

The HomeworkAssignment class*

Adrian C Hinrichs
adrian.hinrichs@rwth-aachen.de

June 17, 2017

Contents

1	Abstract	2
2	Options	2
2.1	Inherited options	2
3	Commands	2
3.1	Document Informations	2
3.1.1	Inherited from <code>article</code>	3
3.2	Sectioning	3
3.2.1	‘plain’ Sectioning	3
3.2.2	‘better’ Sectioning	3
3.3	Useful Macros	4
3.3.1	QUOD ERAT DEMUNSTARNDUM, End of Proof	4
3.3.2	QUOD NON ERAT DEMUNSTARNDUM AT IUCUNDUM EST	4
3.3.3	Stolen Goods	4
3.3.4	Rounding	5
3.4	Grading Table	5
4	Development and support	6
5	Changelog	6
6	Examples	7
7	Implementation	7
7.1	Packages & Options	7
7.2	TikZ-Styles	8
7.3	Geometry	8
7.4	Translations	9
7.5	Headers & Footers	10

*This document corresponds to HomeworkAssignment v2.2,dated 2017/06/17.

8	Redefinition of existing Commands	11
8.1	Internal commands	11
8.1.1	Counter-Commands	11
8.1.2	Counter-Style Parser	11
8.1.3	Counter-Commands II	11
8.1.4	Grading-table	12
8.2	Commands	12

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 Options

<code>problemstyle=<1></code>	These options allow the customizatuion of the displayed numbers. For Example, if <code>problemstyle=Roman</code> , <code>subproblemstyle=arabic</code> , <code>subsubproblemstyle=roman</code> is passed, The first subsubproblem of the first subproblem of the first problem would be labled as i) of Problem I.1 .
<code>subproblemstyle=<1></code>	Available options are <code>arabic</code> , <code>Alph</code> , <code>alph</code> , <code>Roman</code> , and <code>roman</code> . Standard values are: <code>problemstyle=arabic</code> , <code>subproblemstyle=alph</code> , <code>subsubproblemstyle=roman</code> .
<code>subsubproblemstyle=<1></code>	Allows the User to select an older page-style, for backwards compatibility. Recognized values are <code>v1</code> and <code>v2</code> . Everytime a version Changes the default look, a new possible value will be added. Only set this if you really need to get an old look, <i>older styles are not going to be maintained!</i>
<code>design=<1></code>	
<code>tikz</code>	Loads TikZ-Package and a couple of Styles, usefull for Papers in Computer-Science and ;athematics. See 7.2 for more informations

2.1 Inherited options

Because the class is inherited by `article`, every Option that can be passed to `article`, will be passed to `article`.

3 Commands

3.1 Document Informations

<code>\subject</code>	Sets the subject of the document. Takes the subject as argument. Standard Value is "Kein Kurs"
<code>\kurs</code>	<code>\kurs</code> is deprecated.
<code>\tutorial</code>	Sets the tutorial of the author. Takes it as an argument. Stamdard Value is
<code>\tutorium</code>	

empty, so that this command can be omitted.
`\tutorium` is deprecated.

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

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

3.1.1 Inherited from article

`\author` Sets the author of the document.
`\date` Sets the date of the document.

3.2 Sectioning

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

3.2.1 ‘plain’ Sectioning

`\problem` These commands work like theyr counterpart in article, except that there will be
`\subproblem` no number, nor will they increase a counter. Nevertheless, hey will be shown in
`\subsubproblem` the table of contents.

`\keyword{#1}` Creates a new Paragraph ,which will start with the Argument in Bold, followed
by two non-breaking spaces.
The following Macros make use of `\keyword`, so it is suggested to use them instead.

`\solution` They work like `\keyword`, but take only an optional Argument print out “So-
`\proof` lution”, “Proof” “Given”, “To show”, “Assumption”, and “Suppose that”, respec-
`\given` tively ¹, via `\keyword`. If an argument is passed, they print out this argument
`\toShow` after the keyword. They are not mentioned in the table of contents.
`\assumption`
`\supposeThat`

3.2.2 ‘better’ Sectioning

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

`\newproblem` These commands require no argument, and automatically create a numbered
`\newproblem*` title. The optional Argument is the new value for the coresponding counter.
`\newsproblem` Normally, `\newproblem` adds the new Created Problem to the grading-table (see
`\newsproblem` 3.4), `\newproblem*` does not do this.

¹As of v1.6, Translations are added, depending on the choosen Language, there may be an other Text displayed.
See 7.4 for all Translations

3.3 Useful Macros

3.3.1 Quod Erat Demunstarndum, End of Proof

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

3.3.2 Quod Non Erat Demunstarndum at iucundum est

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

3.3.3 Stolen Goods

»Das ist alles nur geklaut«

~Tobias Künzel

These Commands are not mine, they are all stolen from Alexander Bartolomey's² `amath-Class`³

<code>\N</code>	Defines a set of mathematical sets, which are verry usefull (see Table 1)		
<code>\Z</code>			
<code>\R</code>	Command	Output	Description
<code>\Q</code>	<code>\N</code>	\mathbb{N}	Natural Numbers
<code>\C</code>	<code>\Z</code>	\mathbb{Z}	Whole? Numbers
<code>\F</code>	<code>\Q</code>	\mathbb{Q}	Rational Numbers
<code>\Primes</code>	<code>\R</code>	\mathbb{R}	Real Numbers
	<code>\C</code>	\mathbb{C}	Complex Numbers
	<code>\F</code>	\mathbb{F}	Prime-Fieled?
	<code>\Primes</code> ⁴	\mathbb{P}	Set of all Primes

Table 1: Field-Commands

Functions and Operators Output usefull Plaintext-Operators and Functions. See table 2. Require Math-mode

	Command	Output
<code>\divides</code>	and <code>property</code>	Prints a vertical Bar
	<code>\Var</code>	Var
	<code>\Perm</code>	Perm
	<code>\Comb</code>	Comb
	<code>\MComb</code>	MComb

²“Occloxiium” on GitHub:<https://github.com/occloxiium>

³`Amath.sty` is part of Alexander Bartolomey's Alphabet Classes: <https://github.com/occloxiium/AlphabetClasses>

⁴Has to be `\Primes`, because `\P` is already in use

<code>\Im</code>	Im
<code>\Pot</code>	Pot
<code>\Map</code>	Map
<code>\Bin</code>	Bin
<code>\GL</code>	GL
<code>\id</code>	id
<code>\dx</code>	dx
<code>\excup</code>	$\dot{\cup}$
<code>\dim[<1>]</code>	$\dim_{<1>}$
<code>\diff{<1>}</code>	$\frac{d}{d<1>}$

Table 2: Text-like Functions

`\falls` prints out $\ggfalls\ll$ ⁵

3.3.4 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” ($\lfloor <1> + \frac{1}{2} \rfloor$)
<code>\roundHD{<1>}</code>	$\lfloor <1> \rfloor$	Round $<1>$ “half down” ($-\lfloor <1> - \frac{1}{2} \rfloor$)

Table 3: Rounding Functions

3.4 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?

<code>\addToGradingTable</code>	Adds the given parameter as an exercise to the Grading-Table. All Problems, created with <code>\newproblem</code> are added automatically.
<code>\makeGradingTable</code>	Prints out the Table containig all Defined exercises (\neq Problems). Like <code>\tableofcontent</code> , it needs a second run of L ^A T _E X, until all are added. See example documents fot output

⁵In German, actual Translation may differ

4 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.

5 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 `\QNEED`

v2.0 - 2017/05/23 Change Margins,

Add Option to select older Page-Style,

Change standardlayout to twocolumn and twoside

~~Steal~~ Use Macros by Alexander Bartolomey (See 3.3.3)

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

Note: Since Version 2.0 the minor version of releases is always even, the minor version of unstable versions is always odd. There are no unstable patch-versions, instead unstable versions get an alphabetical suffix.

6 Examples

For examples, please see the Git-Repo at <https://github.com/ACHinrichs/LaTeX-templates>

7 Implementation

The following part is very boring, but I have not found a solution to create a .cls-file without including the implementation into the document. Loads L^AT_EX₂ε and sets the Version Loads the `article`, which is the base-class.

7.1 Packages & Options

```
1 \RequirePackage{kvoptions}
2 \SetupKeyvalOptions{ family=hwa,
3   prefix=hwa@ }
4 \DeclareStringOption[arabic]{problemsty}
5 \DeclareStringOption[alph]{subproblemsty}
6 \DeclareStringOption[roman]{subsubproblemsty}
7 \DeclareBoolOption[false]{listings}
8 \DeclareStringOption[v2]{design}
9 \DeclareBoolOption[true]{twoside}
10 \DeclareComplementaryOption{oneside}{twoside}
11 \DeclareBoolOption[true]{twocolumn}
12 \DeclareComplementaryOption{onecolumn}{twocolumn}
13 \DeclareBoolOption[false]{tikz}
14 % Redefine the article-options
15 % \begin{macrocode}
16 \DeclareDefaultOption{\PassOptionsToClass{\CurrentOptionKey}{article}}

    Processes the Options and loads article
17 \ProcessKeyvalOptions*
18 \ifhwa@twoside
19 \PassOptionsToClass{twoside}{article}
20 \else
21 \PassOptionsToClass{oneside}{article}
22 \fi
23 \ifhwa@twocolumn
24 \PassOptionsToClass{twocolumn}{article}
25 \else
26 \PassOptionsToClass{onecolumn}{article}
27 \fi
28 \LoadClass{article}

    Loads required Packages
29 \RequirePackage{suffix}
30 \RequirePackage{fancyhdr}
31 \RequirePackage{ifthen}
32 \RequirePackage{translations}
```

```

33 \PassOptionsToPackage{fleqn}{amsmath}
34 \RequirePackage{amsmath}
35 \RequirePackage{amssymb}
36 \ifhwa@listings
37 \RequirePackage{listings}
38 \lstset{
39   frame = single,
40   breaklines = true,
41   postbreak=\raisebox{0ex}[0ex][0ex]{\ensuremath{\hookrightarrow\space}},
42   basicstyle=\scriptsize
43 }
44 \else
45 \empty
46 \fi
47 \RequirePackage{etoolbox}
48 \RequirePackage{array}

```

7.2 TikZ-Styles

If tikz is Wanted, load Usefull Styles

```

49 \ifhwa@tikz
50 \RequirePackage{tikz}
51 \usetikzlibrary{shapes,arrows,positioning,decorations,
52   automata,backgrounds,petri,bending,
53   shapes.multipart}
54 \tikzset{
55   treenode/.style = {shape=circle, rounded corners,
56     draw, align=center},
57   graynode/.style = {fill=gray},
58   normalnode/.style = {treenode, font=\Large, bottom color=white},
59   array/.style = {rectangle split,
60     rectangle split horizontal,
61     rectangle split,
62     draw}
63 }
64 \fi

```

7.3 Geometry

Make sure that this is the last Package loaded

```

65 % Make sure that this is the last Package loaded
66 \ifthenelse{\equal{\hwa@design}{v2}}{
67   \RequirePackage{geometry}
68   \ifhwa@twocolumn
69     \geometry{top=2cm, bottom=2cm, left=2cm,
70       headsep=14pt,hmarginratio={1:1}}
71   \else
72     \geometry{top=2cm, bottom=2cm, width=35em,
73       headsep=14pt,hmarginratio={4:3}}
74   \fi

```



```

75 }{
76   \ifthenelse{\equal{\hwa@design}{v1}}{
77     \empty
78   }{
79     \ClassError{HomeworkAssignment}{Value '\hwa@design' for key 'design'
80       is not known}{The option design takes an argument to set the
81       Pagestyle to the one of a previous version. Acceptable values are
82       'v1', or 'v2'}
83   }
84 }

```

7.4 Translations

Load translations, currently supports English and German, Fallback is German

```

85 \DeclareTranslationFallback{aufgabe}{Aufgabe}
86 \DeclareTranslationFallback{loesung}{L"osung}
87 \DeclareTranslationFallback{beweis}{Beweis}
88 \DeclareTranslationFallback{uebungsgruppe}{\ "Ubungsgruppe}
89 \DeclareTranslationFallback{abgabe}{Abgabe}
90 \DeclareTranslationFallback{zuZeigen}{Zu zeigen}
91 \DeclareTranslationFallback{gegeben}{Gegeben}
92 \DeclareTranslationFallback{falls}{falls}
93 \DeclareTranslationFallback{Annahme}{Annahme}
94 \DeclareTranslationFallback{Angenommen-dass}{Anngenommen, dass}
95
96 \DeclareTranslation{German}{aufgabe}{Aufgabe}
97 \DeclareTranslation{German}{loesung}{L"osung}
98 \DeclareTranslation{German}{beweis}{Beweis}
99 \DeclareTranslation{German}{uebungsgruppe}{\ "Ubungsgruppe}
100 \DeclareTranslation{German}{abgabe}{Abgabe}
101 \DeclareTranslation{German}{zuZeigen}{Zu zeigen}
102 \DeclareTranslation{German}{gegeben}{Gegeben}
103 \DeclareTranslation{German}{falls}{falls}
104 \DeclareTranslation{German}{Falls}{Falls}
105 \DeclareTranslation{German}{Annahme}{Annahme}
106 \DeclareTranslation{German}{Angenommen-dass}{Anngenommen, dass}
107
108 \DeclareTranslation{English}{aufgabe}{Problem}
109 \DeclareTranslation{English}{loesung}{Solution}
110 \DeclareTranslation{English}{beweis}{Proof}
111 \DeclareTranslation{English}{uebungsgruppe}{Tutorial}
112 \DeclareTranslation{English}{abgabe}{Deadline}
113 \DeclareTranslation{English}{zuZeigen}{To show}
114 \DeclareTranslation{English}{gegeben}{Given}
115 \DeclareTranslation{English}{falls}{if}
116 \DeclareTranslation{English}{Falls}{If}
117 \DeclareTranslation{English}{Annahme}{Assumption}
118 \DeclareTranslation{English}{Angenommen-dass}{Suppose that}

```

7.5 Headers & Footers

Sets the page-headers.

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

The headers look like specified above (??). Also inserts the Titlepage.

```

119 \fancypagestyle{firstpage}{
120   %
121   \fancyhf{}
122   % clear all six fields
123   \renewcommand{\headrulewidth}{.7pt}
124   \renewcommand{\footrulewidth}{0pt}
125   \fancyfoot[R]{\thepage}
126   \fancyhead[L]{\hwa@tutorium}
127   \fancyhead[R]{\@date } }
128 \fancypagestyle{followingpage}{
129   %
130   \fancyhf{}
131
132   \ifthenelse{\equal{\hwa@design}{v2}}{
133     \ifhwa@twoside % IF
134
135     \fancyhead[RO]{\@author}
136     \fancyhead[LO]{\hwa@kurs\
137       \hwa@tutorium}
138     \fancyhead[LE]{
139       \ifthenelse{\equal{\hwa@sheetTitle}{}}{\hwa@sheetTitle\}
140       \GetTranslation{abgabe}: \hwa@abgabe
141     }
142     \fancyfoot[RO,LE]{\thepage}
143
144     \else %ELSE
145
146     \fancyhead[R]{\hwa@kurs\
147       \@author}
148     \fancyhead[L]{\hwa@tutorium\
149       \ifthenelse{\equal{\hwa@sheetTitle}{}}{\hwa@sheetTitle\}
150       \GetTranslation{abgabe}: \hwa@abgabe}
151     \fancyfoot[R]{\thepage}
152     \fi %ENDIF
153   }{
154     %=== LEGACY CODE; DO NOT CHANGE =====
155     \ifthenelse{\equal{\hwa@design}{v1}}{
156       \fancyhead[RE,LO]{\@author}
157       \fancyhead[LE,RO]{\hwa@kurs\
158         \GetTranslation{abgabe}: \hwa@abgabe}
159       \fancyfoot[RE,LO]{\thepage}
160     }{
161       \ClassError{HomeworkAssignment}{Value '\hwa@design' for key 'design'
162         is not known}{The option design takes an argument to set the
163         Pagestyle to the one of a previous version. Acceptable values are

```

```

164         'v1', or 'v2'}
165     }
166     %==== END OF LEGACY CODE =====
167 }
168
169
170 \renewcommand{\headrulewidth}{0.7pt}
171 \renewcommand{\footrulewidth}{0pt} } \pagestyle{followingpage}

```

8 Redefinition of existing Commands

D displays equation-numbers as upper-case roman numbers.

```
172 \renewcommand{\theequation}{\Roman{equation}}
```

8.1 Internal commands

8.1.1 Counter-Commands

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

```

173 \newcommand{\hwa@problemno}{\arabic{problem}}
174 \newcommand{\hwa@subproblemno}{\alph{subproblem}}
175 \newcommand{\hwa@subsubproblemno}{\roman{subsubproblem}}

```

8.1.2 Counter-Style Parser

Counter--Style Parser 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 8.1.3 for example usement.

```

176 \newcommand{\hwa@parseCounterStyle}[3]{
177   \ifthenelse{\equal{#1}{arabic}}{ \renewcommand{#2}{\arabic{#3}} }{
178     \ifthenelse{\equal{#1}{roman}}{ \renewcommand{#2}{\roman{#3}} }{
179       \ifthenelse{\equal{#1}{alph}}{ \renewcommand{#2}{\alph{#3}} }{
180         \ifthenelse{\equal{#1}{Alph}}{ \renewcommand{#2}{\Alph{#3}} }{
181           \ifthenelse{\equal{#1}{Roman}}{
182             \renewcommand{#2}{\Roman{#3}} }{
183             \ClassError{HomeworkAssignment}{Invalid Value #1 for
184               option Counter-Styling}{Possible Values are alph,
185               arabic, Arabic, roman or Roman.} } } } } }

```

8.1.3 Counter-Commands II

Counter--Style ParserCommands II Redefines the three counter-commands

```

186 \hwa@parseCounterStyle{\hwa@problemsty}{\hwa@problemno}{problem}
187 \hwa@parseCounterStyle{\hwa@subproblemsty}{\hwa@subproblemno}{subproblem}
188 \hwa@parseCounterStyle{\hwa@subsubproblemsty}{\hwa@subsubproblemno}{subsubproblem}

```

8.1.4 Grading-table

`\hwa@gradingtbl@...` Defines macros, which 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

```
189 \edef\hwa@gradingtbl@aux@defs{}
190 \newcommand{\hwa@gradingtbl@aux@lineOne}{}
191 \newcommand{\hwa@gradingtbl@aux@lineTwo}{}
192
193 \edef\hwa@gradingtbl@defs{}
194 \newcommand{\hwa@gradingtbl@lineOne}{}
195 \newcommand{\hwa@gradingtbl@lineTwo}{}

```

`\addToGradingTable`

```
196 \newcommand{\addToGradingTable}[1]{
197   \edef\hwa@gradingtbl@aux@defs{\hwa@gradingtbl@aux@defs|p{1cm}}
198   \edef\hwa@gradingtbl@aux@lineOne{\hwa@gradingtbl@aux@lineOne#1 &}
199   \edef\hwa@gradingtbl@aux@lineTwo{\hwa@gradingtbl@aux@lineTwo   &}
200 }

```

Write to aux

```
201 \AtEndDocument{%
202   \immediate\write\@auxout{%
203     \gdef\string\hwa@gradingtbl@defs{\hwa@gradingtbl@aux@defs}
204   }
205   \immediate\write\@auxout{%
206     \gdef\string\hwa@gradingtbl@lineOne{\hwa@gradingtbl@aux@lineOne}%
207   }
208   \immediate\write\@auxout{%
209     \gdef\string\hwa@gradingtbl@lineTwo{\hwa@gradingtbl@aux@lineTwo}%
210   }
211 }

```

`\makeGradingTable`

```
212 \newcommand{\makeGradingTable}{
213   \begin{table}[hb]
214     \centering
215     \large
216     \expandafter\table\expandafter{\hwa@gradingtbl@aux@defs |p{1cm}}\hline
217     \hwa@gradingtbl@lineOne   $\Sigma$      \\\hline
218     \hwa@gradingtbl@lineTwo   \vspace{.15cm}~\\\hline
219   \end{table}
220 \end{table}
221 }

```

8.2 Commands

`\subject` Defines `\kurs`. `\subject` equals `\kurs`

```
222 \newcommand{\hwa@kurs}{?\GetTranslation{subject}??}

```

```

223 \newcommand{\subject}[1]{\renewcommand{\hwa@kurs}{#1}}
224 \newcommand{\kurs}[1]{\subject{#1}}

\tutorial Defines \tutorial. \tutorium equals \tutorial
225 \newcommand{\hwa@tutorial}{?\GetTranslation{uebungsgruppe}??}
226 \newcommand{\tutorial}[1]{\renewcommand{\hwa@tutorial}{#1}}
227 \newcommand{\tutorium}[1]{\tutorial{#1}}

\sheetTitle Defines \sheetTitle.
228 \newcommand{\hwa@sheetTitle}{}
229 \newcommand{\sheetTitle}[1]{\def\hwa@sheetTitle{#1}}

\deadline Defines \deadline. \abgabe equals \deadline
230 \newcommand{\hwa@abgabe}{\today}
231 \newcommand{\deadline}[1]{\def\hwa@abgabe{#1}}
232 \newcommand{\abgabe}[1]{\deadline{#1}}

\maketitle Overrides maketitle.
233
234 \renewcommand{\maketitle} {
235   \thispagestyle{firstpage}
236   \setlength{\headheight}{25pt}
237   \twocolumn[{\%
238     \begin{centering}
239       \huge{\textbf{\hwa@kurs}} \vspace{.25cm} {\hrule height 2pt}
240       \vspace{.25cm} \large
241       \ifthenelse{\equal{\hwa@sheetTitle}{}}{\hwa@sheetTitle\\}
242       \GetTranslation{abgabe}: \hwa@abgabe\\
243       \vspace{.5cm} \hrule \vspace{.25cm}
244       \normalsize{\@author}\\
245       \vspace{.25cm} \hrule \vspace{.25cm} \normalsize
246     \end{centering}
247   }]
248 }

Defines and initialize all counters.
249 \newcounter{problem} \setcounter{problem}{0}
250 \newcounter{subproblem}[problem] \setcounter{subproblem}{0}
251 \newcounter{subsubproblem}[subproblem] \setcounter{subsubproblem}{0}
252

Defines ‘plain’ sectioning-commands. See 3.2 for more informations.
253 \newcommand{\problem}[1]{\@startsection{problem}%Name
254   {1}%Level
255   {\z@}%indent
256   {-2em \@plus -1em \@minus -1em}%beforeskip
257   {1ex \@plus .5ex}%afterskip
258   {\normalfont\Large\bfseries}%style
259   *{#1} \addcontentsline{toc}{section}{#1}

```

```

260 }
261
262 \newcommand{\subproblem}[1]{\@startsection{subproblem}%Name
263   {2}%Level
264   {\z@}%indent
265   {-1em \@plus -.5em \@minus -.5em}%beforeskip
266   {.5ex \@plus .5ex}%afterskip
267   {\normalfont\large\bfseries}%style
268   *{#1} \addcontentsline{toc}{subsection}{#1} }
269
270 \newcommand{\subsubproblem}[1]{\@startsection{subsubproblem}%Name
271   {3}%Level
272   {\z@}%indent
273   {-1em}%beforeskip
274   {.5em}%afterskip
275   {\normalfont\bfseries}%style
276   *{#1} }
277
278 \newcommand{\keyword}[1]{\@startsection{#1}%Name
279   {4}%Level
280   {\parindent}%indent
281   {-1em}%beforeskip
282   {\z@}%afterskip
283   {\normalfont\bfseries}%style
284   {#1:~~} }
285 }
286
287 \newcommand{\solution}[1][]{\keyword{\GetTranslation{loesung}}\ifthenelse{\equal{#1}{}}{}{\sim{#1}}}
288
289 \newcommand{\proof}[1][]{\keyword{\GetTranslation{beweis}}\ifthenelse{\equal{#1}{}}{}{\sim{#1}}}
290
291 \newcommand{\toShow}[1][]{\keyword{\GetTranslation{zuZeigen}}\ifthenelse{\equal{#1}{}}{}{\sim{#1}}}
292
293 \newcommand{\given}[1][]{\keyword{\GetTranslation{gegeben}}\ifthenelse{\equal{#1}{}}{}{\sim{#1}}}
294
295 \newcommand{\assumption}[1][]{\keyword{\GetTranslation{Annahme}}\ifthenelse{\equal{#1}{}}{}{\sim{#1}}}
296
297 \newcommand{\supposeThat}[1][]{\keyword{\GetTranslation{Angenommen-dass}}\ifthenelse{\equal{#1}{}}{}{\sim{#1}}}
298
299
300
301 \newcommand{\newproblem}[1][]{
302   \newproblem*{#1}
303   \addToGradingTable{# \hwa@problemno}
304 }
305
306 \WithSuffix\newcommand\newproblem*[1][]{\stepcounter{problem}
307   \ifthenelse{\equal{#1}{}}{}{\setcounter{problem}{#1}}}

```

Defines ‘better’ sectioning commands. See 3.2 and 3.2.2 for more informations.

```

308 \problem{\GetTranslation{aufgabe} \hwa@problemno}
309 }
310
311 \newcommand{\newsproblem}[1][\stepcounter{subproblem}
312 \ifthenelse{\equal{#1}{}}{ } {\setcounter{subproblem}{#1}}
313 \subproblem{\GetTranslation{aufgabe} \hwa@problemno}{.\hwa@subproblemno} }
314
315 \newcommand{\newsproblem}[1][\stepcounter{subsubproblem}
316 \ifthenelse{\equal{#1}{}}{ } {\setcounter{subsubproblem}{#1}}
317 \subsubproblem{\hwa@subsubproblemno}){ }
318

```

End of Proof

```

319 \newcommand{\QED}{\begin{flushright}
320 \textit{QED}
321 \end{flushright}
322 }
323 \newcommand{\EOP}{\begin{flushright}
324 $\square$
325 \end{flushright}
326 }
327 \newcommand{\eop}{\hfill$\blacksquare$}

```

c demonstrandum at iucundum est

```

328 \newcommand{\QED}{\begin{flushright}
329 $\triangle$
330 \end{flushright}
331 }
332 \newcommand{\qed}{\hfill$\triangle$}

```

Rounding brackets

Round brackets

```

333 \newcommand{\floor}[1]{\ensuremath{\left\lfloor #1 \right\rfloor}}
334 \newcommand{\ceil}[1]{\ensuremath{\left\lceil #1 \right\rceil}}
335 \newcommand{\roundHU}[1]{\ensuremath{\left\lceil #1 \right\rfloor}}
336 \newcommand{\roundHD}[1]{\ensuremath{\left\lfloor #1 \right\rceil}}

```

The following Macros are all stolen (and adapted) from occloxiun (see 3.3.3)

Math Common Set Symbols

```

337 \newcommand{\N}{\ensuremath{\mathbb{N}}}
338 \newcommand{\Z}{\ensuremath{\mathbb{Z}}}
339 \newcommand{\R}{\ensuremath{\mathbb{R}}}
340 \newcommand{\Q}{\ensuremath{\mathbb{Q}}}
341 \newcommand{\C}{\ensuremath{\mathbb{C}}}
342 \newcommand{\F}{\ensuremath{\mathbb{F}}}
343 % The last one is mine
344 \newcommand{\Primes}{\ensuremath{\mathbb{P}}}

```

Mathematical Functions

```

345 \newcommand{\GL}{\ensuremath{\text{GL}}}
346 \newcommand{\id}{\ensuremath{\text{id}}}
347 \newcommand{\diff}[1]{\ensuremath{\frac{d}{d\#1}}}
348 \newcommand{\dx}{\text{:dx}}
349
350 \newcommand{\divides}{\ensuremath{\mid}}
351 \newcommand{\property}{\ensuremath{\mid}}
352
353 \newcommand{\Var}{\ensuremath{\text{Var}}}
354 \newcommand{\Perm}{\ensuremath{\text{Perm}}}
355 \newcommand{\MComb}{\ensuremath{\text{MComb}}}
356 \newcommand{\Comb}{\ensuremath{\text{Comb}}}
357
358 \renewcommand{\dim}[1][\text{cm}]{\ensuremath{\text{dim}_{\#1}}}
359 \renewcommand{\Im}{\ensuremath{\text{Im}}}
360
361 \newcommand{\Pot}{\ensuremath{\text{Pot}}}
362 \newcommand{\Map}{\ensuremath{\text{Map}}}
363
364 \newcommand{\excup}{\ensuremath{\stackrel{\cdot}{\cup}}}
365
366 \newcommand{\falls}{\text{\GetTranslation{falls}}}
367
368 \newcommand{\Bin}{\ensuremath{\text{Bin}}}

```

Math Big Quantors

```

369 \let\forall\forall
370 \let\exists\exists
371 \renewcommand{\forall}{\hspace{2pt}\forall\hspace{2pt}}
372 \renewcommand{\exists}{\hspace{2pt}\exists\hspace{2pt}}
373 \newcommand{\bigforall}{\mbox{\raisebox{-2pt}{\Large $\forall$}}}
374 \newcommand{\bigexists}{\mbox{\raisebox{-2pt}{\Large $\exists$}}}

```

The End

```

375 \endinput

```