

The homeworkassignment*class[†]

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

April 9, 2019

Contents

1	Abstract	2
2	Dependencies	2
2.1	Mandatory Dependencies	2
2.2	Recommended Dependencies	3
3	Options	4
4	Layout	7
4.1	Headers & Footers	7
4.2	Enhance Mathenvironments	8
4.3	fonts	8
5	Commands	9
5.1	Constants	9
5.2	Document Informations	9
5.2.1	Inherited from <code>article</code>	9
5.3	Sectioning	9
5.3.1	‘plain’ Sectioning	10
5.3.2	‘better’ Sectioning	11
5.4	Useful Macros	13
5.4.1	QUOD ERAT DEMUNSTRANDUM, End of Proof	13
5.4.2	QUOD NON ERAT DEMUNSTARNDUM AT IUCUNDUM EST	13
5.4.3	Stolen Goods	13
5.4.4	Rounding	16
5.4.5	ToDo	16
5.5	Grading Table	16

*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` v3.1a, dated 2018/11/29.

5.5.1	Internal commands	17
5.6	Title	18
5.7	Counters	18
6	Environments	19
6.1	Proof	19
6.2	Proof by contradiction	19
7	Development and support	20
8	Changelog	20
8.1	Version-Scheme	21
9	Translations	23

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 upon `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

`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/amsfonts/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

2.2 Recommended Dependencies

These are not loaded automatically, but require a switch as option (see section 3). 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 homeworkassignment automatically loads a shipload of TikZ-librarys and own styles.
See section 3 for more informations

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}
```

array possibly can be removed

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

```

11 \RequirePackage{ifxetex}
12
13 \RequirePackage{wasysym}
14 \RequirePackage{adjustbox}
15
16 \RequirePackage{eso-pic}

```

3 Options

KV-Options is essential for this.

```

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

```

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

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

```

21 \DeclareStringOption[arabic]{problemsty}
22 \DeclareStringOption[alph]{subproblemsty}
23 \DeclareStringOption[roman]{subsubproblemsty}

```

`tikz` Loads TikZ-Package and a couple of Styles, usefull for Papers in Computer-Science
and mathematics. See 3 for more informations

```

24 \DeclareBoolOption[false]{tikz}

```

`listings` Loads Listings Package and sets listing-layout to use a small fontsize. Adds
indication for linebreaks.

```

25 \DeclareBoolOption[false]{listings}

```

`oneside, twoside` Changes layout. `oneside` is the complementary option to `twoside`
Standard layout is `twopaged`.

```

26 \DeclareBoolOption[true]{twoside}
27 \DeclareComplementaryOption{oneside}{twoside}

```

`onecolumn, twocolumn` Changes layout. `onecolumn` is the complementary option to `twocolumn`.
Standard Layout has two columns

```

28 \DeclareBoolOption[true]{twocolumn}
29 \DeclareComplementaryOption{onecolumn}{twocolumn}

```

`punchmark` Adds a mark for an hole puncher. Standard Layout has no marking.

```

30 \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.

```

31 \DeclareStringOption[all]{hlines}

```

`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 already existing command. For a reason, that currently (06th of December 2018) slips my mind completely, `unicode-math` needs to be loaded after `article`, because 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 5.4.1

```

32 \ifxetex
33 \DeclareBoolOption[true]{unicodemath}
34 \else
35 \DeclareBoolOption[false]{unicodemath}
36 \fi

Loads article and processes the options
37 \ProcessKeyvalOptions*
38 \ifhwa@twoside
39 \PassOptionsToClass{twoside}{article}
40 \else
41 \PassOptionsToClass{oneside}{article}
42 \fi
43 \ifhwa@twocolumn
44 \PassOptionsToClass{twocolumn}{article}
45 \else
46 \PassOptionsToClass{onecolumn}{article}
47 \fi
48 \LoadClass{article}
49

Loads listings, if wanted
50 \ifhwa@listings
51 \RequirePackage{listings}
52 \lstset{
53   frame = single,
54   breaklines = true,
55   postbreak=\raisebox{0ex}[0ex][0ex]{\ensuremath{\hookrightarrow\space}},
56   basicstyle=\scriptsize
57 }
58 \else
59 \empty
60 \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`

```

61
62 \newcommand{\hwa@hline@LONE}{\vspace{.25cm} {\hrule height 2pt}
63   \vspace{.25cm}}
64 \newcommand{\hwa@hline@LTWO}{\vspace{.5cm} \hrule \vspace{.25cm}}
65 \newcommand{\hwa@headrulewidth}{.7pt}
66 \ifthenelse{\equal{\hwa@hlines}{all}}{
67   \renewcommand{\hwa@hline@LONE}{\vspace{.25cm} {\hrule height 2pt}
68     \vspace{.25cm}}
69   \renewcommand{\hwa@headrulewidth}{.7pt}
70   \renewcommand{\hwa@hline@LTWO}{\vspace{.5cm} \hrule \vspace{.25cm}}
71 }{
72   \ifthenelse{\equal{\hwa@hlines}{decreased}}{
73     \renewcommand{\hwa@hline@LONE}{\vspace{.25cm} {\hrule height 2pt}
74       \vspace{.25cm}}
75     \renewcommand{\hwa@headrulewidth}{.7pt}
76   }{\ifthenelse{\equal{\hwa@hlines}{header}}{
77     \renewcommand{\hwa@headrulewidth}{.7pt}
78   }{\ifthenelse{\equal{\hwa@hlines}{none}}{
79     \renewcommand{\hwa@headrulewidth}{0pt}
80   }}
81   \ClassError{homeworkassignment}{Value '\hwa@lines' for key 'hlines'
82     is not known}{The option hlines takes an argument to set which
83     hlines are drawn. Possible values are 'all','decreased' , 'header', and
84     'none'. 'all' is standard.}
85   }
86 }
87 \renewcommand{\hwa@hline@LONE}{~\\vspace{.5cm}}
88 }
89 \renewcommand{\hwa@hline@LTWO}{\vspace{.75cm}}
90 }

```

If tikz is Wanted, load Usefull Styles

```

91 \ifhwa@tikz
92 \RequirePackage{tikz}
93 \usetikzlibrary{shapes,arrows,positioning,decorations,
94   automata,backgrounds,petri,bending,
95   shapes.multipart}
96 \tikzset{
97   treenode/.style = {shape=circle, rounded corners,
98     draw, align=center},
99   graynode/.style = {fill=gray},
100  normalnode/.style = {treenode, font=\Large, bottom color=white},

```

```

101 array/.style = {rectangle split,
102     rectangle split horizontal,
103     rectangle split,
104     draw}
105 }
106 \fi

Make sure that this is the last Package loaded
107 \RequirePackage{geometry}
108 \ifhwa@twocolumn
109 \geometry{top=2cm, bottom=2cm, left=2cm,
110     headsep=14pt,hmarginratio={1:1}}
111 \else
112 \geometry{top=2cm, bottom=2cm, width=35em,
113     headsep=14pt,hmarginratio={4:3}}
114 \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.

```

115 \fancypagestyle{firstpage}{
116     %
117     \fancyhf{}
118     % clear all six fields
119     \renewcommand{\headrulewidth}{\hwa@headrulewidth}
120     \renewcommand{\footrulewidth}{0pt}
121     \fancyfoot[R]{\thepage}
122     \fancyhead[L]{\hwa@tutorial}
123     \fancyhead[R]{\@date } }
124 \fancypagestyle{followingpage}{
125     \fancyhf{}
126     \ifhwa@twoside % IF

```

```

127 \fancyhead[R0]{\@author}
128 \fancyhead[L0]{\hwa@kurs\
129 \hwa@tutorium}
130 \fancyhead[LE]{
131 \ifthenelse{\equal{\hwa@sheetTitle}{}}{\hwa@sheetTitle\}
132 \GetTranslation{abgabe}: \hwa@abgabe
133 }
134 \fancyfoot[R0,LE]{\thepage}
135
136 \else %ELSE
137
138 \fancyhead[R]{\hwa@kurs\
139 \@author}
140 \fancyhead[L]{\hwa@tutorium\
141 \ifthenelse{\equal{\hwa@sheetTitle}{}}{\hwa@sheetTitle\}
142 \GetTranslation{abgabe}: \hwa@abgabe}
143 \fancyfoot[R]{\thepage}
144 \fi %ENDIF
145 \renewcommand{\headrulewidth}{\hwa@headrulewidth}
146 \renewcommand{\footrulewidth}{0pt}
147 }
148 \pagestyle{followingpage}

```

4.2 Enhance Mathenvironments

A couple of thigns, to make math-environments more beautifull and compact.

`\theequation` Displays equation-numbers as upper-case roman numbers.

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

`\allowdisplaybreaks` Allow pagebreaks in Mathmode

```
150 \allowdisplaybreaks
```

4.3 fonts

I fancy the Gillius-Font-Family, so that is the default Sans-Serif font, when using XeTeX, The template does default to Gillius ADF, which is available for free, licensed under the GNU License.

```

151
152 \ifthenelse{\boolean{xetex}}{
153 \RequirePackage{fontspec}
154 \setsansfont{Gillius ADF}
155 }{
156 \RequirePackage{gillius2}
157 }

```


5 Commands

5.1 Constants

Defines some constants

`\hwa@pointboxsize` Explains it self.
158 `\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.

```
159 \newcommand{\hwa@kurs}{?\GetTranslation{subject}??} % To store the value
160 \newcommand{\subject}[1]{\renewcommand{\hwa@kurs}{#1}}
161 \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.

```
162 \newcommand{\hwa@tutorium}{?\GetTranslation{uebungsgruppe}??} % To store the value
163 \newcommand{\tutorial}[1]{\renewcommand{\hwa@tutorium}{#1}}
164 \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

```
165 \newcommand{\hwa@abgabe}{\today} % To store the value
166 \newcommand{\deadline}[1]{\def\hwa@abgabe{#1}}
167 \newcommand{\abgabe}[1]{\deadline{#1}}
```

`\sheetTitle` Sets a descriptonal Title of the Sheet, will be written in the header of every page.
168 `\newcommand{\hwa@sheetTitle}{}
169 \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 ‘plain’ Sectioning

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

```

170 \DeclareDocumentCommand\problem{m o}{\@startsection{problem}%Name
171   {1}%Level
172   {\z@}%indent
173   {-2em \@plus -1em \@minus -1em}%beforeskip
174   {1ex \@plus .5ex}%afterskip
175   {\normalfont\Large\sffamily\bfseries}%style
176   *{#1
177     \IfNoValueF{#2}{
178       \hfill
179       \frame{\framebox[\hwa@pointboxsize]{
180         \hfill \normalfont{\large/\small{#2}}}}
181     }
182   }
183   \addcontentsline{toc}{section}{#1}
184 }
185
186 \DeclareDocumentCommand\subproblem{m o}{\@startsection{subproblem}%Name
187   {2}%Level
188   {\z@}%indent
189   {-1em \@plus -.5em \@minus -.5em}%beforeskip
190   {.5ex \@plus .5ex}%afterskip
191   {\normalfont\large\sffamily\bfseries}%style
192   *{#1
193     \IfNoValueF{#2}{
194       \hfill \framebox[\hwa@pointboxsize]{
195         \hfill \normalfont\large/\small{#2}}
196     }
197   }
198   \addcontentsline{toc}{subsection}{#1}
199 }
200
201 \DeclareDocumentCommand\subsubproblem{m o}{\@startsection{subsubproblem}%Name
202   {3}%Level
203   {\z@}%indent
204   {-.5em}%beforeskip
205   {.5em}%afterskip
206   {\normalfont\sffamily\bfseries}%style
207   *{#1
208     \IfNoValueF{#2}{
209       \hfill \framebox[\hwa@pointboxsize]{
210         \hfill \normalfont\large/\scriptsize{#2}}
211     }

```

```

212 }
213 }
214

```

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

```

215 \newcommand{\keyword}[1]{\@startsection{keyword}%Name
216 {4}%Level
217 {\parindent}%indent
218 {-.1em}%beforeskip
219 {\z@}%afterskip
220 {\normalfont \sffamily\bfseries}%style
221 *{#1~~}
222 }

```

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”, respectively², 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> 223 \newcommand{\solution}[1] [] {\keyword{\GetTranslation{loesung}\ifstrempy{#1}{~#1:}}} 224 \newcommand{\toShow}[1] [] {\keyword{\GetTranslation{zuZeigen}\ifstrempy{#1}{~#1:}}} 225 \newcommand{\toDisprove}[1] [] { 226 \keyword{\GetTranslation{zuWiderlegen}\ifstrempy{#1}{~#1:}}} 227 \newcommand{\given}[1] [] {\keyword{\GetTranslation{gegeben}\ifstrempy{#1}{~#1:}}} 228 \newcommand{\assumption}[1] [] {\keyword{\GetTranslation{Annahme}\ifstrempy{#1}{~#1:}}} 229 \newcommand{\supposeThat}[1] [] {\keyword{\GetTranslation{Angenommen-dass}\ifstrempy{#1}{~#1:}}} </pre>
---	---

5.3.2 ‘better’ Sectioning

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

<code>\newproblem</code> <code>\newproblem*</code> <code>\newsproblem</code> <code>\newsproblem*</code>	<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 subsection 5.5), `\newproblem*` does not do this.

²As of v1.6, Translations are added, depending on the choosen Language, there may be an other Text displayed.
See section 9 for all Translations

They use counters, of course:

```

230 \newcounter{problem} \setcounter{problem}{0}
231 \newcounter{subproblem}[problem] \setcounter{subproblem}{0}
232 \newcounter{subsubproblem}[subproblem] \setcounter{subsubproblem}{0}
233
234 \DeclareDocumentCommand\newproblem{0}{ g}{
235   \stepcounter{problem}% to reset the lower counters
236   \ifthenelse{\equal{#1}{}}{
237     % empty
238   }{
239     \setcounter{problem}{#1}
240   }
241
242   \IfNoValueTF{#2}{
243     \problem{\GetTranslation{aufgabe} \hwa@problemno}
244     \addToGradingTable{\# \hwa@problemno}
245   }{
246     \problem{\GetTranslation{aufgabe} \hwa@problemno}[#2]
247     \addToGradingTable{\# \hwa@problemno}{/#2}
248   }
249 }
250
251 \WithSuffix\newcommand\newproblem*[1][ ]{\stepcounter{problem}
252   \ifthenelse{\equal{#1}{}}{ } {
253     \stepcounter{problem}% to reset the lower counters
254     \setcounter{problem}{#1}
255     \problem{\GetTranslation{aufgabe} \hwa@problemno}
256 }
257
258 \DeclareDocumentCommand\newsubproblem{0}{ g}{
259   \stepcounter{subproblem}
260   \ifthenelse{\equal{#1}{}}{ } {
261     \setcounter{subproblem}{#1}
262   }
263   \IfNoValueTF{#2}{
264     \subproblem{\GetTranslation{aufgabe}
265       \hwa@problemno}{.\hwa@subproblemno}
266   }{
267     \subproblem{\GetTranslation{aufgabe}
268       \hwa@problemno}{.\hwa@subproblemno}[#2]
269   }
270 }
271
272 \DeclareDocumentCommand\newsbsubproblem{0}{ g}{
273   \stepcounter{subsubproblem}
274   \ifthenelse{\equal{#1}{}}{ } {\setcounter{subsubproblem}{#1}}
275   \IfNoValueTF{#2}{
276     \subsubproblem{\hwa@subsubproblemno}
277   }

```

```

278 {
279   \subsubproblem{\hwa@subsubproblemno})[#2]
280 }
281 }
282

```

5.4 Useful Macros

5.4.1 QUOD ERAT DEMUNSTRANDUM, 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 keep compatibility to several Math-packages, which define the later.

```

283 \newcommand{\hwa@QED}{\begin{flushright}
284   \textsc{Qed}
285 \end{flushright}}
286 }
287 \newcommand{\QED}{\hwa@QED}
288
289 \ifhwa@unicodemath
290 \RequirePackage{unicode-math}
291 \AtBeginDocument{\let\QEDSymbol\QED
292 \renewcommand{\QED}{\hwa@QED}
293 }
294 \fi
295
296 \newcommand{\EOP}{\begin{flushright}
297   \(\square\)
298 \end{flushright}}
299 }
300 \newcommand{\eop}{\hfill\(\blacksquare\)}

```

5.4.2 QUOD NON ERAT DEMUNSTARNDUM AT IUCUNDUM EST

\QNE Display a flushed-right *triangle*. **\QNE** 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 wich are not proofed mathematically, but are explained in a ay, whcih lets no doubt on their correctness.

```

301 \newcommand{\QNE}{\begin{flushright} \(\triangle\)
302 \end{flushright}}
303 }
304 \newcommand{\qned}{\hfill\(\triangle\)}

```

5.4.3 Stolen Goods

»Das ist alles nur geklaut«

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

$\backslash N$			
$\backslash Z$	Defines a set of mathematical sets, which are very usefull (see Table 1)		
$\backslash R$			
$\backslash Q$	Command	Output	Description
$\backslash C$	$\backslash N$	\mathbb{N}	Natural Numbers
$\backslash F$	$\backslash Z$	\mathbb{Z}	Whole Numbers
$\backslash Primes$	$\backslash Q$	\mathbb{Q}	Rational Numbers
	$\backslash R$	\mathbb{R}	Real Numbers
	$\backslash C$	\mathbb{C}	Complex Numbers
	$\backslash F_n$	\mathbb{F}_n	Prime Field to base n
	$\backslash Primes^5$	\mathbb{P}	Set of all Primes

Table 1: Field-Commands

```

305 \newcommand{\N}{\ensuremath{\mathbb{N}}}
306 \newcommand{\Z}{\ensuremath{\mathbb{Z}}}
307 \newcommand{\R}{\ensuremath{\mathbb{R}}}
308 \newcommand{\Q}{\ensuremath{\mathbb{Q}}}
309 \newcommand{\C}{\ensuremath{\mathbb{C}}}
310 \newcommand{\F}{\ensuremath{\mathbb{F}}}
311 % The last one is mine
312 \newcommand{\Primes}{\ensuremath{\mathbb{P}}}
```

$\backslash GL$		
$\backslash id$	Output usefull Plaintext-Operators and Functions. See table 2. Require	
$\backslash Var$	Mathmode	
$\backslash Perm$		
$\backslash MComb$		
$\backslash Comb$	Command	Output
$\backslash Pot$	$\backslash GL$	GL
$\backslash Map$	$\backslash id$	id
$\backslash Hom$	$\backslash Var$	Var
$\backslash Ker$	$\backslash Perm$	Perm
$\backslash Intpol$	$\backslash Comb$	Comb
$\backslash Pol$	$\backslash MComb$	MComb
$\backslash Sol$	$\backslash Pot$	Pot
$\backslash Bin$	$\backslash Map$	Map
$\backslash charakteristik$	$\backslash Hom$	Hom
$\backslash fo$	³ “Occloxiium” on GitHub: https://github.com/occloxiium	
$\backslash first$	⁴ amath.sty is part of Alexander Bartolomey's Alphabet Classes: https://github.com/occloxiium/AlphabetClasses	
$\backslash la$	⁵ Has to be $\backslash Primes$, because $\backslash P$ is already in use	
$\backslash diff$		
$\backslash partdiff$		
$\backslash dx$		
$\backslash divides$		
$\backslash property$		
$\backslash dim$		
$\backslash Im$		
$\backslash excup$		
$\backslash falls$		

<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

```

313 \DeclareMathOperator{\GL}{GL}
314 \DeclareMathOperator{\id}{id}
315 \DeclareMathOperator{\Var}{Var}
316 \DeclareMathOperator{\Perm}{Perm}
317 \DeclareMathOperator{\MComb}{MComb}
318 \DeclareMathOperator{\Comb}{Comb}
319 \DeclareMathOperator{\Pot}{Pot}
320 \DeclareMathOperator{\Map}{Map}
321 \DeclareMathOperator{\Hom}{Hom}
322 \DeclareMathOperator{\Ker}{Ker}
323 \DeclareMathOperator{\Intpol}{Intpol}
324 \DeclareMathOperator{\Pol}{Pol}
325 \DeclareMathOperator{\Sol}{Sol}
326 \DeclareMathOperator{\Bin}{Bin}
327 \DeclareMathOperator{\charakteristik}{char}
328 \DeclareMathOperator{\fo}{fo}
329 \DeclareMathOperator{\first}{fi}
330 \DeclareMathOperator{\la}{la}
331
332 \newcommand{\diff}[1]{\ensuremath{\frac{d}{d#1}}}
333 \newcommand{\partdiff}[1]{\ensuremath{\frac{\partial}{\partial#1}}}
334 \newcommand{\dx}{\mathop{:}dx}
335 \newcommand{\divides}{\ensuremath{\mid}}
336 \newcommand{\property}{\ensuremath{\mid}}
337
338 \renewcommand{\dim}[1][\text]{\ensuremath{\text{dim}_{#1}}}
339 \renewcommand{\Im}{\ensuremath{\text{Im}}}
340
341 \newcommand{\excup}{\ensuremath{\stackrel{\cdot}{\cup}}}
342 \newcommand{\falls}{\text{\GetTranslation{falls}}}

```

⁶In German, actual Translation may differ

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>	$\lfloor <1> \rfloor$	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

```

343 \newcommand{\floor}[1]{\ensuremath{\left\lfloor\right. #1 \right\rfloor}}
344 \newcommand{\ceil}[1]{\ensuremath{\left\lceil\right. #1 \right\rceil}}
345 \newcommand{\roundHU}[1]{\ensuremath{\left\lfloor\right. #1 \right\rfloor}}
346 \newcommand{\roundHD}[1]{\ensuremath{\left\lfloor\right. #1 \right\rfloor}}

```

`\bigforall`

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

```

347 \let\forall\forallforall
348 \let\exists\existsexists
349 \renewcommand{\forall}{\ensuremath{\hskip 2pt \forall\hskip 2pt}}
350 \renewcommand{\exists}{\ensuremath{\hskip 2pt \exists\hskip 2pt}}
351 \newcommand{\bigforall}{\mbox{\raisebox{-2pt}{\Large $\mathsurround4pt\forall$}}}
352 \newcommand{\bigexists}{\mbox{\raisebox{-2pt}{\Large $\mathsurround4pt\exists$}}}

```

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.

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.

```

353 \DeclareDocumentCommand\addToGradingTable{m g}{
354   \edef\hwa@gradingtbl@aux@defs{\hwa@gradingtbl@aux@defs|p{\hwa@pointboxsize}}
355   \edef\hwa@gradingtbl@aux@lineOne{\hwa@gradingtbl@aux@lineOne{#1} &}
356   \IfNoValueTF{#2}{
357     \edef\hwa@gradingtbl@aux@lineTwo{\hwa@gradingtbl@aux@lineTwo &}
358   }{
359     \edef\hwa@gradingtbl@aux@lineTwo{\hwa@gradingtbl@aux@lineTwo\vfill\hfill

```



```

360      {\string\small #2} &}
361   }
362 }

```

\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 L^AT_EX, until all are added.

It will never overflow the Line-Width, thanks to an **adjustbox**. [#1]: *Optional*.

The total number of points reachable.

```

363 \DeclareDocumentCommand\makeGradingTable{o}{
364   \begin{table}[hb]
365     \centering
366     \large
367     \begin{adjustbox}{max width=\linewidth}
368       \expandafter\table\expandafter{\hwa@gradingtbl@defs ||p{\hwa@pointboxsize}||}\hline
369       \hwa@gradingtbl@lineOne \(\Sigma\) \\\hline\small
370       \hwa@gradingtbl@lineTwo \IfNoValueTF{#1}{~}{\vfill\hfill/#1}\vspace{.15cm}\\ \hline
371     \end{table}
372   \end{adjustbox}
373 \end{table}
374 }

```

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)

```

375 \edef\hwa@gradingtbl@aux@defs{}
376 \newcommand{\hwa@gradingtbl@aux@lineOne}{}
377 \newcommand{\hwa@gradingtbl@aux@lineTwo}{}
378
379 \edef\hwa@gradingtbl@defs{}
380 \newcommand{\hwa@gradingtbl@lineOne}{}
381 \newcommand{\hwa@gradingtbl@lineTwo}{}

```

\write\@auxout Write to aux

```

382 \AtEndDocument{%
383   \immediate\write\@auxout{%
384     \gdef\string\hwa@gradingtbl@defs{\hwa@gradingtbl@aux@defs}
385   }
386   \immediate\write\@auxout{%
387     \gdef\string\hwa@gradingtbl@lineOne{\hwa@gradingtbl@aux@lineOne}%
388   }
389   \immediate\write\@auxout{%
390     \gdef\string\hwa@gradingtbl@lineTwo{\hwa@gradingtbl@aux@lineTwo}%

```

```

391 }
392 }

```

5.6 Title

`\maketitle` Overrides `\maketitle`.

```

393 \renewcommand{\maketitle} {
394   \thispagestyle{firstpage}
395   \ifhwa@twocolumn{
396     \twocolumn[{
397       \hwa@maketitletext
398     }]
399   }\else{
400     \hwa@maketitletext
401   }\fi
402 }

```

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

```

403 \newcommand{\hwa@maketitletext}{
404   \begin{centering}
405     \huge{\textsf{\textbf{\hwa@kurs}}}\hwa@hline@LONE \large
406     \ifthenelse{\equal{\hwa@sheetTitle}{}}{\textsf{\hwa@sheetTitle}}{\}
407     \GetTranslation{abgabe}: \hwa@abgabe\\
408     \hwa@hline@LTWO
409     \normalsize{\@author}\\
410     \hwa@hline@LTWO \normalsize
411   \end{centering}
412 }
413 \ifthenelse{\boolean{hwa@punchmark}}{
414   \newcommand{\hwa@punchmarkRad}{3mm}
415   \newcommand{\hwa@punchmarkDistanceX}{12mm}
416   \newcommand{\hwa@punchmarkDistanceY}{40mm}
417   \AtBeginDocument{
418     % Where will the punch be?
419     \AddToShipoutPictureBG*{\AtPageUpperLeft{
420       \put(\LenToUnit{\hwa@punchmarkDistanceX}-\hwa@punchmarkRad*2},\LenToUnit{-.5\paperheight-\hwa@punchmarkRad*2}){\tikz{\draw (0,0) -- (5mm,0);}}}
421       \put(\LenToUnit{\hwa@punchmarkDistanceX}-\hwa@punchmarkRad*2},\LenToUnit{-.5\paperheight+\hwa@punchmarkRad*2}){\tikz{\draw (0,0) -- (5mm,0);}}}
422     % Punch-Positioningmark
423     \AddToShipoutPictureBG*{\AtPageUpperLeft{
424       \put(\LenToUnit{5mm},\LenToUnit{-.5\paperheight}){\tikz{\draw (0,0) -- (5mm,0);}}}
425     }
426   }{
427 }

```

5.7 Counters

The actual counters are defined in subsection 5.3.2.

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

```
428 \newcommand{\hwa@problemno}{\arabic{problem}}
429 \newcommand{\hwa@subproblemno}{\alph{subproblem}}
430 \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.

```
431 \newcommand{\hwa@parseCounterStyle}[3]{
432   \ifthenelse{\equal{#1}{arabic}}{\renewcommand{#2}{\arabic{#3}}}
433   \ifthenelse{\equal{#1}{roman}}{\renewcommand{#2}{\roman{#3}}}
434   \ifthenelse{\equal{#1}{alph}}{\renewcommand{#2}{\alph{#3}}}
435   \ifthenelse{\equal{#1}{Alph}}{\renewcommand{#2}{\Alph{#3}}}
436   \ifthenelse{\equal{#1}{Roman}}{\renewcommand{#2}{\Roman{#3}}}
437   \renewcommand{#2}{\Roman{#3}}}
438   \ClassError{homeworkassignment}{Invalid Value #1 for
439     option Counter-Styling}{Possible Values are alph,
440     arabic, Arabic, roman or Roman.} } } } }
```

Redefines the three counter-commands:

```
441 \hwa@parseCounterStyle{\hwa@problemsty}{\hwa@problemno}{problem}
442 \hwa@parseCounterStyle{\hwa@subproblemsty}{\hwa@subproblemno}{subproblem}
443 \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.

```
444 \NewDocumentEnvironment{proof}{G}{\GetTranslation{beweis}} 0{\QED}
445 {
446   \keyword{#1:~~}
447 }
448 {
449   #2
450 }
```

6.2 Proof by contradiction

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

```
451 \NewDocumentEnvironment{contradiction}{}
452 {
453   \begin{proof}{\GetTranslation{beweis}~\GetTranslation{per}~\GetTranslation{Widerspruch}}[\hfi
454   }
455   {
456   \end{proof}
457 }
```

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 `\QED`

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 5.4.3)

 • 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 - 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 beeing reset
 - Merry Christmas!
- v3.2 - pending** • Make XeLaTeX-Compatible
- Fix `\newproblem` requiring a Problem-Number
 - Add `\toDisprove` macro similar to the `\toShow` macro

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.

```
458 \DeclareTranslationFallback{aufgabe}{Aufgabe}
459 \DeclareTranslationFallback{loesung}{L\"osung}
460 \DeclareTranslationFallback{beweis}{Beweis}
461 \DeclareTranslationFallback{uebungsgruppe}{\"Ubungsgruppe}
462 \DeclareTranslationFallback{abgabe}{Abgabe}
463 \DeclareTranslationFallback{zuZeigen}{Zu zeigen}
464 \DeclareTranslationFallback{zuWiderlegen}{Zu widerlegen}
465 \DeclareTranslationFallback{gegeben}{Gegeben}
466 \DeclareTranslationFallback{falls}{falls}
467 \DeclareTranslationFallback{Annahme}{Annahme}
468 \DeclareTranslationFallback{Angenommen-dass}{Angenommen, dass}
469 \DeclareTranslationFallback{per}{per}
470 \DeclareTranslationFallback{Widerspruch}{Widerspruch}
471
472 \DeclareTranslation{German}{aufgabe}{Aufgabe}
473 \DeclareTranslation{German}{loesung}{L\"osung}
474 \DeclareTranslation{German}{beweis}{Beweis}
475 \DeclareTranslation{German}{uebungsgruppe}{\"Ubungsgruppe}
476 \DeclareTranslation{German}{abgabe}{Abgabe}
477 \DeclareTranslation{German}{zuZeigen}{Zu zeigen}
478 \DeclareTranslation{German}{zuWiderlegen}{Zu widerlegen}
479 \DeclareTranslation{German}{gegeben}{Gegeben}
480 \DeclareTranslation{German}{falls}{falls}
481 \DeclareTranslation{German}{Falls}{Falls}
482 \DeclareTranslation{German}{Annahme}{Annahme}
483 \DeclareTranslation{German}{Angenommen-dass}{Angenommen, dass}
484 \DeclareTranslation{German}{per}{per}
485 \DeclareTranslation{German}{Widerspruch}{Widerspruch}
486
487 \DeclareTranslation{English}{aufgabe}{Problem}
488 \DeclareTranslation{English}{loesung}{Solution}
489 \DeclareTranslation{English}{beweis}{Proof}
490 \DeclareTranslation{English}{uebungsgruppe}{Tutorial}
491 \DeclareTranslation{English}{abgabe}{Deadline}
492 \DeclareTranslation{English}{zuZeigen}{To show}
493 \DeclareTranslation{English}{zuWiderlegen}{To disprove}
494 \DeclareTranslation{English}{gegeben}{Given}
495 \DeclareTranslation{English}{falls}{if}
496 \DeclareTranslation{English}{Falls}{If}
497 \DeclareTranslation{English}{Annahme}{Assumption}
498 \DeclareTranslation{English}{Angenommen-dass}{Suppose that}
499 \DeclareTranslation{English}{per}{by}
```

500 \DeclareTranslation{English}{Widerspruch}{contradiction}

End

The End

501 \endinput