The homeworkassignment*class†

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

November 18, 2017

Contents

---

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

†This document corresponds to homeworkassignment v2.5f, dated 2017/11/18.

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   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 LaTeX-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 LaTeX–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 LaTeXclass and package authors

array Frank Mittelbach, David Carlisle, The LaTeX–Team, `https://www.ctan.org/pkg/array`,
A new implementations for tables and arrays

> **array** possibly can be removed

xparse Frank Mittelbach, Chris Rowley, David Carlisle, The LaTeX3 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 LaTeX 2$_\varepsilon$'s `\newcommand` macro, with significantly improved functionality.

gillius Bob Tennent, `https://ctan.org/pkg/gillius`,
A Gillian Sans inspired font, used for all sans serifes 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

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

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{gillius2}
```

3   Options

KV-Options is essential for this.

```
12 \RequirePackage{kvoptions}
13 \SetupKeyvalOptions{ family=hwa,
14   prefix=hwa@ }
15 \DeclareDefaultOption{\PassOptionsToClass{\CurrentOptionKey}{article}}
```

`problemstyle=<1>`
`subproblemstyle=<1>`
`subsubproblemstyle=<1>`

These options allow the customizatuion of the displayed numbers. For Example, if problemstyle=Roman, subproblemstyle=arabic, subsubproblemstyle=roman 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.

```
16 \DeclareStringOption[arabic]{problemsty}
17 \DeclareStringOption[alph]{subproblemsty}
18 \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

```
19 \DeclareBoolOption[false]{tikz}
```

`listings`

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

```
20 \DeclareBoolOption[false]{listings}
```

`oneside, twoside`

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

```
21 \DeclareBoolOption[true]{twoside}
22 \DeclareComplementaryOption{oneside}{twoside}
```

`onecolumn,twocolumn`

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

```
23 \DeclareBoolOption[true]{twocolumn}
24 \DeclareComplementaryOption{onecolumn}{twocolumn}
```

`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.

```
25 \DeclareStringOption[all]{hlines}
```

Loads article and processes the options

```
26 \ProcessKeyvalOptions*
27 \ifhwa@twoside
```

4

```
28 \PassOptionsToClass{twoside}{article}
29 \else
30 \PassOptionsToClass{oneside}{article}
31 \fi
32 \ifhwa@twocolumn
33 \PassOptionsToClass{twocolumn}{article}
34 \else
35 \PassOptionsToClass{onecolumn}{article}
36 \fi
37 \ifhwa@listings
38 \RequirePackage{listings}
39 \lstset{
40   frame = single,
41   breaklines = true,
42   postbreak=\raisebox{0ex}[0ex][0ex]{\ensuremath{\hookrightarrow\space}},
43   basicstyle=\scriptsize
44 }
45 \else
46 \empty
47 \fi
48 \LoadClass{article}
```

\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

```
49
50 \newcommand{\hwa@hline@LONE}{\vspace{.25cm} {\hrule height 2pt}
51   \vspace{.25cm}}
52 \newcommand{\hwa@hline@LTWO}{\vspace{.5cm} \hrule \vspace{.25cm}}
53 \newcommand{\hwa@headrulewidth}{.7pt}
54 \ifthenelse{\equal{\hwa@hlines}{all}}{
55   \renewcommand{\hwa@hline@LONE}{\vspace{.25cm} {\hrule height 2pt}
56     \vspace{.25cm}}
57   \renewcommand{\hwa@headrulewidth}{.7pt}
58   \renewcommand{\hwa@hline@LTWO}{\vspace{.5cm} \hrule \vspace{.25cm}}
59 }{
60   \ifthenelse{\equal{\hwa@hlines}{decreased}}{
61     \renewcommand{\hwa@hline@LONE}{ \vspace{.25cm} {\hrule height 2pt}
62       \vspace{.25cm}}
63     \renewcommand{\hwa@headrulewidth}{.7pt}
64   }{\ifthenelse{\equal{\hwa@hlines}{header}}{
65       \renewcommand{\hwa@headrulewidth}{.7pt}
66     }{\ifthenelse{\equal{\hwa@hlines}{none}}{
67         \renewcommand{\hwa@headrulewidth}{0pt}
68       }{
69         \ClassError{homeworkassignment}{Value '\hwa@lines' for key 'hlines'
```

```
70            is not known}{The option hlines takes an argument to set which
71            hlines are drawn. Possible values are 'all','decreased' , 'header', and
72            'none'. 'all' is standard.}
73        }
74     }
75     \renewcommand{\hwa@hline@LONE}{~\\\vspace{.5cm}}
76  }
77  \renewcommand{\hwa@hline@LTWO}{\vspace{.75cm}}
78 }
```

If tikz is Wanted, load Usefull Styles

```
79 \ifhwa@tikz
80 \RequirePackage{tikz}
81 \usetikzlibrary{shapes,arrows,positioning,decorations,
82   automata,backgrounds,petri,bending,
83   shapes.multipart}
84 \tikzset{
85   treenode/.style = {shape=circle, rounded corners,
86     draw, align=center},
87   graynode/.style = {fill=gray},
88   normalnode/.style    = {treenode, font=\Large, bottom color=white},
89   array/.style = {rectangle split,
90     rectangle split horizontal,
91     rectangle split,
92     draw}
93 }
94 \fi
```

Make sure that this is the last Package loaded

```
95 \RequirePackage{geometry}
96 \ifhwa@twocolumn
97 \geometry{top=2cm, bottom=2cm, left=2cm,
98    headsep=14pt,hmarginratio={1:1}}
99 \else
100 \geometry{top=2cm, bottom=2cm, width=35em,
101   headsep=14pt,hmarginratio={4:3}}
102 \fi
```

4   Page-Layout

Initially, the homeworkassignment had a verry *special* appereance, which became much mor 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.

```
103 \fancypagestyle{firstpage}{
104   %
105   \fancyhf{}
106   % clear all six fields
107   \renewcommand{\headrulewidth}{\hwa@headrulewidth}
108   \renewcommand{\footrulewidth}{0pt}
109   \fancyfoot[R]{\thepage}
110   \fancyhead[L]{\hwa@tutorium}
111   \fancyhead[R]{\@date } }
112 \fancypagestyle{followingpage}{
113   \fancyhf{}
114   \ifhwa@twoside % IF
115   \fancyhead[RO]{\@author}
116   \fancyhead[LO]{\hwa@kurs\\
117     \hwa@tutorium}
118   \fancyhead[LE]{
119     \ifthenelse{\equal{\hwa@sheetTitle}{}}{}{\hwa@sheetTitle\\}
120     \GetTranslation{abgabe}: \hwa@abgabe
121   }
122   \fancyfoot[RO,LE]{\thepage}
123
124   \else %ELSE
125
126   \fancyhead[R]{\hwa@kurs\\
127     \@author}
128   \fancyhead[L]{\hwa@tutorium\\
129     \ifthenelse{\equal{\hwa@sheetTitle}{}}{}{\hwa@sheetTitle\\}
130     \GetTranslation{abgabe}: \hwa@abgabe}
131   \fancyfoot[R]{\thepage}
132   \fi %ENDIF
133   \renewcommand{\headrulewidth}{\hwa@headrulewidth}
134   \renewcommand{\footrulewidth}{0pt}
135 }
136 \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.

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

\allowdisplaybreaks	Allow pagebreaks in Mathmode

138 `\allowdisplaybreaks`

5 Commands

5.1 Constants

Defines some constants

\hwa@pointboxsize	Explains it self.

139 `\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.

140 `\newcommand{\hwa@kurs}{?\GetTranslation{subject}?} % To store the value`
141 `\newcommand{\subject}[1]{\renewcommand{\hwa@kurs}{#1}}`
142 `\newcommand{\kurs}[1]{\subject{#1}}`

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

\tutorium is deprecated.

143 `\newcommand{\hwa@tutorium}{?\GetTranslation{uebungsgruppe}?}} % To store the value`
144 `\newcommand{\tutorial}[1]{\renewcommand{\hwa@tutorium}{#1}}`
145 `\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

146 `\newcommand{\hwa@abgabe}{\today}} % To store the value`
147 `\newcommand{\deadline}[1]{\def\hwa@abgabe{#1}}`
148 `\newcommand{\abgabe}[1]{\deadline{#1}}`

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

149 `\newcommand{\hwa@sheetTitle}{}`
150 `\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`
`\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.

```
151 \DeclareDocumentCommand\problem{m o}{\@startsection{problem}%Name
152   {1}%Level
153   {\z@}%indent
154   {-2em \@plus -1em \@minus -1em}%beforeskip
155   {1ex \@plus .5ex}%afterskip
156   {\normalfont\Large \sffamily\bfseries}%style
157   *{#1
158     \IfNoValueF{#2}{
159       \hfill
160     \frame{\framebox[\hwa@pointboxsize]{
161         \hfill \normalfont{\large/\small{#2}}}}}
162   }
163   }
164   \addcontentsline{toc}{section}{#1}
165 }
166
167 \DeclareDocumentCommand\subproblem{m o}{\@startsection{subproblem}%Name
168   {2}%Level
169   {\z@}%indent
170   {-1em \@plus -.5em \@minus -.5em}%beforeskip
171   {.5ex \@plus .5ex}%afterskip
172   {\normalfont\large \sffamily\bfseries}%style
173   *{#1
174     \IfNoValueF{#2}{
175       \hfill \framebox[\hwa@pointboxsize]{
176         \hfill\normalfont\large/\small{#2}}
177   }
178   }
179   \addcontentsline{toc}{subsection}{#1}
180 }
181
```

```
182 \DeclareDocumentCommand\subsubproblem{m o}{\@startsection{subsubproblem}%Name
183   {3}%Level
184   {\z@}%indent
185   {-.5em}%beforeskip
186   {.5em}%afterskip
187   {\normalfont \sffamily\bfseries}%style
188   *{#1
189     \IfNoValueF{#2}{
190       \hfill \framebox[\hwa@pointboxsize]{
191         \hfill\normalfont\large/\scriptsize{#2}}
192     }
193   }
194 }
195
```

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

```
196 \newcommand{\keyword}[1]{\@startsection{keyword}%Name
197   {4}%Level
198   {\parindent}%indent
199   {-.1em}%beforeskip
200   {\z@}%afterskip
201   {\normalfont \sffamily\bfseries}%style
202   *{#1~~}
203 }
```

        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 "Solu-
\proof   tion", "Proof" "Given", "To show", "Assumption", and "Suppose that", respectively
\given   [1], via \keyword. If an argument is passed, they print out this argument after the
\toShow   keyword. They are not mentioned in the table of contents.
\assumption
\supposeThat

```
204 \newcommand{\solution}[1][]{\keyword{\GetTranslation{loesung}\ifstrempty{#1}{}{~#1}:}}
205 \newcommand{\proof}[1][]{\keyword{\GetTranslation{beweis}\ifstrempty{#1}{}{~#1}:}}
206 \newcommand{\toShow}[1][]{\keyword{\GetTranslation{zuZeigen}\ifstrempty{#1}{}{~#1}:}}
207 \newcommand{\given}[1][]{\keyword{\GetTranslation{gegeben}\ifstrempty{#1}{}{~#1}:}}
208 \newcommand{\assumption}[1][]{\keyword{\GetTranslation{Annahme}\ifstrempty{#1}{}{~#1}:}}
209 \newcommand{\supposeThat}[1][]{\keyword{\GetTranslation{Angenommen-dass}\ifstrempty{#1}{}{~#1}}
```

5.3.2 '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.
\newsubproblem
\newsubsubproblem   ─────────────────────────
                    [1]As of v1.6, Translations are added, depending on the choosen Language, there may be an
                    other Text displayed.
                    See section 9 for all Translations

Normally, \newproblem adds the new Created Problem to the grading–table (see 5.5), \newproblem* does not do this.

5.4 Useful Macros

5.4.1 QUOD ERAT DEMUNSTARNDUM, End of Proof

\QED
\EOP
\eop

Display a flushed-right QED, □, or ■, respectively. \qed is not implemented, to keep compatibility to several Math-packages, which define the later.

5.4.2 QUOD NON ERAT DEMUNSTARNDUM AT IUCUNDUM EST

\QNED
\qned

Display a flushed-right △. \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.

5.4.3 Stolen Goods

»Das ist alles nur geklaut«

~Tobias Künzel

These Commands are not mine, they are all stolen from Alexander Bartolomey's[2] amath-Class[3]

\N
\Z
\R
\Q
\C
\F
\Primes

Defines a set of mathematical sets, which are verry usefull (see Table 1)

Command	Output	Description
\N	$\mathbb{N}$	Natural Numbers
\Z	$\mathbb{Z}$	Whole Numbers
\Q	$\mathbb{Q}$	Rational Numbers
\R	$\mathbb{R}$	Real Numbers
\C	$\mathbb{C}$	Complex Numbers
\F_n	$\mathbb{F}_n$	Prime Field to base n
\Primes[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
\divides and property	Prints a vertical Bar

---

[2]"Occloxium" on GitHub:https://github.com/occloxium

[3]amath.sty is part of Alexander Bartolomey's Alphabet Classes: https://github.com/occloxium/AlphabetClasses

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

Command	Output
`\Var`	Var
`\Perm`	Perm
`\Comb`	Comb
`\MComb`	MComb
`\Pot`	Pot
`\Map`	Map
`\Bin`	Bin
`\GL`	GL
`\id`	id
`\dx`	dx
`\excup`	$\dot{\cup}$
`\diff{<1>}`	$\frac{d}{d<1>}$

Table 2: Text-like Functions

`\falls` prints out »falls«[5]

5.4.4 Rounding

Require Mathmode

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

Table 3: Rounding Functions

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 excercise to the Grading-Table. All Problems, created with `\newproblem` are added automatically.

`\makeGradingTable`   Prints out the Table containig all Defined exercises ($\neq$Problems). Like `\tableofcontent`, it needs a second run of LaTeX, until all are added.
See example documents fot output

---

[5] In German, actual Translation may differ

6 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.

7 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 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 - pending "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 the hlines

7.1   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 developtment 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 developtment, developtment-versions do not have numeric patch–numbers, but alphabetic identifiers, directly after the minor–version.

8   Examples

For examples, pleas see the Git-Repo at `https://github.com/ACHinrichs/LaTeX-templates`

9   Translations

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

```
210 \DeclareTranslationFallback{aufgabe}{Aufgabe}
211 \DeclareTranslationFallback{loesung}{L\"osung}
212 \DeclareTranslationFallback{beweis}{Beweis}
213 \DeclareTranslationFallback{uebungsgruppe}{\"Ubungsgruppe}
214 \DeclareTranslationFallback{abgabe}{Abgabe}
215 \DeclareTranslationFallback{zuZeigen}{Zu zeigen}
216 \DeclareTranslationFallback{gegeben}{Gegeben}
217 \DeclareTranslationFallback{falls}{falls}
218 \DeclareTranslationFallback{Annahme}{Annahme}
219 \DeclareTranslationFallback{Angenommen-dass}{Anngenommen, dass}
220
221 \DeclareTranslation{German}{aufgabe}{Aufgabe}
222 \DeclareTranslation{German}{loesung}{L\"osung}
223 \DeclareTranslation{German}{beweis}{Beweis}
224 \DeclareTranslation{German}{uebungsgruppe}{\"Ubungsgruppe}
225 \DeclareTranslation{German}{abgabe}{Abgabe}
226 \DeclareTranslation{German}{zuZeigen}{Zu zeigen}
227 \DeclareTranslation{German}{gegeben}{Gegeben}
228 \DeclareTranslation{German}{falls}{falls}
229 \DeclareTranslation{German}{Falls}{Falls}
230 \DeclareTranslation{German}{Annahme}{Annahme}
231 \DeclareTranslation{German}{Angenommen-dass}{Anngenommen, dass}
232
233 \DeclareTranslation{English}{aufgabe}{Problem}
234 \DeclareTranslation{English}{loesung}{Solution}
235 \DeclareTranslation{English}{beweis}{Proof}
236 \DeclareTranslation{English}{uebungsgruppe}{Tutorial}
237 \DeclareTranslation{English}{abgabe}{Deadline}
238 \DeclareTranslation{English}{zuZeigen}{To show}
239 \DeclareTranslation{English}{gegeben}{Given}
240 \DeclareTranslation{English}{falls}{if}
241 \DeclareTranslation{English}{Falls}{If}
242 \DeclareTranslation{English}{Annahme}{Assumption}
243 \DeclareTranslation{English}{Angenommen-dass}{Suppose that}
```

10 Implementation

The following part is verry boring, but I have not found a solution to create a
`.cls`–file without including the implemetation into the document. Loads LaTeX2e
and sets the Version Loads the `article`, which is the base-class.

10.1 Packages & Options

10.2 TikZ-Styles

10.3 Geometry

10.4 Internal commands

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

```
244 \newcommand{\hwa@maketitletext}{
245   \begin{centering}
246     \huge{\textsf{\textbf{\hwa@kurs}}}\hwa@hline@LONE \large
247     \ifthenelse{\equal{\hwa@sheetTitle}{}}{}{\textsf{\hwa@sheetTitle}\\}
248     \GetTranslation{abgabe}: \hwa@abgabe\\
249     \hwa@hline@LTWO
250     \normalsize{\@author}\\
251     \hwa@hline@LTWO \normalsize
252   \end{centering}
253 }
```

10.4.1 Counter–Commands

```
254 \newcounter{problem} \setcounter{problem}{0}
255 \newcounter{subproblem}[problem] \setcounter{subproblem}{0}
256 \newcounter{subsubproblem}[subproblem] \setcounter{subsubproblem}{0}
257
```

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

```
258 \newcommand{\hwa@problemno}{\arabic{problem}}
259 \newcommand{\hwa@subproblemno}{\alph{subproblem}}
260 \newcommand{\hwa@subsubproblemno}{\roman{subsubproblem}}
```

10.4.2 Counter–Style Parser

Counter-Style Parser This takes a style-input (#1), one of the three previous defined commands (#2)
and the coresponding counter (#3) to redefine #1, so that it corresponds to #2.
See 10.4.3 for example usement.

```
261 \newcommand{\hwa@parseCounterStyle}[3]{
262   \ifthenelse{\equal{#1}{arabic}}{ \renewcommand{#2}{\arabic{#3}} }{
263     \ifthenelse{\equal{#1}{roman}}{ \renewcommand{#2}{\roman{#3}} }{
264       \ifthenelse{\equal{#1}{alph}}{ \renewcommand{#2}{\alph{#3}} }{
265         \ifthenelse{\equal{#1}{Alph}}{ \renewcommand{#2}{\Alph{#3}} }{
```

```
266           \ifthenelse{\equal{#1}{Roman}}{
267             \renewcommand{#2}{\Roman{#3}} }{
268             \ClassError{homeworkassignment}{Invalid Value #1 for
269               option Counter-Styling}{Possible Values are alph,
270               arabic, Arabic, roman or Roman.}  } } } } } }
```

10.4.3   Counter–Commands II

Redefines the three counter-commands

```
271 \hwa@parseCounterStyle{\hwa@problemsty}{\hwa@problemno}{problem}
272 \hwa@parseCounterStyle{\hwa@subproblemsty}{\hwa@subproblemno}{subproblem}
273 \hwa@parseCounterStyle{\hwa@subsubproblemsty}{\hwa@subsubproblemno}{subsubproblem}
```

10.4.4   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)

```
274 \edef\hwa@gradingtbl@aux@defs{}
275 \newcommand{\hwa@gradingtbl@aux@lineOne}{}
276 \newcommand{\hwa@gradingtbl@aux@lineTwo}{}
277
278 \edef\hwa@gradingtbl@defs{}
279 \newcommand{\hwa@gradingtbl@lineOne}{}
280 \newcommand{\hwa@gradingtbl@lineTwo}{}
```

\addToGradingTable

```
281 \DeclareDocumentCommand\addToGradingTable{m g}{
282   \edef\hwa@gradingtbl@aux@defs{\hwa@gradingtbl@aux@defs|p{\hwa@pointboxsize}}
283   \edef\hwa@gradingtbl@aux@lineOne{\hwa@gradingtbl@aux@lineOne{#1} &}
284   \IfNoValueTF{#2}{
285     \edef\hwa@gradingtbl@aux@lineTwo{\hwa@gradingtbl@aux@lineTwo &}
286   }{
287     \edef\hwa@gradingtbl@aux@lineTwo{\hwa@gradingtbl@aux@lineTwo\vfill\hfill
288       {\string\small #2} &}
289   }
290 }
```

W   rite to aux

```
291 \AtEndDocument{%
292   \immediate\write\@auxout{%
293     \gdef\string\hwa@gradingtbl@defs{\hwa@gradingtbl@aux@defs}
294   }
295   \immediate\write\@auxout{%
296     \gdef\string\hwa@gradingtbl@lineOne{\hwa@gradingtbl@aux@lineOne}%
297   }
298   \immediate\write\@auxout{%
299     \gdef\string\hwa@gradingtbl@lineTwo{\hwa@gradingtbl@aux@lineTwo}%
300   }
```

```
301 }
```

\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

Arguments [#1]: *Optional.* The total number of points reachable.

```
302 \DeclareDocumentCommand\makeGradingTable{o}{
303   \begin{table}[hb]
304     \centering
305     \large
306     \expandafter\tabular\expandafter{\hwa@gradingtbl@defs ||p{\hwa@pointboxsize}|}\hline
307     \hwa@gradingtbl@lineOne  $\Sigma$         \\\hline\small
308     \hwa@gradingtbl@lineTwo  \IfNoValueTF{#1}{~}{\vfill\hfill/#1}\vspace{.15cm}\\\hline
309     \endtabular
310   \end{table}
311   }
```

10.5   Commands

\deadline Defines `\deadline`. `\abgabe` equals `\deadline`

\maketitle Overrides maketitle.

```
312
313 \renewcommand{\maketitle} {
314   \thispagestyle{firstpage}
315   \ifhwa@twocolumn{
316     \twocolumn[{
317       \hwa@maketitletext
318     }]
319   }\else{
320     \hwa@maketitletext
321   }\fi
322 }
```

Defines and initialize all counters.
Defines 'plain' sectioning-commands. See 5.3 for more informations.
Defines 'better' sectioning commands. See 5.3 and 5.3.2 for more informations.

```
323 \DeclareDocumentCommand\newproblem{O{} g}{
324   \IfNoValueTF{#2}{
325     \newproblem*[#1]
326     \addToGradingTable{\# \hwa@problemno}
327   }{
328     \IfNoValueF{#1}{
329       \setcounter{problem}{#1}
330     }
331     %\newproblem*[#1]
```

```
332     \problem{\GetTranslation{aufgabe} \hwa@problemno}[#2]
333     \addToGradingTable{\# \hwa@problemno}{/#2}
334   }
335 }
336
337 \WithSuffix\newcommand\newproblem*[1][]{\stepcounter{problem}
338   \ifthenelse{\equal{#1}{}} { } {\setcounter{problem}{#1}}
339   \problem{\GetTranslation{aufgabe} \hwa@problemno}
340 }
341
342 \DeclareDocumentCommand\newsubproblem{O{} g}{
343   \stepcounter{subproblem}
344   \ifthenelse{\equal{#1}{}} { } {\setcounter{subproblem}{#1}}
345   \IfNoValueTF{#2}{
346     \subproblem{\GetTranslation{aufgabe}
347        \hwa@problemno{}.\hwa@subproblemno}
348   }
349   {
350     \subproblem{\GetTranslation{aufgabe}
351        \hwa@problemno{}.\hwa@subproblemno}[#2]
352   }
353 }
354
355 \DeclareDocumentCommand\newsubsubproblem{O{} g}{
356   \stepcounter{subsubproblem}
357   \ifthenelse{\equal{#1}{}} { } {\setcounter{subsubproblem}{#1}}
358   \IfNoValueTF{#2}{
359     \subsubproblem{\hwa@subsubproblemno)}
360   }
361   {
362     \subsubproblem{\hwa@subsubproblemno)}[#2]
363   }
364 }
365
```

End of Proof

```
366 \newcommand{\QED}{\begin{flushright}
367     \textsc{Qed}
368   \end{flushright}
369 }
370 \newcommand{\EOP}{\begin{flushright}
371     $\square$
372   \end{flushright}
373 }
374 \newcommand{\eop}{\hfill$\blacksquare$}
```

t demonstrandum at iucundum est

```
375 \newcommand{\QNED}{\begin{flushright}
376     $\triangle$
377   \end{flushright}
```

```latex
378 }
379 \newcommand{\qned}{\hfill$\triangle$}
```

Rounding brakets

```latex
380 \newcommand{\floor}[1]{\ensuremath{\left\lfloor #1 \right\rfloor}}
381 \newcommand{\ceil}[1]{\ensuremath{\left\lceil #1 \right\rceil}}
382 \newcommand{\roundHU}[1]{\ensuremath{\left\lceil #1 \right\rfloor}}
383 \newcommand{\roundHD}[1]{\ensuremath{\left\lfloor #1 \right\rceil}}
```

The following Macros are all stolen (and adapted) from occloxium (see 5.4.3)

```latex
384 \newcommand{\N}{\ensuremath{\mathbb{N}}}
385 \newcommand{\Z}{\ensuremath{\mathbb{Z}}}
386 \newcommand{\R}{\ensuremath{\mathbb{R}}}
387 \newcommand{\Q}{\ensuremath{\mathbb{Q}}}
388 \newcommand{\C}{\ensuremath{\mathbb{C}}}
389 \newcommand{\F}{\ensuremath{\mathbb{F}}}
390 % The last one is mine
391 \newcommand{\Primes}{\ensuremath{\mathbb{P}}}
```

```latex
392 \DeclareMathOperator{\GL}{GL}
393 \DeclareMathOperator{\id}{id}
394 \DeclareMathOperator{\Var}{Var}
395 \DeclareMathOperator{\Perm}{Perm}
396 \DeclareMathOperator{\MComb}{MComb}
397 \DeclareMathOperator{\Comb}{Comb}
398 \DeclareMathOperator{\Pot}{Pot}
399 \DeclareMathOperator{\Map}{Map}
400 \DeclareMathOperator{\Hom}{Hom}
401 \DeclareMathOperator{\Ker}{Ker}
402 \DeclareMathOperator{\Intpol}{Intpol}
403 \DeclareMathOperator{\Pol}{Pol}
404 \DeclareMathOperator{\Sol}{Sol}
405 \DeclareMathOperator{\Bin}{Bin}
406 \DeclareMathOperator{\charakteristik}{char}
407 \newcommand{\diff}[1]{\ensuremath{\frac{d}{d#1}}}
408 \newcommand{\dx}{\:dx}
409
410 \newcommand{\divides}{\ensuremath{\ |\ }}
411 \newcommand{\property}{\ensuremath{\ |\ }}
412
413 \renewcommand{\dim}[1][]{\ensuremath{\text{dim}_{#1}\ }}
414 \renewcommand{\Im}{\ensuremath{\text{Im}\ }}
415
416 \newcommand{\excup}{\ensuremath{\stackrel{.}{\cup}}}
417
418 \newcommand{\falls}{\text{\ \GetTranslation{falls}}\ }
```

Math Big Quantors

419 \let\oforall\forall
420 \let\oexists\exists
421 \renewcommand{\forall}{\ensuremath{\hskip 2pt \oforall \hskip 2pt}}
422 \renewcommand{\exists}{\ensuremath{\hskip 2pt \oexists \hskip 2pt}}
423 \newcommand{\bigforall}{\mbox{\raisebox{-2pt}[\height][\depth]{\Large $\mathsurround4pt\forall$
424 \newcommand{\bigexists}{\mbox{\raisebox{-2pt}[\height][\depth]{\Large $\mathsurround4pt\exists$

The End

425 \endinput