

Anleitung zum Schreiben von Aufgaben für SMART

R. REINDL 2001

1 Benötigte Files

Unbedingt benötigt werden die Files `vorspann.tex`, `layout.tex` und `makros.tex`. Diese Files müssen, wie die Quellfiles der einzelnen Aufgaben, in einem Verzeichnis stehen, das von \LaTeX automatisch durchsucht wird. In einer Standard- \TeX -Installation geht z.B. alles unterhalb von `texmf/tex/`.

Auf \LaTeX -Seite müssen die Packages `amsmath`, `amssymb`, `pictex`, `graphicx`, `psfrag`, `ifthen`, `german`, `marvosym` (für Eurozeichen), `textcomp` (für aufrechtes μ), `dsfont` (für Mengenzeichen) und `longtable` zur Verfügung stehen.

In der zweiten Ebene einer `enumerate`-Umgebung werden die Labels bei Verwendung einer \LaTeX -Standardklasse in Klammern gesetzt (z.B. (a)). Bei kurzen, nebeneinander gesetzten Teilaufgaben schreibt man also von Hand (a) ... (b) ... usw. In der Klasse `scrbook`, die wir für die Jahrgangsstufenbücher verwenden, steht bei den Labels nur die abschließende Klammer. Daher muss man in

```
.../texmf/tex/latex/koma-script/scrbook.cls
```

die Zeile

```
\renewcommand*\p@enumiii{\p@enumii\theenumii)}
```

durch

```
\renewcommand*\p@enumiii{\p@enumii(\theenumii)}
```

ersetzen.

Das Paket `marvosym` enthält einen Fehler (es überschreibt den \LaTeX -Befehl `\Rightarrow`).

Daher muss man in

```
.../texmf/tex/latex/misc/marvosym.sty
```

die Zeile

```
\def\Corresponds{\mvchr61}\def\Rightarrow{\mvchr62}
```

durch z.B.

```
\def\Corresponds{\mvchr61}\def\MRightarrow{\mvchr62}
```

ersetzen.

2 Struktur eines SMART-Dokuments

Jede Aufgabe ist in einem eigenen File gespeichert (Quellfile). Der Name eines Quellfiles besteht aus der Jahrgangsstufe (2 Zeichen), dem Autorenkürzel (2 Zeichen) und einer meist dreistelligen Zahl. Die Aufgaben des Autors xy für die fünfte Jahrgangsstufe tragen also die Filenamen `05xy001.tex` bis `05xy999.tex`, wobei das erste Zeichen eine Null und kein „Oh“ ist. Sollte ein Autor für eine Klasse mehr als 1000 Aufgaben schreiben, kann er auch vierstellige Zahlen verwenden :-)

Ein SMART-Dokument besteht aus dem Masterfile, in dem alle Aufgaben eingebunden werden. Das Masterfile beginnt grundsätzlich mit `\input vorspann`. Dadurch wird das File `vorspann.tex` eingebunden, das wiederum alle Packages und die Files `makros.tex` und `layout.tex` einbindet.

Noch vor dem `\begin{document}` werden die Schalter `\ANGABE`, `LOESUNG` und `\FILE` eingestellt:

<code>\ANGABEtrue</code>	Angaben werden gedruckt
<code>\ANGABEfalse</code>	Angaben werden nicht gedruckt
<code>\LOESUNGtrue</code>	Lösungen werden gedruckt
<code>\LOESUNGfalse</code>	Lösungen werden nicht gedruckt
<code>\FILEtrue</code>	Filenamen der Aufgaben am linken Rand
<code>\FILEfalse</code>	keine Filenamen

Standardmäßig (durch `vorspann.tex`) sind `\ANGABEtrue`, `\LOESUNGtrue` und `\FILEfalse` eingestellt.

Für eine Schulaufgabe verwendet man natürlich die Kombination

`\ANGABEtrue`, `\LOESUNGfalse` und `\FILEfalse`,

für ein reines Lösungsblatt dagegen

`\ANGABEfalse`, `\LOESUNGtrue` und `\FILEfalse`.

Die Überschrift eines Aufgabenblattes steht zwischen `\begin{ABTITEL}` und `\end{ABTITEL}`, das Einbinden der Aufgabe `05xy001.tex` geschieht mit `\ABINPUT 05xy001` (ohne `.tex`). Es folgt ein Beispiel eines Aufgabenblatts:

```
\input vorspann
\ANGABEtrue
\LOESUNGtrue
\FILEfalse
\begin{document}
\begin{ABTITEL}
  Gro"se Zahlen
\end{ABTITEL}
\begin{AUFGABEN}
  \ABINPUT 05cm001
  \ABINPUT 05rr009
  \ABINPUT 05rr051
  \ABINPUT 05rr052
  \ABINPUT 05rr053
  \ABINPUT 05rr055
\end{AUFGABEN}
\end{document}
```

3 Struktur eines Quellfiles

Ein Quellfile muss folgendermaßen aufgebaut sein:

```
\ANG{
Text der Angabe
}
```

```
\LSG{
Text der Loesung
}
```

Es gibt Aufgabentypen, z.B. Beweise, bei denen man keine Lösung angibt. Trotzdem darf die Lösung im Quellfile nicht fehlen, da sonst reine Lösungsblätter in der Nummerierung nicht mit den Aufgabenblättern übereinstimmen würden. Aus dem gleichen Grund muss in der Lösung mindestens ein (unsichtbares) `\mbox{}` stehen. Eine Aufgabe ohne Lösung hat also die Struktur

```
\ANG{
Text der Angabe
}
\LSG{\mbox{}}
}
```

4 Grundsätzliches

Umlaute werden mit Anführungszeichen (auf der Tastatur über der 2) gesetzt:

Quelltext	"a	"o	"u	"s
Ergebnis	ä	ö	ü	ß

Einen guten Editor kann man so einstellen, dass er beim betätigen der ä-Taste "a schreibt. Die korrekten deutschen Anführungszeichen erhält man mit `\GF{Text}` (Gänsefüßchen).

Quelltext	Ergebnis
Hans sagt: <code>\GF{Ich nicht.}</code>	Hans sagt: „Ich nicht.“

5 Typografie

Wir halten uns an die üblichen typografischen Regeln:

- Variable in *Italic*, Benennungen in Roman. Benennungen werden durch einen kleinen Abstand (`\,`) von der Maßzahl getrennt. Im Mathemodus setzt man Einheiten also mit `\mathrm{}`, für Brüche von Einheiten steht das Makro `\RFRAC{ }{ }` zur Verfügung. Ein eigenes Makro für „Grad Celsius“ ist auch vorhanden (`\GC`), das gleich den richtigen Abstand zwischen Zahl und Einheit setzt.

Quelltext	Ergebnis
<code>\$A=5\,\mathrm{cm}\cdot 7\,\mathrm{cm}\$</code>	$A = 5 \text{ cm} \cdot 7 \text{ cm}$
<code>\$v=27\,\RFRAC{km}{h}\$</code>	$v = 27 \frac{\text{km}}{\text{h}}$
<code>\$\vartheta=23{,}4\GC\$</code>	$\vartheta = 23,4^\circ\text{C}$

- Bezeichner für geometrische Größen (Punkte, Geraden, Ebenen) werden aufrecht (Roman) gesetzt, ebenso beschreibende Indizes.

Quelltext	Ergebnis
Die maximale Fläche von <code>\Delta\,ABC</code> ist <code>\$A_{\mathrm{max}}=5\,\mathrm{cm}^2\$</code>	Die maximale Fläche von ΔABC ist $A_{\max} = 5 \text{ cm}^2$

- Die Bezeichner fester Funktionen (sin, cos, ...) werden aufrecht gesetzt. Die gängigen Funktionsnamen sind in L^AT_EX schon vorhanden, für ggT und kgV stehen im Mathemodus die Makros \ggT und \kgV bereit.

Quelltext	Ergebnis
<code>\sin^2x+\cos^2x=1</code>	$\sin^2 x + \cos^2 x = 1$
<code>\ggT(a,b)\cdot\kgV(a,b)=a,b</code>	$\text{ggT}(ab) \cdot \text{kgV}(ab) = ab$

- Dezimalbrüche: Im Mathemodus wird nach dem Komma ein hässlicher Abstand gesetzt. Das vermeidet man durch das Schreiben des Kommas in geschweifte Klammern: `$3,75$` liefert 3,75, `$3{,}75$` dagegen 3,75.
- Für die im Mathemodus verwendeten Größenangaben `\displaystyle`, `\textstyle`, `\scriptstyle` und `\scriptscriptstyle` sind die Abkürzungen `\D`, `\T`, `\SC` und `\SSC` vorhanden. Am häufigsten verwendet man `\D`, um Brüche in Textformeln größer darzustellen: `\frac{1}{2}` liefert $\frac{1}{2}$, `\D\frac{1}{2}` dagegen $\frac{1}{2}$.

6 Sonderzeichen

- Das Eurozeichen erhält man im Textmodus mit `\EUR`. Wenn nach dem Zeichen ein Leerzeichen folgen soll, muss man `\EUR{}` verwenden. Im Mathemodus erhält man das Eurozeichen mit `\text{\EUR}` oder `\mbox{\EUR}` oder noch einfacher durch das Makro `\MEUR`.

Quelltext	Ergebnis
<code>5\,EUR, das sind 5\,EUR{} zuviel.</code>	5 €, das sind 5 € zuviel.
<code>\$4{,}20\,MEUR+3{,}72\,MEUR\$</code>	4,20 € + 3,72 €

- Das griechische mu (μ) für die Vorsilbe „Mikro“ muss, da es ja zu einer Benennung gehört, aufrecht gesetzt werden. Das aufrechte μ erhält man im Textmodus mit `\textmu` (wenn ein Leerzeichen folgt, muss `\textmu{}` verwendet werden), im Mathemodus mit `\text{\textmu}`, `\mbox{\textmu}` oder durch das Makro `\Mmu`.

Quelltext	Ergebnis
<code>Die Stromst"arke betr"agt 20\,textmu A.</code>	Die Stromstärke beträgt 20 μ A.
<code>Das aufrechte \textmu{} erh"alt man</code> <code>\$20\,Mmu\mathrm{A}+33\,Mmu\mathrm{A}\$</code>	Das aufrechte μ erhält man 20 μ A + 33 μ A

Funktioniert momentan aus technischen Gründen leider nicht!

Trotzdem verwenden!!! (Wenn es wieder funktioniert, sind alle μ 's richtig gesetzt!)

7 Mathematisches

- Mengenzeichen:

<code>\NN\$</code>	<code>\BB\$</code>	<code>\ZZ\$</code>	<code>\QQ\$</code>	<code>\RR\$</code>	<code>\CC\$</code>	<code>\GG\$</code>	<code>\DD\$</code>	<code>\LL\$</code>	<code>\WW\$</code>
N	B	Z	Q	R	C	G	D	L	W

- $\$DANN\$$ (\implies) und $\$GENAUDANN\$$ (\iff) sind Abkürzungen für $\$Longrightrightarrow\$$ und $\$Longlefttrightarrow\$$.
- Das „ohne“-Zeichen für Mengen (Abkürzung für \backslash): $\$OHNE\$$.
 $A \setminus B = A \cap \overline{B}$ wird erzeugt von $\$A\OHNE B=A\cap \overline{B}\$$.
- Vektorpfeil über Zeichenkette: $\$VPFEIL{\mathrm{AB}}\$$ liefert \overrightarrow{AB} .
- Zwei Formen komplexer Zahlen in Polarform:

Quelltext	Ergebnis
$\$POLAR{r}{\varphi}\$$	$(r \mid \varphi)_p$
$\$D\POLAR{5}{\frac{3\pi}{4}}\$$	$\left(5 \mid \frac{3\pi}{4} \right)_p$
$\$POLARE{r}{\varphi}\$$	$r \cdot E(\varphi)$
$\$D\POLARE{5}{\frac{3\pi}{4}}\$$	$5 \cdot E\left(\frac{3\pi}{4}\right)$

- Zweidimensionale ($\$EPUNKT\$$) und dreidimensionale ($\$RPUNKT\$$) Punkte mit aufrechtem Symbol:

Das erste Argument ist der Name des Punktes, dann folgen die Koordinaten. Die Größen der Klammern und senkrechten Striche sind angepasst.

Quelltext	Ergebnis
$\$D\EPUNKT{P}{3\cdot 10^4}{\frac{3}{4}}\$$	$P\left(3 \cdot 10^4 \mid \frac{3}{4}\right)$
$\$RPUNKT{A}{3}{-0{,}032}{\frac{3}{4}}\$$	$A\left(3 \mid -0,032 \mid \frac{3}{4}\right)$

- Zweidimensionale ($\$EVEKTOR\$$) und dreidimensionale ($\$RVEKTOR\$$) Spaltenvektoren:
Das erste Argument ist l (left), c (center) oder r (right), dann folgen die Koordinaten.

Quelltext	Ergebnis
$\$EVEKTOR{c}{3}{-12}\$$	$\begin{pmatrix} 3 \\ -12 \end{pmatrix}$
$\$EVEKTOR{r}{3}{-12}\$$	$\begin{pmatrix} 3 \\ -12 \end{pmatrix}$
$\$RVEKTOR{c}{3\cdot 10^4}{-0{,}032}{x}\$$	$\begin{pmatrix} 3 \cdot 10^4 \\ -0,032 \\ x \end{pmatrix}$
$\$RVEKTOR{l}{a_1}{a_2}{a_3\cdot b}\$$	$\begin{pmatrix} a_1 \\ a_2 \\ a_3 \cdot b \end{pmatrix}$

8 Makros zum Einbinden von Grafiken

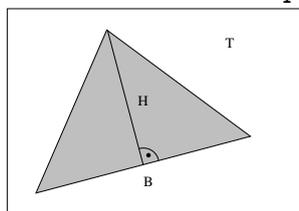
Die Bilder müssen als `.eps`- und `.jpg`- oder `.pdf`-Files vorliegen, da die fertigen Blätter im PostScript- und im pdf-Format erzeugt werden. **Alle** Filenamen sind **ohne** die Endung `.eps` einzugeben!

8.1 EPSBASIS

Aufruf: `\EPSBASIS{Filename}{Breite}{Hoehe}`
oder `\EPSBASIS{Filename}{Breite}{ }`
oder `\EPSBASIS{Filename}{ }{Hoehe}`

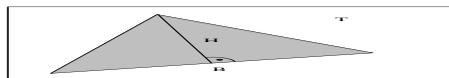
EPSBASIS ist das Grundmakro zur Einbindung von `.eps`-Files, auf das die folgenden Makros aufbauen. Gibt man nur die Breite oder nur die Höhe als Parameter an, dann wird das Originalverhältnis von Höhe zu Breite verwendet.

Beispiel: `\EPSBASIS{bild1}{4cm}{}` bindet das File `bild1.eps` (oder `bild1.jpg`)



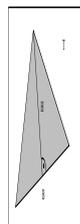
mit der Breite 5 cm im Originalverhältnis ein:

`\EPSBASIS{bild1}{6cm}{1cm}`



, etwas breiter mit

und etwas schmaler



mit `\EPSBASIS{bild1}{1cm}{3cm}`

. Mit **EPSBASIS** werden die Bilder so eingebunden, dass ihre Unterkante auf der Zeile steht. Das ist unpraktisch, wenn z.B. in einer `enumerate`-Umgebung ein neues `\item` mit einer Graphik beginnt. Ein weiterer Nachteil ist, dass die Schrift in den Grafiken mitvergrößert und mitverkleinert wird. Diese Mankos werden mit dem Makro `\EPS` behoben.

8.2 EPS

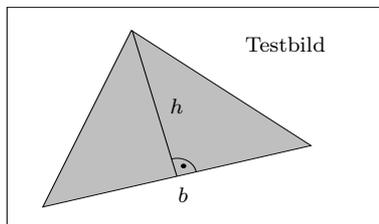
Aufruf: `\EPS{Filename}{Breite}{Hoehe}{psfrag}`
Dabei besteht `psfrag` aus Konstrukten der Form `\PSF{t1}{t2}` oder `\PSFM{t1}{t2}`. Beide Makros ersetzen den Originaltext `t1` des `eps`-Files durch den `TEX`-Text `t2`, wobei `\PSF` linksbündig und `\PSFM` zentriert arbeitet. Die Größe des `TEX`-Textes ist `\scriptstyle`. Mit `\PSFROT{w}{t1}{t2}` erscheint der `TEX`-Text um den Winkel `w` im Gegenuhrzeigersinn gedreht. Den ersetzten Text sieht man nur an der richtigen Stelle im Bild, wenn man das fertige `dvi`-File mit `dvips` nach `ps` wandelt und das `ps`-File mit z.B. `ghostview` betrachtet oder wenn man gleich mit `pdflatex` ein `pdf`-File erzeugt.

Beispiel:

`\EPS{bild1}{5cm}{3cm}{`

```
\PSFM{B}{\$b\$} \PSFM{H}{\$h\$} \PSFM{45}{T}{Testbild}
}
```

bindet das File `bild1` mit der Breite 5 cm und der Höhe 3 cm ein. Dabei liegt die Oberkante des Bildes ungefähr auf der Höhe eines Großbuchstabens der laufenden Zeile. Die Strings B, H und T des Files `bild1.eps` werden durch die TeX-Texte `\$b\$`, `\$h\$` und `Testbild` ersetzt:

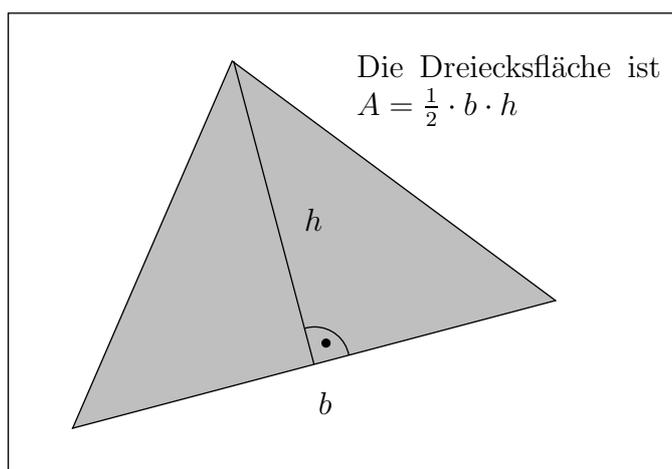


8.3 EPSC

Aufruf: `\EPSC{Filename}{Breite}{Hoehe}{psfrag}`

`\EPSC` arbeitet wie `\EPS`, das Bild wird jedoch zentriert in einem eigenen Absatz gesetzt. Im folgenden Beispiel setzen wir das Bild mit der Breite 9 cm im Originalverhältnis, den String T ersetzen wir durch den zweizeiligen Text „Die Dreiecksfläche ist $A = \frac{1}{2} \cdot b \cdot h$ “. Dazu muss der Text in eine `parbox` gepackt werden. Da das Bild größer ist, wählen wir `\normalsize` als Schriftgröße.

```
\EPSC{bild1}{9cm}{}{
  \PSFM{B}{\normalsize \$b\$} \PSFM{H}{\normalsize \$h\$}
  \PSFM{T}{\parbox[t]{4cm}{\normalsize Die Dreiecksfl"ache ist
    $A=\frac{1}{2}\cdot b\cdot h$}}
}
```



8.4 EPSR

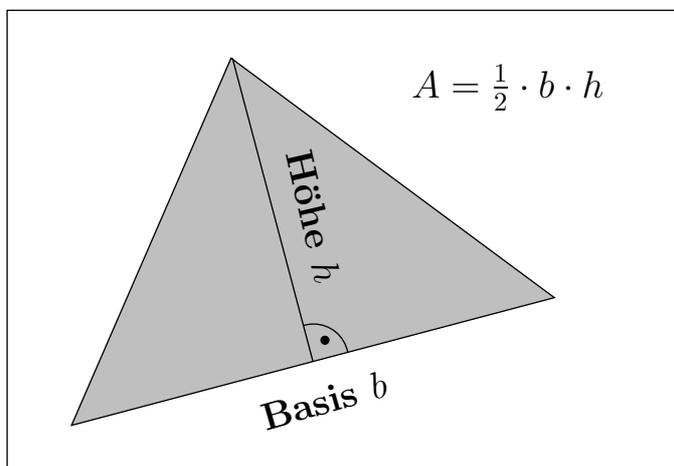
Aufruf: `\EPSR{Textbreite}{Filename}{Bildbreite}{Bildhöhe}{psfrag}{Text}`

Beispiel:

```
\EPSR{6cm}{bild1}{9cm}{}{
  \PSFRONT{16}{B}{\large\bfseries Basis $b$}
  \PSFM{T}{\large $A=\frac{1}{2}\cdot b\cdot h$}
  \PSFRONT{-77}{H}{\large\bfseries H"öhe $h$}
}{
Hier steht der Text, der links vom Bild angeordnet ist! ...
}
```

erzeugt eine parbox der Breite 6 cm für den Text und bindet das File bild1 mit der Breite 9 cm ein. Und so sieht das Ergebnis aus:

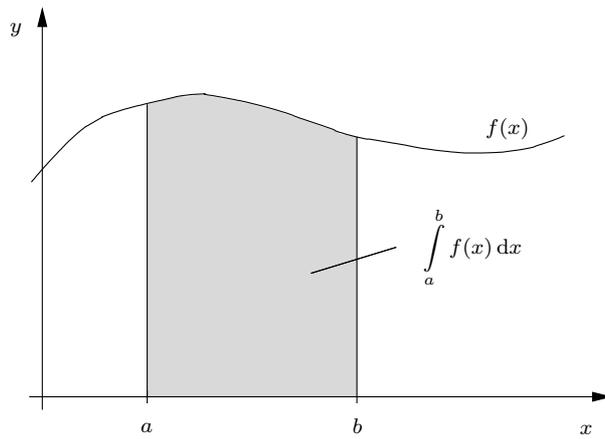
Hier steht der Text, der links vom Bild angeordnet ist! Der Rahmen um das Bild gehört schon zum eps-File. Hier steht der Text, der links vom Bild angeordnet ist! Der Rahmen um das Bild gehört schon zum eps-File. Hier steht der Text, der links vom Bild angeordnet ist! Der Rahmen um das Bild gehört schon zum eps-File. Hier steht der Text, der links vom Bild angeordnet ist! Der Rahmen um das Bild gehört schon zum eps-File.



8.5 EPSB, EPSCB, EPSRB

`\EPSB`, `\EPSCB` und `\EPSRB` funktionieren wie `\EPS`, `\EPSC` und `\EPSR`, jedoch wird ein Rahmen um die Grafik gezeichnet.

```
\EPSC{bild2}{8cm}{}{
  \PSF{T}{\D\int\limits_{a}^bf(x)\, \mathrm{d}x} \PSFM{x}{x}
  \PSFM{y}{y} \PSFM{f}{f(x)} \PSFM{a1}{a} \PSFM{a2}{b}
}
```



```

\EPSCB{bild2}{8cm}{}{
\PSF{T}{\D\int\limits_{a}^{b}f(x)\,\mathrm{d}x} \PSFM{x}{x}
\PSFM{y}{y} \PSFM{f}{f(x)} \PSFM{a1}{a} \PSFM{a2}{b}
}

```

