

The homeworkassignment*class[†]

Adrian C. Hinrichs

`adrian.hinrichs@rwth-aachen.de`

Alexander Bartolomey

`alexander.bartolomey@rwth-aachen.de`

May 3, 2019

Contents

1 Abstract

This class provides a relative simple document-type for homework, mainly created for assignments at the University. This class is inherited from `article`, it is not perfect, but I am trying my very best.

2 Dependencies

2.1 Mandatory Dependencies

This class is build uppon `article`, so of course the first dependency is:

`article` 1992 LESLIE LAMPORT, 1994-97 FRANK MITTELBACH JOHANNES BRAAMS,
THE L^AT_EX-TEAM, <https://www.ctan.org/pkg/kvoptions>,

Because I am very lazy, the `homeworkassignment` is “a little bit” bloated. These are all required packages:

`kvoptions` HEIKO OBERDIEK, <https://www.ctan.org/pkg/kvoptions>,
for `key=value-style` options

`suffix` DAVID KASTRUP, <https://www.ctan.org/pkg/suffix>,
Makes it easy to define `\macro*` commands

^{*}The name was changed with version v3.0, to become compatible with CTANs guidelines and to maintain a degree of backwards compatibility. The class was called `HomeworkAssignment` prior to v3.0

[†]This document corresponds to `homeworkassignment` v4.0, dated 2019/04/17.

`xifthen` JOSSELIN NOIREL, <https://www.ctan.org/pkg/xifthen>,
For if-else-structures

`translations` CLEMENS NIEDERBERGER, <https://www.ctan.org/pkg/translations>,
Implements an easy method of translations.

`amsmath` THE L^AT_EX-TEAM, FRANK MITTELBACH RAINER SCHÖPF, ET AL.,
<https://www.ctan.org/pkg/amsmath>,
For better math-typesetting

`amssymb` AMERICAN MATHEMATICAL SOCIETY, mirror.ctan.org/fonts/amssymb/doc/amssymb.pdf,
For more mathematical symbols

`etoolbox` PHILIPP LEHMAN (INACTIVE), JOSEPH WRIGHT, <https://www.ctan.org/pkg/etoolbox>,
The package is a toolbox of programming facilities geared primarily towards L^AT_EXclass and package authors

`array` FRANK MITTELBACH, DAVID CARLISLE, THE L^AT_EX-TEAM, <https://www.ctan.org/pkg/array>,
A new implementations for tables and arrays

`xparse` FRANK MITTELBACH, CHRIS ROWLEY, DAVID CARLISLE, THE L^AT_EX3 PROJECT, <https://ctan.org/pkg/xparse>,
The package provides a high-level interface for producing documentlevel commands. In that way, it offers a replacement for L^AT_EX 2_ε's `\newcommand` macro, with significantly improved functionality.

`gillius` BOB TENNENT, <https://ctan.org/pkg/gillius>,
A Gillian Sans inspired font, used for all sans serifs fonts

`hyperref` [HTTPS://CTAN.ORG/PKG/HYPERREF](https://CTAN.ORG/PKG/HYPERREF), SebastianRahtz, HeikoOberdiek,
For hyperrefs, obviously

`xcolor` DR. UWE KERN, <https://www.ctan.org/pkg/xcolor>,
For coloring of ToDos

array possibly can be removed

2.2 Recommended Dependencies

These are not loaded automatically, but require a switch as option (see ??). The switch is typically the name of the package.

`tikz` TILL TANTAU, MARK WIBROW, CHRISTIAN FEUERSÄNGER ET AL., <https://www.ctan.org/pkg/pgf>,
An incredible powerfull image tool. When loading TikZ, the homeworkas-signment automatically loads a shipload of TikZ-libraris and own styles. See ?? for more informations

I intend to move these styles to a own package, so that they are usable without the homeworkas-signment

listings CARSTEN HEINZ, BROOKS MOSES, JOBST HOFFMANN, <https://www.ctan.org/pkg/listings>,
 For source-code. Sourcecode in the homeworkassignment is automatically
 framed, printed in scriptsize, and linebeals will be introduced

Loads required Packages

```

1 \RequirePackage{suffix}
2 \RequirePackage{fancyhdr}
3 \RequirePackage{xifthen}
4 \RequirePackage{translations}
5 \PassOptionsToPackage{fleqn}{amsmath}
6 \RequirePackage{amsmath}
7 \RequirePackage{amssymb}
8 \RequirePackage{etoolbox}
9 \RequirePackage{array}
10 \RequirePackage{xparse}
11 \RequirePackage{ifxetex}
12
13 \RequirePackage{wasysym}
14 \RequirePackage{adjustbox}
15
16 \RequirePackage{eso-pic}
17
18 \RequirePackage{xcolor}

```

3 Options

KV-Options is essential for this.

```

19 \RequirePackage{kvoptions}
20 \SetupKeyvalOptions{ family=hwa,
21   prefix=hwa@ }
22 \DeclareDefaultOption{\PassOptionsToClass{\CurrentOptionKey}{article}}

```

problemstyle=<1> These options allow the customizatuion of the displayed numbers. For Example,
 subproblemstyle=<1> if problemstyle=Roman, subproblemstyle=arabic, subsubproblemstyle=roman
 subsubproblemstyle=<1> is passed, The first subsubproblem of the first subproblem of the first problem
 would be labled as **i)** of **Problem I.1**.

Available options are arabic, Alph, alph, Roman, and roman. Standard values are:
 problemstyle=arabic, subproblemstyle=alph, subsubproblemstyle=roman.

```

23 \DeclareStringOption[arabic]{problemsty}
24 \DeclareStringOption[alph]{subproblemsty}
25 \DeclareStringOption[roman]{subsubproblemsty}

```

tikz Loads TikZ-Package and a couple of styles, useful for papers in computer science
 and mathematics. See ?? for more informations

```

26 \DeclareBoolOption[false]{tikz}

```

listings Loads listings package and sets listing layout to use a small fontsize. Adds
 indication for linebreaks.

	27\DeclareBoolOption[false]{listings}
oneside, twoside	Changes layout. oneside is the complementary option to twoside Standard layout is twopaged.
	28\DeclareBoolOption[true]{twoside}
	29\DeclareComplementaryOption{oneside}{twoside}
onecolumn, twocolumn	Changes layout. onecolumn is the complementary option to twocolumn. Standard Layout has one columns
	30\DeclareBoolOption[false]{twocolumn}
	31\DeclareComplementaryOption{onecolumn}{twocolumn}
punchmark	Adds a mark for an hole puncher. Standard Layout has no marking.
	32\DeclareBoolOption[false]{punchmark}
hlines=<1>	KeyValue-option. Takes the level of hlines. Available are all,decreased,header, none, with decreasing number of lines; none displays none, header only the one under headers and decreased adds the big line in the title, while all displays all.
	33\DeclareStringOption[all]{hlines}
todos=<1>	KeyValue-option. Takes which ToDos shall be displayed. Available are all (de- fault),nolist,none. See ?? for explanation of the levels.
	34\DeclareStringOption[all]{todos}
unicode-math	Loads the unicode-math-package and overwrites the damn \QED-Command unicode-math introduces, that creates a filled out box and only works in math- mode, but not telling you that it only works in math-mode or overwrites an al- ready existing command. For a reason, that currently (06 th of December 2018) slips my mind completly, unicode-math needs to be loaded after article, be- cause it needs to be defined

ATTENTION: Please do never, never, never, never, never ever load unicode-math your self, because this breaks **everything**¹

\end{rant}

If XeTeX is used, the default option for this is true, otherwise it is false.

For the handling of the option, see ??

```

35\ifxetex
36\DeclareBoolOption[true]{unicodemath}
37\else
38\DeclareBoolOption[false]{unicodemath}
39\fi

```

Loads article and processes the options

```

40\ProcessKeyvalOptions*
41\ifhwa@twoside
42\PassOptionsToClass{twoside}{article}
43\else

```

```

44 \PassOptionsToClass{oneside}{article}
45 \fi
46 \ifhwa@twocolumn
47 \PassOptionsToClass{twocolumn}{article}
48 \else
49 \PassOptionsToClass{onecolumn}{article}
50 \fi
51 \LoadClass{article}
52
53 \newboolean{hwa@todos@inplace}
54 \newboolean{hwa@todos@list}
55 \setboolean{hwa@todos@inplace}{true}
56 \setboolean{hwa@todos@list}{true}
57 \ifthenelse{\equal{hwa@todos}{all}}{
58 }{
59   \ifthenelse{\equal{hwa@todos}{nolist}}{
60     \ClassWarning{homeworkassignment}{You specified todos=none,
61       there will be no list of TODO}
62     \setboolean{hwa@todos@list}{false}
63   }{
64     \ifthenelse{\equal{hwa@todos}{none}}{
65       \ClassWarning{homeworkassignment}{You specified todos=none,
66         there will be no TODOs printed in the final document}
67       \setboolean{hwa@todos@list}{false}
68       \setboolean{hwa@todos@inplace}{false}
69     }{
70       \ClassError{homeworkassignment}{hwa@todos is not a valid value
71         for the option 'todos'}
72     }
73   }
74 }

```

Load Hyperref (breaks if it is loaded before article

```
75 \RequirePackage{hyperref}
```

Loads listings, if wanted

```

76 \ifhwa@listings
77 \RequirePackage{listings}
78 \lstset{
79   frame = single,
80   breaklines = true,
81   postbreak=\raisebox{0ex}[0ex][0ex]{\ensuremath{\hookrightarrow\space}},
82   basicstyle=\scriptsize
83 }
84 \else
85 \empty
86 \fi

```

`\hwa@hline@L...` Defines new commands to output desired lines and change the constant `\hwa@headrulewidth`

ATTENTION: `\hwa@hline@LONE` breaks the line automatically, in opposite to `\hwa@hline@LTWO`

```

87
88 \newcommand{\hwa@hline@LONE}{\vspace{.25cm} {\hrule height 2pt}
89 \vspace{.25cm}}
90 \newcommand{\hwa@hline@LTWO}{\vspace{.5cm} \hrule \vspace{.25cm}}
91 \newcommand{\hwa@headrulewidth}{.7pt}
92 \ifthenelse{equal{\hwa@hlines}{all}}{
93 \renewcommand{\hwa@hline@LONE}{\vspace{.25cm} {\hrule height 2pt}
94 \vspace{.25cm}}
95 \renewcommand{\hwa@headrulewidth}{.7pt}
96 \renewcommand{\hwa@hline@LTWO}{\vspace{.5cm} \hrule \vspace{.25cm}}
97 }{
98 \ifthenelse{equal{\hwa@hlines}{decreased}}{
99 \renewcommand{\hwa@hline@LONE}{\vspace{.25cm} {\hrule height 2pt}
100 \vspace{.25cm}}
101 \renewcommand{\hwa@headrulewidth}{.7pt}
102 }{\ifthenelse{equal{\hwa@hlines}{header}}{
103 \renewcommand{\hwa@headrulewidth}{.7pt}
104 }{\ifthenelse{equal{\hwa@hlines}{none}}{
105 \renewcommand{\hwa@headrulewidth}{0pt}
106 }{
107 \ClassError{homeworkassignment}{Value '\hwa@hlines' for key 'hlines'
108 is not known}{The option hlines takes an argument to set which
109 hlines are drawn. Possible values are 'all','decreased' , 'header', and
110 'none'. 'all' is standard.}
111 }
112 }
113 \renewcommand{\hwa@hline@LONE}{~\vspace{.5cm}}
114 }
115 \renewcommand{\hwa@hline@LTWO}{\vspace{.75cm}}
116 }

```

If TikZ is wanted, load useful styles

```

117 \ifhwa@tikz
118 \RequirePackage{tikz}
119 \usetikzlibrary{shapes,arrows,positioning,decorations,
120 automata,backgrounds,petri,bending,
121 shapes.multipart}
122 \tikzset{
123 treenode/.style = {shape=circle, rounded corners,
124 draw, align=center},
125 graynode/.style = {fill=gray},
126 normalnode/.style = {treenode, font=\Large, bottom color=white},
127 array/.style = {rectangle split,
128 rectangle split horizontal,
129 rectangle split,

```

```

130     draw}
131 }
132 \fi

Make sure that this is the last Package loaded
133 \RequirePackage{geometry}
134 \ifhwa@twocolumn
135 \geometry{top=2cm, bottom=2cm, left=2cm,
136     headsep=14pt,hmarginratio={1:1}}
137 \else
138 \geometry{top=2cm, bottom=2cm, width=35em,
139     headsep=14pt,hmarginratio={4:3}}
140 \fi

```

4 Layout

Initially, the homeworkassignment had a verry *special* appereance, which became much more customizable with version 3.0, see ?? if you want to know how.

4.1 Headers & Footers

Sets the page-headers.

All headers are cleared before they get any Text — just to be sure.

The headers have the date on the subject and the author on the right side, the tutorial, sheat-title and deadline on the left side, the pagenumber is displayed in the right footer.

If the document is twopaged, the informations in the headers are splittet, so that author and subject are displayed only on odd pages and the title on even, the pagenumber is displayed on the right side on odd pages and on the left side on even pages.

On the first page, only the date and tutorial will be displayed in the header, the rest of infomration should be in the title.

```

141 \fancypagestyle{firstpage}{
142     %
143     \fancyhf{}
144     % clear all six fields
145     \renewcommand{\headrulewidth}{\hwa@headrulewidth}
146     \renewcommand{\footrulewidth}{0pt}
147     \fancyfoot[R]{\thepage}
148     \fancyhead[L]{\hwa@tutorium}
149     \fancyhead[R]{\@date } }
150 \fancypagestyle{followingpage}{
151     \fancyhf{}
152     \ifhwa@twoside % IF
153     \fancyhead[R0]{\@author}
154     \fancyhead[L0]{\hwa@kurs\

```

```

155 \hwa@tutorium}
156 \fancyhead[LE]{
157 \ifthenelse{\equal{\hwa@sheetTitle}{}}{\hwa@sheetTitle\\}
158 \GetTranslation{abgabe}: \hwa@abgabe
159 }
160 \fancyfoot[R0,LE]{\thepage}
161
162 \else %ELSE
163
164 \fancyhead[R]{\hwa@kurs\\
165 \@author}
166 \fancyhead[L]{\hwa@tutorium\\
167 \ifthenelse{\equal{\hwa@sheetTitle}{}}{\hwa@sheetTitle\\}
168 \GetTranslation{abgabe}: \hwa@abgabe}
169 \fancyfoot[R]{\thepage}
170 \fi %ENDIF
171 \renewcommand{\headrulewidth}{\hwa@headrulewidth}
172 \renewcommand{\footrulewidth}{0pt}
173 }
174 \pagestyle{followingpage}

```

4.2 Enhance Math Environments

A couple of things to make math environments more beautiful and compact.

`\theequation` Displays equation-numbers as upper-case roman numbers.
175 \renewcommand{\theequation}{\Roman{equation}}

`\allowdisplaybreaks` Allow pagebreaks in mathmode
176 \allowdisplaybreaks

4.3 Fonts

4.4 Serif (Default)

The default font for text in HWA is the TeX Gyre clone of Palatino, Pagella. Sans Serif The default sans serif font is TeX Gyre Adventor.

```

177 \DeclareBoolOption[false]{helvet}

```

`helvet` Including helvet in the class options will replace the default sans serif font Aven-
tor with the Helvetica equivalent of TeX Gyre, Heros, for a more metric appere-
ance.

```

178 \ifthenelse{\boolean{xetex}}{
179 \RequirePackage{fontspec}
180 \ifhwa@helvet
181 \setsansfont{TeX Gyre Heros}
182 \setmainfont{TeX Gyre Pagella}
183 \else
184 \setsansfont{TeX Gyre Adventor}

```



```

185 \setmainfont{TeX Gyre Pagella}
186 \fi
187 \setmonofont{Fira Mono}
188 }{
189 \ifhwa@helvet
190 \RequirePackage{mathpazo}
191 \RequirePackage{helvet}
192 \else
193 \RequirePackage{tgadventor}
194 \RequirePackage{tgpagella}
195 \fi
196 \RequirePackage{FiraMono}
197 }

```

4.4.1 Monospace

5 Commands

5.1 Constants

Defines some constants

`\hwa@pointboxsize` Explains it self.
198 `\newcommand{\hwa@pointboxsize}{3em}`

5.2 Document Informations

`\subject`, `\kurs` Sets the subject of the document. Takes the subject as argument. Standard Value is “Kein Kurs”
`\kurs` is deprecated.

```

199 \newcommand{\hwa@kurs}{?\GetTranslation{subject}??} % To store the value
200 \newcommand{\subject}[1]{\renewcommand{\hwa@kurs}{#1}}
201 \newcommand{\kurs}[1]{\subject{#1}}

```

`\tutorial`, `\tutorium` Sets the tutorial of the author. Takes it as an argument. Standard Value is empty, so that this command can be omitted.
`\tutorium` is deprecated.

```

202 \newcommand{\hwa@tutorium}{?\GetTranslation{uebungsgruppe}??} % To store the value
203 \newcommand{\tutorial}[1]{\renewcommand{\hwa@tutorium}{#1}}
204 \newcommand{\tutorium}[1]{\tutorial{#1}}

```

`\deadline`, `\abgabe` Sets the deadline of the document. Takes it as an argument. Standard value is `\today`.
`\abgabe` is deprecated

```

205 \newcommand{\hwa@abgabe}{\today} % To store the value
206 \newcommand{\deadline}[1]{\def\hwa@abgabe{#1}}
207 \newcommand{\abgabe}[1]{\deadline{#1}}

```

`\sheetTitle` Sets a descriptonal Title of the Sheet, will be written in the header of every page.

```

208 \newcommand{\hwa@sheetTitle}{}
209 \newcommand{\sheetTitle}[1]{\def\hwa@sheetTitle{#1}}

```

5.2.1 Inherited from article

`\author` Sets the author of the document.

`\date` Sets the date of the document.

5.3 Sectioning

Because the class is designed for Assignments, the sectioning-commands are different from Article

5.3.1 internal Sectioning

`\hwa@problem`
`\hwa@subproblem`
`\hwa@subsubproblem`

These commands work like their counterpart in article, except that there will be no number, nor will they increase a counter. Nevertheless, they will be shown in the table of contents. With version 4.0 they were made private, because I figured that they are not usefull enough and I am now able to introduce environments with the old names

```

210 \DeclareDocumentCommand\hwa@problem{m o}{\@startsection{problem}%Name
211   {1}%Level
212   {\z@}%indent
213   {-2em \@plus -1em \@minus -1em}%beforeskip
214   {1ex \@plus .5ex}%afterskip
215   {\normalfont\Large\sffamily\bfseries}%style
216   *{#1
217     \IfNoValueF{#2}{
218       \hfill
219       \frame{\framebox[\hwa@pointboxsize]{
220         \hfill \normalfont{\large/\small{#2}}}}
221     }
222   }
223   \addcontentsline{toc}{section}{#1}
224 }
225
226 \DeclareDocumentCommand\hwa@subproblem{m o}{\@startsection{subproblem}%Name
227   {2}%Level
228   {\z@}%indent
229   {-1em \@plus -.5em \@minus -.5em}%beforeskip

```

```

230 { .5ex \@plus .5ex}%afterskip
231 {\normalfont\large\sffamily\bfseries}%style
232 *{#1
233   \IfNoValueF{#2}{
234     \hfill\framebox[\hwa@pointboxsize]{
235       \hfill\normalfont\large/\small{#2}}
236   }
237 }
238 \addcontentsline{toc}{subsection}{#1}
239 }
240
241 \DeclareDocumentCommand\hwa@subsubproblem{m o}{\@startsection{subsubproblem}%Name
242   {3}%Level
243   {\z@}%indent
244   {- .5em}%beforeskip
245   {.5em}%afterskip
246   {\normalfont\sffamily\bfseries}%style
247   *{#1
248     \IfNoValueF{#2}{
249       \hfill\framebox[\hwa@pointboxsize]{
250         \hfill\normalfont\large/\scriptsize{#2}}
251     }
252   }
253 }
254

```

`\keyword` Creates a new Paragraph ,which will start with the Argument in Bold, followed by two non-breaking spaces.

```

255 \newcommand{\keyword}[1]{\@startsection{keyword}%Name
256   {4}%Level
257   {\parindent}%indent
258   {- .1em}%beforeskip
259   {\z@}%afterskip
260   {\normalfont\sffamily\bfseries}%style
261   *{#1~~}
262 }

```

The following Macros make use of `\keyword`, so it is suggested to use them instead.

<code>\solution</code> <code>\proof</code> <code>\given</code> <code>\toShow</code> <code>\toDisprove</code> <code>\assumption</code> <code>\supposeThat</code>	<p>They work like <code>\keyword</code>, but take only an optional Argument print out “Solution”, “Proof” “Given”, “To show”, “Assumption”, and “Suppose that”, respectively ², via <code>\keyword</code>. If an argument is passed, they print out this argument after the keyword. They are not mentioned in the table of contents.</p> <pre> 263 \newcommand{\solution}[1] []{\keyword{\GetTranslation{loesung}\ifstrempy{#1}{~}{~#1}:}} </pre>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

²As of v1.6, Translations are added, depending on the choosen Language, there may be an other Text displayed.
See ?? for all Translations

```

264 \newcommand{\toShow}[1] [] {\keyword{\GetTranslation{zuZeigen}\ifstrempy{#1}{~#1}:}}
265 \newcommand{\toDisprove}[1] [] {
266   \keyword{\GetTranslation{zuWiderlegen}\ifstrempy{#1}{~#1}:}}
267 \newcommand{\given}[1] [] {\keyword{\GetTranslation{gegeben}\ifstrempy{#1}{~#1}:}}
268 \newcommand{\assumption}[1] [] {\keyword{\GetTranslation{Annahme}\ifstrempy{#1}{~#1}:}}
269 \newcommand{\supposeThat}[1] [] {\keyword{\GetTranslation{Angenommen-dass}\ifstrempy{#1}{~#1}:}}

```

5.3.2 ‘better’ Sectioning

The following commands are an augmented version of the “plain” commands.

<pre> \newproblem \newproblem* \newsubproblem \newsbsubproblem </pre>	<p>These commands require no argument, and automatically create a numbered title. They have two optional arguments: <code>\newproblem[#1]{#2}</code> where #1 is the (sub(sub))problem-number and #2 are the points. If there is a number of Points assigned to a (sub(sub))problem, then the command will generate a box to write the reched number of points down next to it.</p>
-----------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Normally, `\newproblem` adds the new Created Problem to the grading-table (see ??), `\newproblem*` does not do this.

They use coutners, of course:

```

270 \newcounter{problem} \setcounter{problem}{0}
271 \newcounter{subproblem}[problem] \setcounter{subproblem}{0}
272 \newcounter{subsubproblem}[subproblem] \setcounter{subsubproblem}{0}
273
274 \DeclareDocumentCommand\newproblem{0{ } g}{
275   \stepcounter{problem}% to reset the lower counters
276   \ifthenelse{\equal{#1}{}}{
277     % empty
278   }{
279     \setcounter{problem}{#1}
280   }
281
282   \IfNoValueTF{#2}{
283     \hwa@problem{\GetTranslation{aufgabe} \hwa@problemno}
284     \addToGradingTable{\# \hwa@problemno}
285   }{
286     \hwa@problem{\GetTranslation{aufgabe} \hwa@problemno}{#2}
287     \addToGradingTable{\# \hwa@problemno}{/#2}
288   }
289 }
290
291 \WithSuffix\newcommand\newproblem*[1] [] {\stepcounter{problem}
292 \ifthenelse{\equal{#1}{}} { } {
293   \stepcounter{problem}% to reset the lower counters
294   \setcounter{problem}{#1}}
295 \hwa@problem{\GetTranslation{aufgabe} \hwa@problemno}

```

```

296 }
297
298 \DeclareDocumentCommand\newsubproblem{0}{ g}{
299   \stepcounter{subproblem}
300   \ifthenelse{\equal{#1}{}}{ } { {
301     \setcounter{subproblem}{#1}}
302   \IfNoValueTF{#2}{
303     \hwa@subproblem{\GetTranslation{aufgabe}
304       \hwa@problemno}{.\hwa@subproblemno}
305   }
306   {
307     \hwa@subproblem{\GetTranslation{aufgabe}
308       \hwa@problemno}{.\hwa@subproblemno}[#2]
309   }
310 }
311
312 \DeclareDocumentCommand\newsubsubproblem{0}{ g}{
313   \stepcounter{subsubproblem}
314   \ifthenelse{\equal{#1}{}}{ } { {\setcounter{subsubproblem}{#1}}
315   \IfNoValueTF{#2}{
316     \hwa@subsubproblem{\hwa@subsubproblemno}}
317   }
318   {
319     \hwa@subsubproblem{\hwa@subsubproblemno}}[#2]
320   }
321 }
322

```

5.3.3 Even Better Sectioning-Environments

h jk

```

323 \NewDocumentEnvironment{problem}{0}{ g}{
324   \newproblem[#1]{#2}
325   \newcommand{\task}[1]{
326     \begin{framed}
327       \keyword{Problem:} ##1
328     \end{framed}
329   }
330 }{}
331 \NewDocumentEnvironment{problem*}{0}{ g}{
332   \newproblem*[#1]{#2}
333   \newcommand{\task}[1]{
334     \begin{framed}
335       \keyword{Problem:} ##1
336     \end{framed}
337   }
338 }{}
339 \NewDocumentEnvironment{subproblem}{0}{ g}{
340   \newsupproblem[#1]{#2}

```

```

341 \newcommand{\task}[1]{
342   \begin{framed}
343     \keyword{Problem:} ##1
344   \end{framed}
345 }
346 {}
347 \NewDocumentEnvironment{subsubproblem}{0}{ g}{
348   \newsbssubproblem[#1]{#2}
349   \newcommand{\task}[1]{
350     \begin{framed}
351       \keyword{Problem:} ##1
352     \end{framed}
353   }
354 {}

```

5.4 Useful Macros

5.4.1 QUOD ERAT DEMUNSTRANDUM, End of Proof

`\QED`
`\EOP` Display a flushed-right *QED*, \square , or \blacksquare , respectively. `\qed` is not implemented, to
`\eop` keep compatibility to several Math-packages, which define the later.

```

355 \newcommand{\hwa@QED}{\begin{flushright}
356   \textsc{Qed}
357 \end{flushright}
358 }
359 \newcommand{\QED}{\hwa@QED}
360
361 \ifhwa@unicodemath
362 \RequirePackage{unicode-math}
363 \AtBeginDocument{\let\QEDSymbol\QED}
364 \renewcommand{\QED}{\hwa@QED}
365 }
366 \fi
367
368 \newcommand{\EOP}{\begin{flushright}
369   \(\square\)
370 \end{flushright}
371 }
372 \newcommand{\eop}{\hfill\(\blacksquare\)}

```

5.4.2 QUOD NON ERAT DEMUNSTARNDUM AT IUCUNDUM EST

`\QNED`
`\qned` Display a right-flushed *triangle*. `\QNED` displays it in a new line, `\qned` at the end
of the same line.
In Mathematical proofs this symbol is used to mark things, which we did not

intend to proof, but are interesting anyway or things wick are not proofed mathematically, but are explained in a way which lets no doubt on their correctness.

```
373 \newcommand{\QNEd}{\begin{flushright} \(\triangleleft\)}
374 \end{flushright}
375 }
376 \newcommand{\qned}{\hfill\(\triangleleft\)}
```

5.4.3 Stolen Goods

»Das ist alles nur geklaut«

~Tobias Künzel

These Commands are not mine, they are all stolen from Alexander Bartolomey's³ amath-Class⁴

\N			
\Z	Defines a set of mathematical sets, which are verry usefull (see ??)		
\R			
\Q	Command	Output	Description
\Compl	\N	N	Natural Numbers
\F	\Z	Z	Whole Numbers
\Primes	\Q	Q	Rational Numbers
	\R	R	Real Numbers
	\Compl	C	Complex Numbers
	\F_n	F _n	Prime Field to base n
	\Primes ⁵	P	Set of all Primes

Table 1: Field-Commands

```
377 \newcommand{\N}{\ensuremath{\mathbb{N}}}
378 \newcommand{\Z}{\ensuremath{\mathbb{Z}}}
379 \newcommand{\R}{\ensuremath{\mathbb{R}}}
380 \newcommand{\Q}{\ensuremath{\mathbb{Q}}}
381 \newcommand{\Compl}{\ensuremath{\mathbb{C}}}
382 \newcommand{\F}{\ensuremath{\mathbb{F}}}
383 % The last one is mine
384 \newcommand{\Primes}{\ensuremath{\mathbb{P}}}
```

\GL	
\id	Output useful plaintext operators and functions. See table ??.
\Var	Require
\Perm	mathmode
\MComb	
\Comb	³ "Occloxiium" on GitHub: https://github.com/occlxiium
\Pot	⁴ amath.sty is part of Alexander Bartolomey's Alphabet Classes: https://github.com/occlxiium/AlphabetClasses
\Map	⁵ Has to be \Primes, because \P is already in use
\Hom	
\Ker	
\Intpol	15
\Pol	
\Sol	
\Bin	
\charakteristik	
\fo	
\first	
\la	

Command	Output
<code>\GL</code>	GL
<code>\id</code>	id
<code>\Var</code>	Var
<code>\Perm</code>	Perm
<code>\Comb</code>	Comb
<code>\MComb</code>	MComb
<code>\Pot</code>	Pot
<code>\Map</code>	Map
<code>\Hom</code>	Hom
<code>\Intpol</code>	Intpol
<code>\Pol</code>	Pol
<code>\Sol</code>	Sol
<code>\Bin</code>	Bin
<code>\charakteristik</code>	char
<code>\diff{<1>}</code>	$\frac{d}{d<1>}$
<code>\partdiff{<1>}</code>	$\frac{\partial}{\partial<1>}$
<code>\divides and property</code>	Prints a vertical line
<code>\dx</code>	dx
<code>\excup</code>	$\dot{\cup}$
<code>\fo</code>	fo
<code>\first</code>	fi
<code>\la</code>	la

Table 2: Common Functions

`\falls` prints out »falls«⁶

```

385 \DeclareMathOperator{\GL}{GL}
386 \DeclareMathOperator{\id}{id}
387 \DeclareMathOperator{\Var}{Var}
388 \DeclareMathOperator{\Perm}{Perm}
389 \DeclareMathOperator{\MComb}{MComb}
390 \DeclareMathOperator{\Comb}{Comb}
391 \DeclareMathOperator{\Pot}{Pot}
392 \DeclareMathOperator{\Map}{Map}
393 \DeclareMathOperator{\Hom}{Hom}
394 \DeclareMathOperator{\Ker}{Ker}
395 \DeclareMathOperator{\Intpol}{Intpol}
396 \DeclareMathOperator{\Pol}{Pol}
397 \DeclareMathOperator{\Sol}{Sol}
398 \DeclareMathOperator{\Bin}{Bin}
399 \DeclareMathOperator{\charakteristik}{char}
400 \DeclareMathOperator{\fo}{fo}
401 \DeclareMathOperator{\first}{fi}
402 \DeclareMathOperator{\la}{la}

```

⁶In German, actual Translation may differ


```

403
404 \newcommand{\diff}[1]{\ensuremath{\frac{d}{d#1}}}
405 \newcommand{\partdiff}[1]{\ensuremath{\frac{\partial}{\partial #1}}}
406 \newcommand{\dx}{\:dx}
407 \newcommand{\divides}{\ensuremath{\mid}}
408 \newcommand{\property}{\ensuremath{\imath}}
409
410 \renewcommand{\dim}[1][ ]{\ensuremath{\text{dim}_{#1}}}
411 \renewcommand{\Im}{\ensuremath{\text{Im}}}
412
413 \newcommand{\excup}{\ensuremath{\stackrel{\cup}{\sim}}}
414 \newcommand{\falls}{\text{\ \GetTranslation{falls}}\ }

```

5.4.4 Rounding

Require mathmode

Command	Output	Meaning
<code>\floor{<1>}</code>	$\lfloor <1> \rfloor$	floor <1>
<code>\ceil{<1>}</code>	$\lceil <1> \rceil$	ceil <1>
<code>\roundHU{<1>}</code>	$\lceil <1> \rceil$	Round <1> “half up” ($\lceil <1> + \frac{1}{2} \rceil$)
<code>\roundHD{<1>}</code>	$\lfloor <1> \rfloor$	Round <1> “half down” ($\lfloor <1> - \frac{1}{2} \rfloor$)

Table 3: Rounding Functions

```

415 \newcommand{\floor}[1]{\ensuremath{\left\lfloor #1 \right\rfloor}}
416 \newcommand{\ceil}[1]{\ensuremath{\left\lceil #1 \right\rceil}}
417 \newcommand{\roundHU}[1]{\ensuremath{\left\lceil #1 \right\rceil}}
418 \newcommand{\roundHD}[1]{\ensuremath{\left\lfloor #1 \right\rfloor}}

```

`\bigforall`

`\bigexists` Redefines big versions of quantors, adds an h-skip to normal version.

```

419 \let\oforall\forall
420 \let\oexists\exists
421 \renewcommand{\forall}{\hspace{2pt}\oforall\hspace{2pt}}
422 \renewcommand{\exists}{\hspace{2pt}\oexists\hspace{2pt}}
423 \newcommand{\bigforall}{\mbox{\raisebox{-2pt}{\Large $\mathsurround{4pt}\forall$}}}
424 \newcommand{\bigexists}{\mbox{\raisebox{-2pt}{\Large $\mathsurround{4pt}\exists$}}}

```

5.4.5 ToDos

Utility for the Documentation of ToDos

`\todo` Creates a todo at the location of the command, highlighted in red. The ToDos will be listed after maketitle, unless the option todos=nolist or todos=none is specified.

```

425 \DeclareDocumentCommand\todo{G}{\ }

```

```

426 \ifthenelse{\boolean{hwa@todos@inplace}}{
427   {\color{red}\textbf{~\label{TODO\alph{todoNum}}TODO~\#1~}
428   \xdef\hwa@todoList@aux{\hwa@todoList@aux
429     \string\item\string\hyperref[TODO\alph{todoNum}]{TODO \#1}
430   }
431   \stepcounter{todoNum}
432 }{}
433 }

```

Uses the internal hwa@todo-counter

```

434 \newcounter{todoNum} \setcounter{todoNum}{1}

```

`\hwa@tableOfTodos` Prints all ToDos

```

435 \DeclareDocumentCommand\hwa@tableOfTodos{}{
436   \ifthenelse{\boolean{hwa@todos@list}}{
437     \ifthenelse{\equal{\hwa@todoList}{}}{\%Nothing
438     }{
439       {\color{red}
440         \hwa@problem{Table of ToDos}
441         \begin{itemize}
442           \hwa@todoList
443         \end{itemize}}
444     }
445   }{}
446 }

```

5.5 Grading Table

This Document-Class is still mainly designed for Homework, so it would be nice, if there was a table to write Grades into, wouldn't it?

`\addToGradingTable` Adds the given parameter as an exercise to the Grading-Table. All Problems, created with `\newproblem` are added automatically.

```

447 \DeclareDocumentCommand\addToGradingTable{m g}{
448   \edef\hwa@gradingtbl@aux@defs{\hwa@gradingtbl@aux@defs|p{\hwa@pointboxsize}}
449   \edef\hwa@gradingtbl@aux@lineOne{\hwa@gradingtbl@aux@lineOne{\#1} &}
450   \IfNoValueTF{\#2}{
451     \edef\hwa@gradingtbl@aux@lineTwo{\hwa@gradingtbl@aux@lineTwo &}
452   }{
453     \edef\hwa@gradingtbl@aux@lineTwo{\hwa@gradingtbl@aux@lineTwo\vfill\hfill
454       {\string\small \#2} &}
455   }
456 }

```

`\makeGradingTable` Outputs a table to fill in the reached Points. Only shows Problems created by `\newproblem`.

Displays the according number of maximum points for each problem, if specified.
Displays the total number of maximum Problems, if given by Argument Like

`\tableofcontent`, it needs a second run of \LaTeX , until all are added.
It will never overflow the Line-Width, thanks to an `adjustbox`. [#1]: *Optional*.
The total number of points reachable.

```

457 \DeclareDocumentCommand\makeGradingTable{o}{
458   \begin{table}[hb]
459     \centering
460     \large
461     \begin{adjustbox}{max width=\linewidth}
462       \expandafter\table\expandafter{\hwa@gradingtbl@defs ||p{\hwa@pointboxsize}}\hline
463       \hwa@gradingtbl@lineOne \(\Sigma\) \\ \hline \small
464       \hwa@gradingtbl@lineTwo \IfNoValueTF{#1}{~}{\vfill\hfill/#1}\vspace{.15cm}\\ \hline
465     \end{table}
466   \end{adjustbox}
467 \end{table}
468 }
```

See example documents for output

5.5.1 Internal commands

`\hwa@gradingtbl@...` Defines macros whose contents will be written to the AUX-File and read in the next run, and the usable commands. The later will contain the information, but have to be defined (incase the aux-file does not exist)

```

469 \edef\hwa@gradingtbl@aux@defs{}
470 \newcommand{\hwa@gradingtbl@aux@lineOne}{}
471 \newcommand{\hwa@gradingtbl@aux@lineTwo}{}
472
473 \edef\hwa@gradingtbl@defs{}
474 \newcommand{\hwa@gradingtbl@lineOne}{}
475 \newcommand{\hwa@gradingtbl@lineTwo}{}

```

`\hwa@todoList@...` See `\hwa@gradingtbl@...`

```

476 \newcommand{\hwa@todoList}{}
477 \newcommand{\hwa@todoList@aux}{}
478 % \end{macro}
479 % \begin{macro}{\write\@auxout}
480 %   Write to aux
481 %   \begin{macrocode}
482 \AtEndDocument{%
483   \immediate\write\@auxout{%
484     \gdef\string\hwa@gradingtbl@defs{\hwa@gradingtbl@aux@defs}
485   }
486   \immediate\write\@auxout{%
487     \gdef\string\hwa@gradingtbl@lineOne{\hwa@gradingtbl@aux@lineOne}%
488   }
489   \immediate\write\@auxout{%
490     \gdef\string\hwa@gradingtbl@lineTwo{\hwa@gradingtbl@aux@lineTwo}%
491   }
492   \immediate\write\@auxout{%

```

```

493 \gdef\string\hwa@todoList{\hwa@todoList@aux}%
494 }
495 }

```

5.6 Title

`\maketitle` Overrides `\maketitle`.

```

496 \renewcommand{\maketitle} {
497   \thispagestyle{firstpage}
498   \ifhwa@twocolumn{
499     \twocolumn[{
500       \hwa@maketitletext
501     }]
502   }\else{
503     \hwa@maketitletext
504   }\fi
505   \hwa@tableOfTodos
506 }

```

`\hwa@maketitletext` Prints out the title with author etc. Used to reduce code duplication for two- and onecolumn styles

```

507 \newcommand{\hwa@maketitletext}{
508   \begin{centering}
509     \huge{\textsf{\textbf{\hwa@kurs}}}\hwa@hline@LONE \large
510     \ifthenelse{\equal{\hwa@sheetTitle}{}}{\textbf{\hwa@sheetTitle}}{\}
511     \GetTranslation{abgabe}: \hwa@abgabe\
512     \hwa@hline@LTW0
513     \normalsize{\@author}\
514     \hwa@hline@LTW0 \normalsize
515   \end{centering}
516 }
517 \ifthenelse{\boolean{hwa@punchmark}}{
518   \newcommand{\hwa@punchmarkRad}{3mm}
519   \newcommand{\hwa@punchmarkDistanceX}{12mm}
520   \newcommand{\hwa@punchmarkDistanceY}{40mm}
521   \AtBeginDocument{
522     % Where will the punch be?
523     \AddToShipoutPictureBG*{\AtPageUpperLeft{
524       \put(\LenToUnit{\hwa@punchmarkDistanceX}-\hwa@punchmarkRad*2),\LenToUnit{-.5\paperheight}-\
525       \put(\LenToUnit{\hwa@punchmarkDistanceX}-\hwa@punchmarkRad*2),\LenToUnit{-.5\paperheight}+\
526       % Punch-Positioningmark
527       \AddToShipoutPictureBG*{\AtPageUpperLeft{
528         \put(\LenToUnit{5mm},\LenToUnit{-.5\paperheight}){\tikz{\draw (0,0) -- (5mm,0);}}}
529     }
530 }{
531 }

```

5.7 Counters

The actual counters are defined in ??.

Counter-Commands These are used to output the Exercise numbers in the desired style

```
532 \newcommand{\hwa@problemno}{\arabic{problem}}
533 \newcommand{\hwa@subproblemno}{\alph{subproblem}}
534 \newcommand{\hwa@subsubproblemno}{\roman{subsubproblem}}
```

\hwa@parseCounterStyle This takes a style-input (#1), one of the three previous defined commands (#2) and the corresponding counter (#3) to redefine #1, so that it corresponds to #2. See ?? for example usement.

```
535 \newcommand{\hwa@parseCounterStyle}[3]{
536   \ifthenelse{equal{#1}{arabic}}{ \renewcommand{#2}{\arabic{#3}} }{
537     \ifthenelse{equal{#1}{roman}}{ \renewcommand{#2}{\roman{#3}} }{
538       \ifthenelse{equal{#1}{alph}}{ \renewcommand{#2}{\alph{#3}} }{
539         \ifthenelse{equal{#1}{Alph}}{ \renewcommand{#2}{\Alph{#3}} }{
540           \ifthenelse{equal{#1}{Roman}}{
541             \renewcommand{#2}{\Roman{#3}} }{
542             \ClassError{homeworkassignment}{Invalid Value #1 for
543               option Counter-Styling}{Possible Values are alph,
544               arabic, Arabic, roman or Roman.} } } } } }
```

Redefines the three counter-commands:

```
545 \hwa@parseCounterStyle{\hwa@problemsty}{\hwa@problemno}{problem}
546 \hwa@parseCounterStyle{\hwa@subproblemsty}{\hwa@subproblemno}{subproblem}
547 \hwa@parseCounterStyle{\hwa@subsubproblemsty}{\hwa@subsubproblemno}{subsubproblem}
```

6 Environments

6.1 Proof

Used for proofes. Starts bth proof and ends with a End-Of-Proof symbol.

```
548 \NewDocumentEnvironment{proof}{G}{\GetTranslation{beweis}} 0{\QED}}
549 {
550   \keyword{#1:~~}
551 }
552 {
553   #2
554 }
```

6.2 Proof by contradiction

Used for proofes. Starts bth proof and ends with a End-Of-Proof symbol.

```
555 \NewDocumentEnvironment{contradiction}{}
556 {
557   \begin{proof}{\GetTranslation{beweis}~\GetTranslation{per}~\GetTranslation{Widerspruch}}[\hfi
```

```
558 }  
559 {  
560   \end{proof}  
561 }
```

7 Development and support

The package is developed at *GitHub*:

<https://github.com/ACHinrichs/LaTeX-templates>

Please refer to that site for any bug report or development information.

8 Changelog

v1.0 - 2016/10/23 Initial

v1.1 - 2016/11/02 ...

v1.2 - 2016/11/03 ...

v1.3 - 2016/12/01 • Provide the Class as .dtx file and more

v1.4 - 2017/04/29 • “Minor” bugfixes

v1.5 - 2017/04/29 • Problems are displayed in the table of contents. Type of numeration is now configurable.

v1.5.1 - 2017/04/29 • Bugfix

v1.5.2 - 2017/04/29 • Add version-number

v1.6 - 2017/05/02 • Add Translations (German and English)

- Add `\given` and `\toShow`
- Add `\QED`, `\EOP`, and `\eop`

v1.6.3 - 2017/05/05

- Bugfixes

v1.6.4 - 2017/05/09 • Change `\eop` to be in the same line

v1.7 - 2017/05/09 • Add \QNEd

v2.0 - 2017/05/23 "Layout 2.0"

- Change Margins
- Add Option to select older Page-Style
- Change standardlayout to twocolumn and twoside
- ~~Steal~~ Use Macros by Alexander Bartolomey (See ??)
- Add some TikZ-Styles
- Add round functions

v2.2 - 2017/06/17 • Add Grading-table

- Add `\keyword`, `\assumption`, and `\supposeThat`

- Add `\newproblem*`
 - Add `\sheetTitle`
 - Change equation-numbering to uppercase roman
- v2.2.1 - 2017/06/20** • Fix error with commands like `\solution` and `\keyword`.
- v2.4 - 2017/04/07** • Fix math alignment
- Add option for flushed left equations
 - Update amath port to use
- v3.0 - 2017/12/26** “WS 2017”
- Rename to `homeworkassignment`
 - Add Environment for various proofs
 - Add points for exercises and a place to fill them in
 - Add option to remove or decrease or remove the hlines
 - Remove legacy styles
 - Rework the documentation
 - Beautify Maths
 - Fix OneColumn-Maktitle-Bug
 - Fix Subproblem-Counter not being reset
 - Merry Christmas!
- v3.2 - 2018/12/06** Nikolaus Release
- Make XeLaTeX-Compatible
 - Fix `\newproblem` requiring a Problem-Number
 - Add `\toDisprove` macro similar to the `\toShow` macro
 - Add option for punchmarks
 - Add option to load unicode-math and work around a incompatibility
- v4.0 - 2019/04/17** Pre-Easter Release
- Add option to use `unicode-math`
 - Rename `\C` to `\Comp1`, because of a `unicode-math` incompatibility
 - Rework Fonts
 - Make `onecolumn` the default
 - Implement `ToDo`s
 - Replace `\problem`-commands by `problem` environments (which behave like `\newproblem`)
 - Add `Task-Command`
- v4.1 - 2019/05/03** Experimental Font Support
- Included `helvet` binary option for a more metric font support
 - Language and grammar fixes

8.1 Version-Scheme

Since Version 2.0 the following version-scheme applies:

Major Version has to be increased, if

- there are changes, which create visible changes in the output of existing documents (except for bugfixes), or
- a command is removed or changed in a way, that existing documents do not compile with the new version.

Minor Version has to be increased, if

- new backwards compatible commands are introduced
 - Bugfixes may be introduced too.

The minor version of stable releases is always even, the minor version of development versions is always odd. (0 counts as even).

Patches May be introduced on Stable Branch. With every non-document-breaking bugfix, the patch-number has to be incremented.

Because Fixing Bugs is a part of development, development-versions do not have numeric patch-numbers, but alphabetic identifiers, directly after the minor-version.

9 Translations

Homeworkassignment currently supports English and German, fallback language is German. Unfortunately these two are the only Languages I am capable of translating reliable, so if you want to use an other language, I would be very happy if you would help me to translate homeworkassignment! Please open an issue, author a pull-request or send me an e-mail.

```
562 \DeclareTranslationFallback{aufgabe}{Aufgabe}
563 \DeclareTranslationFallback{loesung}{L\"osung}
564 \DeclareTranslationFallback{beweis}{Beweis}
565 \DeclareTranslationFallback{uebungsgruppe}{\"Ubungsgruppe}
566 \DeclareTranslationFallback{abgabe}{Abgabe}
567 \DeclareTranslationFallback{zuZeigen}{Zu zeigen}
568 \DeclareTranslationFallback{zuWiderlegen}{Zu widerlegen}
569 \DeclareTranslationFallback{gegeben}{Gegeben}
570 \DeclareTranslationFallback{falls}{falls}
571 \DeclareTranslationFallback{Annahme}{Annahme}
572 \DeclareTranslationFallback{Angenommen-dass}{Angenommen, dass}
573 \DeclareTranslationFallback{per}{per}
574 \DeclareTranslationFallback{Widerspruch}{Widerspruch}
575
576 \DeclareTranslation{German}{aufgabe}{Aufgabe}
577 \DeclareTranslation{German}{loesung}{L\"osung}
578 \DeclareTranslation{German}{beweis}{Beweis}
579 \DeclareTranslation{German}{uebungsgruppe}{\"Ubungsgruppe}
580 \DeclareTranslation{German}{abgabe}{Abgabe}
581 \DeclareTranslation{German}{zuZeigen}{Zu zeigen}
582 \DeclareTranslation{German}{zuWiderlegen}{Zu widerlegen}
583 \DeclareTranslation{German}{gegeben}{Gegeben}
584 \DeclareTranslation{German}{falls}{falls}
585 \DeclareTranslation{German}{Falls}{Falls}
586 \DeclareTranslation{German}{Annahme}{Annahme}
587 \DeclareTranslation{German}{Angenommen-dass}{Angenommen, dass}
588 \DeclareTranslation{German}{per}{per}
589 \DeclareTranslation{German}{Widerspruch}{Widerspruch}
590
591 \DeclareTranslation{English}{aufgabe}{Problem}
592 \DeclareTranslation{English}{loesung}{Solution}
593 \DeclareTranslation{English}{beweis}{Proof}
594 \DeclareTranslation{English}{uebungsgruppe}{Tutorial}
595 \DeclareTranslation{English}{abgabe}{Deadline}
596 \DeclareTranslation{English}{zuZeigen}{To show}
597 \DeclareTranslation{English}{zuWiderlegen}{To disprove}
598 \DeclareTranslation{English}{gegeben}{Given}
599 \DeclareTranslation{English}{falls}{if}
600 \DeclareTranslation{English}{Falls}{If}
601 \DeclareTranslation{English}{Annahme}{Assumption}
602 \DeclareTranslation{English}{Angenommen-dass}{Suppose that}
603 \DeclareTranslation{English}{per}{by}
```

604 \DeclareTranslation{English}{Widerspruch}{contradiction}

End

The End

605 \endinput