetzt werden soll, kann der Titel und der Untertitel manuell angegeben werden. Dabei ist die Reihenfolge der Optionen zu beachten!

```
\task [<Untertitel>] [<Titel>]
```

Soll statt intern von `\task` ein anderer Befehl statt `\section` verwendet werden, so kann der Befehl `\taskcommand` undefiniert werden:

```
\renewcommand{\taskcommand}{subsection}
```

Listen Und wenn meine Aufgabe mehr Teile hat?

Um möglichst elegant einzelne Teilaufgaben markieren zu können, wird das Paket `paralist` [9] eingebunden. Diese Paket löst zwei wesentliche Probleme der klassischen \LaTeX -Listen. Zum einen können damit die Arten der Nummerierung elegant gewechselt werden, und zum zweiten gibt es zusätzliche Umgebungen für Listen ohne vertikalen Abstand. Enthalten alle Listenelemente nur eine Zeile Text, sieht der große Abstand zwischen den Listenelementen meist etwas übertrieben aus. Hier können die Umgebungen `compactenum` und `compactitem` verwendet werden.

Da Teilaufgaben meistens mit »a)«, »b)«, ... nummeriert werden, wird die Art der Nummerierung der Umgebungen `enumerate` und `compactenum` von der `exercise`-Klasse folgendermaßen gesetzt:

- a) erste Ebene
 - 1. zweite Ebene
 - i. dritte Ebene
 - A. vierte Ebene

Eine alternative Nummerierung wird durch die zusätzliche Umgebung `varenumerate` und `varcompactenum` bereitgestellt. Da die Nummerierung von den Umgebungen gewechselt wird, können innerhalb der Umgebungen auch wieder die normalen Umgebungen verwendet werden und die alternative Nummerierung wird trotzdem fortgesetzt:

- 1) erste Ebene
 - a) zweite Ebene
 - i. dritte Ebene
 - A. vierte Ebene

Nicht nummerierte Listen haben auch in der `exercise`-Klasse die normale \LaTeX -Form:

- erste Ebene
 - zweite Ebene
 - * dritte Ebene
 - vierte Ebene

Gerade bei der Verwendung von `compactitem` möchte man allerdings oft auf die erste Ebene verzichten und direkt mit den Strichen beginnen. Für diesen Fall steht auch hier eine alternative Umgebung `varitemize` und `varcompactitem` zur Verfügung:

- erste Ebene
 - * zweite Ebene
 - dritte Ebene
 - vierte Ebene

Soll die Nummerierung noch weiter den eigenen Ansprüchen oder dem Aufgabenzettel angepasst werden stehen zwei Möglichkeiten zur Auswahl bereit: Die Nummerierung kann entweder als Option für genau eine Liste geändert werden oder durch die Befehle `\setdefaultitem` und `\setdefaultenum`. Beiden Befehlen werden jeweils die Werte für Listen erster bis vierter Ordnung übergeben. Wird ein Parameter leer gelassen, bleibt der Standard erhalten. Für ungeordnete Listen kann eine beliebige Zeichenfolge übergeben werden, die exakt in der Form als

Listenpunkt verwendet wird. Für Aufzählungen werden die Buchstaben i , I , a , A und 1 durch die Nummer im entsprechenden Format ersetzt. Soll ein solches Zeichen nicht ersetzt werden, muss es in geschweifte Klammern eingeschlossen werden. Soll die farbliche Hervorhebung der Nummerierungen erhalten bleiben, so muss diese explizit angegeben werden. So kann zum Beispiel die Art der Nummerierung für Listen erster Ordnung geändert werden durch

```
\setdefaultenum{\color{maincolor}{Teil} I}{}{}{}
```

Der linke Abstand der Liste wird durch diesen Befehl *nicht* automatisch angepasst. Reicht der Standard-Abstand nicht aus, muss der Abstand mithilfe des Befehls `\setdefaultleftmargin` vergrößert werden. Die L^AT_EX-Standardwerte werden eingestellt durch

```
\setdefaultleftmargin{2.5em}{2.2em}{1.87em}{1.7em}{1em}{1em}
```

Die Standardwerte heißen natürlich so, weil sie voreingestellt sind. Die Werte dienen daher nur der Orientierung für Umdefinitionen. Ein erneutes Setzen ist sinnfrei.

Soll die Nummerierung nur für eine Liste geändert werden, reicht eine Option aus.

```
\begin{compactenum}[A]
  \item eine
  \item kurze
  \item Liste.
\end{compactenum}
```

Wir erhalten

- A) eine
- B) kurze
- C) Liste.

Der linke Abstand passt sich in diesem Fall automatisch an. Der Mechanismus versagt allerdings manchmal, sodass man zu obiger Methode greifen muss. Man beachte, dass trotz manuellen Setzens der Nummerierung, die Zahlen trotzdem in Farbe gesetzt werden. Durch explizites Angeben einer Farbe kann dieses Verhalten aber überschrieben werden.

Listen++ Wenn langweilige normale Listen einfach nicht mehr ausreichen

Der Befehl

```
\begin{compactdingitem}{52}
  \item eine ganz
  \item neue Form
  \item der Liste.
\end{compactdingitem}
```

liefert

- ✓ eine ganz
- ✓ neue Form
- ✓ der Liste.

Dabei ist 52 ein Symbol aus dem Zapf Dingbats Font. Andere Fonts aus dem `pifont` Paket können ebenfalls verwendet werden, wenn der Name des Fonts als optionaler Parameter angegeben wird. Für Zapf Dingbats ist `pzb` voreingestellt. Analog zur Umgebung `compactdingitem` existiert auch noch die Umgebungen `dingitem`.

33:		34:		35:		36:		37:		38:		39:	
40:		41:		42:		43:		44:		45:		46:	
47:		48:		49:		50:		51:		52:		53:	
54:		55:		56:		57:		58:		59:		60:	
61:		62:		63:		64:		65:		66:		67:	
68:		69:		70:		71:		72:		73:		74:	
75:		76:		77:		78:		79:		80:		81:	
82:		83:		84:		85:		86:		87:		88:	
89:		90:		91:		92:		93:		94:		95:	
96:		97:		98:		99:		100:		101:		102:	
103:		104:		105:		106:		107:		108:		109:	
110:		111:		112:		113:		114:		115:		116:	
117:		118:		119:		120:		121:		122:		123:	
124:		125:		126:		161:		162:		163:		164:	
165:		166:		167:		168:		169:		170:		171:	
172:		173:		174:		175:		176:		177:		178:	
179:		180:		181:		182:		183:		184:		185:	
186:		187:		188:		189:		190:		191:		192:	
193:		194:		195:		196:		197:		198:		199:	
200:		201:		202:		203:		204:		205:		206:	
207:		208:		209:		210:		211:		212:		213:	
214:		215:		216:		217:		218:		219:		220:	
221:		222:		223:		224:		225:		226:		227:	
228:		229:		230:		231:		232:		233:		234:	
235:		236:		237:		238:		239:		240:		241:	
242:		243:		244:		245:		246:		247:		248:	
249:		250:		251:		252:		253:		254:			

Tabelle 1: Referenz aller Zeichen aus der Zapf Dingbats.

Die Umgebungen `dingenum` und `compactdingenum` funktionieren ähnlich, inkrementieren den Index des Zeichens aber für jeden Listenelement. Auf diese Weise können die Zahlen der Zapf Dingbats für nummerierte Listen verwendet werden. Es können natürlich auch andere Zeichen auf diese Weise benutzt werden. So liefert

```
\begin{compactdingenum}{168}
  \item eine Liste,
  \item die irgendwie
  \item an Kartenspiele
  \item erinnert.
\end{compactdingenum}
```

zum Beispiel

- eine Liste,
- die irgendwie
- an Kartenspiele
- erinnert.

Andere Fonts aus dem `pifont` Paket können auch hier verwendet werden, wenn der Name des Fonts als optionaler Parameter angegeben wird. Enumerations dieser Art können leider im Moment nicht verschachtelt werden, da der Autor dieser Klasse die Counter dazu nicht ausreichend im Griff hat.

In Tabelle 1 findet sich eine Übersicht aller Symbole aus der Zapf Dingbats.

vollständiges Beispiel Wie sieht sowas denn jetzt aus?

exercise-example.tex

```
\documentclass[sheet=3, task=6, prefix]{exercise}

\setgroup{Gruppe 4}
\settitle{Analysis I}
\addstudent[123456]{Malte Schmitz}
\addstudent[789012]{Max Muster}

\begin{document}
  \task[ein Beispiel]
  \begin{enumerate}
    \item Hier kommt die Bearbeitung der erste Aufgabe.

    Noch ein Absatz.

    \item Zweiter Teil der ersten Aufgaben.

    Noch ein Absatz
  \end{enumerate}

  \task[ein mathematisches Beispiel]
  Hier kommt die Bearbeitung der zweiten Aufgabe.

  \[ a^2+b^2=c^2 \]

  Noch ein Absatz.
\end{document}
```

Beweis w. z. b. w. – was zu bezweifeln wäre

Für den Satz von Beweisen wird gerne die `proof`-Umgebung aus dem `amsthm`-Paket [3] empfohlen. Soll diese verwendet werden, muss das Paket explizit geladen werden und überschreibt dann die `proof`-Umgebung dieser Klasse. Da auf den meisten Übungsblättern nur die Umgebung für Beweis benötigt wird und die weiteren Funktionen der `Theorem`-Klasse nicht gebraucht werden, stellt diese Klasse eine eigenen Beweis-Umgebung zur Verfügung.

Da dem Autor dieser Klasse die optische Kennzeichnung von Beweisen nicht zusagte, sind Beweise in dieser Klasse etwas deutlicher hervorgehoben. Wie das Original schreibt die Umgebung vor den Beweis das Wort »Beweis« und beendet den Beweis automatisch mit dem Kasten-Symbol.

Damit man zunächst formschön etwas behaupten und anschließend beweisen kann, steht ebenfalls eine `claim`-Umgebung zur Verfügung. Zusammen ergibt sich folgende Verwendung

```
\begin{claim}
  <Behauptung>
\end{claim}

\begin{proof}
  <Beweis>
\end{proof}
```

Endet der Beweis mit einer mathematischen Formel oder einer Liste, landet das q. e. d.-Symbol unter der Umgebung und damit eine Zeile zu tief. Zur Vermeidung dieser unschönen Verschiebung kann das Symbol mit dem Befehl `\qed` in der Umgebung gesetzt werden. Da der Befehl in einer `proof`-Umgebung nur bei der ersten Verwendung ein Symbol setzt, wird am Ende der Umgebung dann kein Symbol mehr gesetzt:

```
\begin{proof}
  \ldots

  Und damit ergibt sich
  \[ f(\code{G, s, t}) \in 2k\textsc{-colorable}. \qed \]
\end{proof}
```

Soll nur der Kasten unter einen Text gesetzt werden, ohne die Beweis-Umgebung zu verwenden, kann der Befehl `\qed` verwendet werden. Der Kasten wird automatisch nach rechts geschoben.

Soll ein anderes Symbol gesetzt werden, muss der Befehl `\qedsymbol` umdefiniert werden:

```
\renewcommand{\qedsymbol}{q.\,e.\,d.}
```

Farbe Mir ist das alles zu orange hier!

Die Standardfarbe der `exercise`-Klasse für alle Fälle ist Orange. Denn Orange hat den großen Vorteil, dass im Gegensatz zu rot und grün keine unmittelbare Bedeutung hat, sich besser als blau vom schwarzen Text abhebt und besser als gelb zu vom weißen Hintergrund. Wem trotz aller Argumente das Orange einfach nicht gefallen will, der kann die Hauptfarbe durch die Klassenoption `maincolor` ändern. Diese Option muss beim Einbinden der Klasse angegeben werden:

```
\documentclass[maincolor=purple]{exercise}
```

Im Dokument kann `maincolor` als normale Farbe verwendet werden. Das hat den Vorteil, dass sich auch durch eigenen Farbspielereien an den aktuellen Farbstil anpassen. Natürlich lässt sich der ganze Farbenspuk durch setzen von `maincolor=black` auch ganz abschalten.

Für Studenten der Universität zu Lübeck kann durch

```
\documentclass[maincolor=uni-luebeck]{exercise}
```

die Farbe für alle Fälle auf die offizielle Farbe der Universität gesetzt werden. Bezüglich der Tauglichkeit dieser Farbe oder gar dem neuen Logo der Uni überlasse ich an dieser Stelle das Urteil dem geneigten Leser.

Anführungszeichen deutsche Gänsefüßchen haben es schwer

Da die Verwendung korrekter Anführungszeichen in \LaTeX immer etwas trickreich ist, wird das Paket `csquotes` [10] eingebunden und die Form der Anführungszeichen auf die deutschen Guillemets gesetzt. Auf diese Weise können Wörter oder ganze Sätze durch einen einfachen Befehl in Anführungszeichen gesetzt werden, ohne dass der Autor sich um die korrekte Verwendung kümmern muss.

```
\enquote{<Text>}
```


Da der Autor dieser Klasse die deutschen Guillemets den klassischen deutschen Gänsefüßchen vorzieht, werden diese von `\enquote` gesetzt. Werden in besonderen Fällen klassische Anführungszeichen benötigt, können diese am einfachsten mit den Befehlen `\glqq` („), `\glq` (,), `\grq` (‘) und `\grqq` (“) gesetzt werden.

Sprachunterstützung Wir können auch fremdländisch

Die `exercise`-Klasse hat eine eingebaute Sprachunterstützung, die es bisher erlaubt, die Klasse auf englisch umzuschalten. Die Sprachoption wird an das `babel`-Paket [8] weitergereicht, was viele \LaTeX -Standard-Texte übersetzt und die typografischen Konventionen auf englisch umschaltet. Weiterhin werden die Texte der `exercise`-Klasse (Aufgabe, Übungszettel, Beweis, Behauptung) auf Englisch gesetzt. Um die Sprache auf Englisch umzuschalten muss die Klasse mit der Option `language` geladen werden.

```
\documentclass[language=en]{exercise}
```

Gültige Werte für den Schlüssel `language` sind derzeit `en` und `de`.

Sollen die Texte der `exercise`-Klasse darüber hinaus angepasst werden, müssen die entsprechenden Befehle undefiniert werden. Die deutschen Texte würde man durch folgende Befehle erhalten.

```
\renewcommand{\taskname}{Aufgabe}  
\renewcommand{\sheetname}{Übungszettel}  
\renewcommand{\proofname}{Beweis}  
\renewcommand{\claimname}{Behauptung}
```

Die Anweisungen sind natürlich sinnlos, da die Texte bereits deutsch sind. Sie dienen hier nur der Veranschaulichung des Prinzips und der allgemeinen Orientierung.

Quellcode des echten Informatikers liebste Aufgabe

Zum Formschönen setzen von Quellcode ist das `listings` Paket [11] ideal. Da es automatisch eingebunden wird, stehen die folgenden Befehle zur Verfügung:

- `\lstset`

Hiermit können global Optionen gesetzt werden. Dies ist auch als optionaler Parameter jeweils in den folgenden Befehlen möglich. Eine Auswahl wichtiger Optionen:

numbers Standardmäßig werden die Zeilen nummeriert. Mit dem Wert `none` kann dies abgeschaltet werden.

caption Beschriftung für das Listing angeben.

gobble Anzahl Zeichen am Anfang jeder Zeile abschneiden. Sinnvolle Option, wenn der Quellcode im \LaTeX -Quellcode eingerückt werden soll.

mathescape Mit dem Wert `true` kann das automatische Escapen von $\$$ -Zeichen aktiviert werden. So können mathematische Formeln in Pseudo-Code verwendet werden.

language Die Sprache für das Syntax-Highlighting.

identifiestyle Kommandos, die den Stil für Bezeichner festlegen. Durch die Verwendung von `\color{black}` kann zum Beispiel die blaue Hervorhebung deaktiviert werden.

morekeywords erlaubt die Definition weiterer Schlüsselwörter.

- `lstlisting`

Umgebung, die das direkte Einfügen von Quellcode erlaubt.

- `\lstinputlisting`

Befehl, der dann Einfügen von Quellcode aus externen Dateien erlaubt.

- `\lstinline`

Befehl, der das Auszeichnen von Quellcode-Stücken im Fließtext erlaubt.

Leider hat dieses Paket einige Schwierigkeiten mit UTF-8 Sonderzeichen. Daher funktioniert es in dieser Klasse nur stabil mit ASCII-Zeichen. Einige wenige deutsche Sonderzeichen können auch verwendet werden, werden aber immer schwarz dargestellt. Erlaubt sind ä, ü, ö und ß.

Tabellen Striche sind ganz böse

Auch wenn Tabellen gerne durch eine Vielzahl von (faszinierender Weise in L^AT_EX auch gerne doppelt und dreifachen) Strichen dargestellt werden, so erhöht dies die Lesbarkeit selten. Hat man mit dem Bleistift oder der Schreibmaschine wenig andere Möglichkeiten, so bietet es sich im professionellen Textsatz doch an, Tabellen mit möglichst wenigen Strichen zu setzen. Vertikale Striche, die in den meisten Fällen einfach nur den horizontalen Lesefluss unterbrechen, sollten so wenig wie möglich eingesetzt werden. Horizontale Striche können zur Abtrennung von Kopfzeilen und Bereichen verwendet werden. Eleganter ist aber die dezente farbige Hinterlegung der Tabellenzellen.

Wie das vorgangegenagen Plädoyer für farbige Tabellen schon vermuten lässt, sind diese in der `exercise`-Klasse leicht zu realisieren. Die Umgebung `zebratabular` verhält sich genau wie `tabular` mit dem einzigen Unterschied, dass die Tabelle automatisch abwechselnd gefärbte Zeilen hat. Um die automatisch Färbung innerhalb der Tabelle ab- und wieder einzuschalten können die Befehle `\hiderowcolors` und `\showrowcolors` aus dem `xcolor`-Paket [4] verwendet werden. Um die Kopfzeile etwas kräftiger einzufärben steht der Befehl `\headerrow` zur Verfügung. Dieser muss zu Beginn der Kopfzeile eingefügt werden.

Im Gegensatz zu früheren Versionen der `exercise`-Klasse werden mit der `tabular`-Umgebung gesetzte Tabellen nicht mehr automatisch eingefärbt, sodass auch mit der `array`-Umgebung und verwandten Matrix- und Vektor-Umgebungen gesetzte Formeln nicht mehr farbig hinterlegt werden.

Wem das jetzt alles zu schnell ging, der vergnüge sich mit dem folgenden Beispiel.

Jahr	Prozessor	MHz
1975	6502 (C64)	1
1985	80386	16
2005	Pentium 4	2 800
2030	Phoenix 3	7 320 000
2050	...	
2070	...	

Und hier der entsprechende L^AT_EX-Code:

```
\begin{zebratabular}{llr}
  \headerrow Jahr & Prozessor & MHz \\
  1975 & 6502 (C64) & 1 \\
  1985 & 80386 & 16 \\
  2005 & Pentium 4 & 2\,800 \\
  2030 & Phoenix 3 & 7\,320\,000 \\
  \hiderowcolors
  2050 & \ldots & \\
  2070 & \ldots & \end{zebratabular}
```

```
2070 & \ldots
\end{zebratabular}
```

Weitere Informationen über die Verwendung von `xcolor` in Tabellen und anderen Fällen finden sich in [5].

Vektoren Pfeil oder nicht Pfeil?

Der Befehl `\vec a` kann verwendet werden, um einen Vektor \mathbf{a} zu erzeugen. Mathematiker setzen Vektoren normalerweise fett. Da Tafeln und Stifte keinen Fettdruck beherrschen, werden handschriftliche Vektoren unterstrichen. Physiker markieren Vektoren gerne mit einem kleinen Pfeil über dem Zeichen. Um zwischen allen diesen Varianten auswählen zu können, steht die Klassenoption `vec` mit den möglichen Werten `bold`, `underline` und `arrow` zur Verfügung. Voreingestellt ist die Option `bold`.

weitere Befehle kleine Tricks vereinfachen das Leben

Der Befehl `\tick` setzt einen Haken \checkmark .

Die Befehle `\R`, `\N`, `\Z`, `\Q` setzen in der Mathe-Umgebung die Zahlenräume $\mathbb{R}, \mathbb{N}, \mathbb{Z}, \mathbb{Q}$. Für weitere Mengen steht die Abkürzung `\set` zur Verfügung. So führen zum Beispiel die Befehle `\set K \subset \set P` zu $\mathbb{K} \subset \mathbb{P}$.

Der Befehl `\blitz` kann verwendet werden, um einen Widerspruch anzuzeigen. ⚡

kleine Überschrift Der Befehl `\inhead` kann ähnlich dem Befehl `\paragraph` verwendet werden, um einem Absatz eine Überschrift zu geben. Anders als `\paragraph` führt `\inhead` aber keine weitere interne L^AT_EX-Magie aus, sodass er an jeder Stelle verwendet werden kann und die Überschrift nicht in Verzeichnisse aufgenommen wird.

Der Befehl `\dx` kann verwendet werden, um das dx nach einem Integral schöner zu setzen. Für eine andere Integrationsvariable steht `\dx[Variable]` zur Verfügung. Für die Integrationsvariable t existiert die Abkürzung `\dt`.

Der Befehl `\trans` kann verwendet werden, um das Transponiert-Symbol an einer Matrix (A^T) schöner zu setzen. `\trans A` ist eine Abkürzung für A^{T} .

Der Befehl `\zz` erzeugt im Mathe-Modus das Symbol \mathbb{Z} aus zwei Zs um eine Behauptung zu markieren.

Um Operatoren im Mathe-Modus verwenden zu können, kann `\DeclareMathOperator` aus dem Paket `amsmath` [2] im Header verwendet werden. Dabei wird aber ein eigener Befehl definiert. Um nur mal schnell einen Operator zu setzen, kann der Befehl `\operatorname` verwendet werden. In dieser Klasse wird der Alias `\op` definiert.

Literatur

- [1] Frank Neukam, Markus Kohm und Axel Kielhorn:
KOMA-Script, 20. Januar 2009.
 CTAN://macros/latex/contrib/koma-script/.
- [2] American Mathematical Society:
User's Guide for the amsmath Package, 25. Februar 2002.
 CTAN://macros/latex/required/amslatex/math/.

- [3] American Mathematical Society:
Using the amsthm Package, August 2004.
CTAN://macros/latex/required/amslatex/classes/
- [4] Dr. Uwe Kern:
Extending L^AT_EX's color facilities: the xcolor package, 21. Januar 2007.
CTAN://macros/latex/contrib/xcolor/.
- [5] Dr. Uwe Kern:
Farbspielereien in L^AT_EX mit dem xcolor-Paket, erschienen in *Die TEXnische Komödie 2/2004*, S. 35–53.
<http://www.olos.de/ukern/publ/tex/pdf/dtk200402.pdf>.
- [6] Sebastian Rahtz und Heiko Oberdiek:
Hypertext marks in L^AT_EX: a manual for hyperref, Dezember 2009.
CTAN://macros/latex/contrib/hyperref/.
- [7] Heiko Oberdiek:
The hypcap package, 8. September 2008.
CTAN://macros/latex/contrib/oberdiek/.
- [8] Johannes Braams:
Babel, a multilingual package for use with L^AT_EX's standard document classes, 6. Juli 2008.
CTAN://macros/latex/required/babel/.
- [9] Bernd Schandl:
paralist, Extended List Environments, 9. Mai 2005.
CTAN://macros/latex/contrib/paralist/.
- [10] Philipp Lehmann:
The csquotes package, 23. September 2009.
CTAN://macros/latex/contrib/csquotes/.
- [11] Carsten Heinz, Brooks Moses:
The Listings Package, 22. Februar 2007.
CTAN://macros/latex/contrib/listings.