

The HomeworkAssignment class*

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

July 14, 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	4
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
5.1	Version-Scheme	7
6	Examples	8
7	Implementation	8
7.1	Packages & Options	8
7.2	TikZ-Styles	9
7.3	Geometry	9

*This document corresponds to `HomeworkAssignment` v2.4,dated 2017/06/27.

7.4	Translations	10
7.5	Headers & Footers	11
8	Redefinition of existing Commands	12
8.1	Internal commands	12
8.1.1	Counter-Commands	12
8.1.2	Counter-Style Parser	12
8.1.3	Counter-Commands II	12
8.1.4	Grading-table	13
8.2	Commands	13

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 verry best.

2 Options

<code>problemstyle=<1></code>	These options allow the customizatuion of the displayed numbers. For Example, if
<code>subproblemstyle=<1></code>	<code>problemstyle=Roman</code> , <code>subproblemstyle=arabic</code> , <code>subsubproblemstyle=roman</code>
<code>subsubproblemstyle=<1></code>	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 <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>design=<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>tikz</code>	Loads TikZ-Package and a couple of Styles, usefull for Papers in Computer-Science and ;athematics. See 7.2 for more informations
<code>fleqn</code>	Passes <code>fleqn</code> to <code>amsmath</code>

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
<code>\kurs</code>	

is “Kein Kurs”
`\kurs` is deprecated.

`\tutorial` 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.

`\deadline` Sets the deadline of the document. Takes it as an argument. Standard value 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 no number, nor will they increase a counter. Nevertheless, hey will be shown in the table of contents.
`\subproblem`
`\subsubproblem`

`\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 “Solution”, “Proof” “Given”, “To show”, “Assumption”, and “Suppose that”, respectively ¹, via `\keyword`. If an argument is passed, they print out this argument after the keyword. They are not mentioned in the table of contents.
`\proof`
`\given`
`\toShow`

`\assumption`
`\supposeThat`

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

3.2.2 ‘better’ Sectioning

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

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

3.3 Useful Macros

3.3.1 Quod Erat Demunstarndum, End of Proof

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

3.3.2 Quod Non Erat Demunstarndum at iucundum est

<code>\QNE</code>	Display a flushed-right \triangle . <code>\QNE</code> displays it in a new line, <code>\qne</code> at the end of
<code>\qne</code>	the same line.

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 very 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_n</code>	\mathbb{F}_n	Prime Field to base n
	<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-

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

mode

Command	Output
<code>\divides and property</code>	Prints a vertical Bar
<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>\Bin</code>	Bin
<code>\GL</code>	GL
<code>\id</code>	id
<code>\dx</code>	dx
<code>\excup</code>	$\dot{\cup}$
<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 `\QED`

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

v2.2.1 - 2017/06/20 Fix error with commands like `\solution` and `\keyword`.

v2.4 - Pending Fix math alignment,
Add option for flushed left equations,
Update amath port to use

5.1 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.

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_EX2e 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{xifthen}
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   \fancyhf{}
130
131   \ifthenelse{\equal{\hwa@design}{v2}}{
132     \ifhwa@twoside % IF
133
134     \fancyhead[RO]{\@author}
135     \fancyhead[LO]{\hwa@kurs\
136       \hwa@tutorium}
137     \fancyhead[LE]{
138       \ifthenelse{\equal{\hwa@sheetTitle}{}}{\hwa@sheetTitle\}
139       \GetTranslation{abgabe}: \hwa@abgabe
140     }
141     \fancyfoot[RO,LE]{\thepage}
142
143     \else %ELSE
144
145     \fancyhead[R]{\hwa@kurs\
146       \@author}
147     \fancyhead[L]{\hwa@tutorium\
148       \ifthenelse{\equal{\hwa@sheetTitle}{}}{\hwa@sheetTitle\}
149       \GetTranslation{abgabe}: \hwa@abgabe}
150     \fancyfoot[R]{\thepage}
151     \fi %ENDIF
152   }{
153     % ==== LEGACY CODE; DO NOT CHANGE =====
154     \ifthenelse{\equal{\hwa@design}{v1}}{
155       \fancyhead[RE,LO]{\@author}
156       \fancyhead[LE,RO]{\hwa@kurs\
157         \GetTranslation{abgabe}: \hwa@abgabe}
158       \fancyfoot[RE,LO]{\thepage}
159     }{
160       \ClassError{HomeworkAssignment}{Value '\hwa@design' for key 'design'
161         is not known}{The option design takes an argument to set the
162         Pagestyle to the one of a previous version. Acceptable values are
163         'v1', or 'v2'}

```

```

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

```

8 Redefinition of existing Commands

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

```
171 \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

```

172 \newcommand{\hwa@problemno}{\arabic{problem}}
173 \newcommand{\hwa@subproblemno}{\alph{subproblem}}
174 \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.

```

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

```

8.1.3 Counter-Commands II

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

```

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

```

8.1.4 Grading-table

`\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)

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

```

`\addToGradingTable`

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

```

Write to aux

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

```

`\makeGradingTable`

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

```

8.2 Commands

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

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

```

```

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

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

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

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

\maketitle Overrides maketitle.
232
233 \renewcommand{\maketitle} {
234   \thispagestyle{firstpage}
235   \setlength{\headheight}{25pt}
236   \twocolumn[{\%
237     \begin{centering}
238       \huge{\textbf{\hwa@kurs}} \vspace{.25cm} {\hrule height 2pt}
239       \vspace{.25cm} \large
240       \ifthenelse{\equal{\hwa@sheetTitle}{}}{\hwa@sheetTitle\\}
241       \GetTranslation{abgabe}: \hwa@abgabe\\
242       \vspace{.5cm} \hrule \vspace{.25cm}
243       \normalsize{\@author}\\
244       \vspace{.25cm} \hrule \vspace{.25cm} \normalsize
245     \end{centering}
246   }]
247 }

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

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

```

```

259 }
260
261 \newcommand{\subproblem}[1]{\@startsection{subproblem}%Name
262   {2}%Level
263   {\z@}%indent
264   {-1em \@plus -.5em \@minus -.5em}%beforeskip
265   {.5ex \@plus .5ex}%afterskip
266   {\normalfont\large\bfseries}%style
267   *{#1} \addcontentsline{toc}{subsection}{#1} }
268
269 \newcommand{\subsubproblem}[1]{\@startsection{subsubproblem}%Name
270   {3}%Level
271   {\z@}%indent
272   {-1em}%beforeskip
273   {.5em}%afterskip
274   {\normalfont\bfseries}%style
275   *{#1} }
276
277 \newcommand{\keyword}[1]{\@startsection{keyword}%Name
278   {4}%Level
279   {\parindent}%indent
280   {-1em}%beforeskip
281   {\z@}%afterskip
282   {\normalfont\bfseries}%style
283   *{#1~}}
284 }
285
286 \newcommand{\solution}[1] [] {\keyword{\GetTranslation{loesung}}\ifstrempy{#1}{~}{~}}
287
288 \newcommand{\proof}[1] [] {\keyword{\GetTranslation{beweis}}\ifstrempy{#1}{~}{~}}
289
290 \newcommand{\toShow}[1] [] {\keyword{\GetTranslation{zuZeigen}}\ifstrempy{#1}{~}{~}}
291
292 \newcommand{\given}[1] [] {\keyword{\GetTranslation{gegeben}}\ifstrempy{#1}{~}{~}}
293
294 \newcommand{\assumption}[1] [] {\keyword{\GetTranslation{Annahme}}\ifstrempy{#1}{~}{~}}
295
296 \newcommand{\supposeThat}[1] [] {\keyword{\GetTranslation{Angenommen-dass}}\ifstrempy{#1}{~}{~}}
297
298
299 \newcommand{\newproblem}[1] [] {
300   \newproblem*{#1}
301   \addtoGradingTable{\# \hwa@problemno}
302 }
303
304 \WithSuffix\newcommand\newproblem*[1] [] {\stepcounter{problem}
305   \ifthenelse{\equal{#1}{}} { } {\setcounter{problem}{#1}}
306   \problem{\GetTranslation{aufgabe} \hwa@problemno}

```

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

```

307 }
308
309 \newcommand{\newsubproblem}[1][\stepcounter{subproblem}
310 \ifthenelse{\equal{#1}{}}{ } {\setcounter{subproblem}{#1}}
311 \subproblem{\GetTranslation{aufgabe} \hwa@problemno{.}\hwa@subproblemno} }
312
313 \newcommand{\newsbsubproblem}[1][\stepcounter{subsubproblem}
314 \ifthenelse{\equal{#1}{}}{ } {\setcounter{subsubproblem}{#1}}
315 \subsubproblem{\hwa@subsubproblemno}} }
316

```

End of Proof

```

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

```

c demonstrandum at iucundum est

```

326 \newcommand{\QNE}{\begin{flushright}
327   $\triangle$
328 \end{flushright}
329 }
330 \newcommand{\qne}{\hfill$\triangle$}

```

Rounding brackets

Round brackets

```

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

```

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

Math Common Set Symbols

```

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

```


Mathematical Functions

```

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

```

Math Big Quantors

```

370 \let\forall\forall
371 \let\exists\exists
372 \renewcommand{\forall}{\hspace{2pt}\forall\hspace{2pt}}
373 \renewcommand{\exists}{\hspace{2pt}\exists\hspace{2pt}}
374 \newcommand{\bigforall}{\mbox{\raisebox{-2pt}{\Large $\mathsurround{4pt}\forall$}}}
375 \newcommand{\bigexists}{\mbox{\raisebox{-2pt}{\Large $\mathsurround{4pt}\exists$}}}

```

The End

```

376 \endinput

```