The HomeworkAssignment class[*]

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

May 19, 2017

1 Abstract

This class provides a relative simple docuemnt–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

`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`.

`design=<1>` Allows the User to select an older page-style, for backwards compatibility. Recognized values are `v1` and `v2`. 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, *older styles are not going to be maintained!*

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

`\subject`
`\kurs`

Sets the subject of the document. Takes the subject as argument. Standard Value

[*] This document corresponds to HomeworkAssignment v2.0,dated 2017/05/13.

is “Kein Kurs”
`\kurs` is deprecated.

`\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.

`\deadline`
`\abgabe`

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

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`
`\subproblem`
`\subsubproblem`

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.

`\solution`
`\proof`
`\given`
`\toShow`

They work like Paragraph, but do not take an argument, instead they print out “Lösung”, “Beweis” “Gegeben”, and “Zu zeigen”, respectively[1]. They are not mentioned in the table of contents.

3.2.2 ‘better’ Sectioning

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

`\newproblem`
`\newsubproblem`
`\newsubsubproblem`

These commands require no argument, and automatically create a numbered title. The optional Argument is the new value for the coresponding counter.

3.3 Useful Macros

3.3.1 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.

[1] As of v1.6, Translations are added, depending on the choosen Language, there may be an other Text displayed.
See 8.2 for all Translations

3.3.2 Quod Non Erat Demunstarndum at iucundum est

`\QNED`
`\qned`

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

3.4 Stolen Goods

≫Das ist alles nur geklaut≪

~Tobias Künzel

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

These Commands are not mine, they are all stolen from Alexander Bartolomey's[2] `amath`-Class[3] [4]$\mathbb{P}$Set of all Primes[5] Defines a set of mathematical fields, which are verry usefull:

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`	$\mathbb{F}$	Prime-Fieled?
`\Primes`		

Table 1: Field-Commands

4 Pagestyle

4.1 Headers

[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 $\mathbb{P}$, because ¶ creates ¶

[5] not a Field

5 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.

6 Changelog

v1.0 - 2016/10/23 Intial

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 - 2017/05/09 Change Margins,
Add Option to select older Page-Style,
Change standardlayout to twocolumn and twoside

7 Examples

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

8 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.

8.1 Packages & Options

```
\RequirePackage{kvoptions}
\SetupKeyvalOptions{ family=hwa,
  prefix=hwa@ }
\DeclareStringOption[arabic]{problemsty}
\DeclareStringOption[alph]{subproblemsty}
\DeclareStringOption[roman]{subsubproblemsty}
\DeclareBoolOption[false]{listings}
\DeclareStringOption[v2]{design}
\DeclareBoolOption[true]{twoside}
\DeclareComplementaryOption{oneside}{twoside}
\DeclareBoolOption[true]{twocolumn}
\DeclareComplementaryOption{onecolumn}{twocolumn}
% Redefine the article-options
%    \begin{macrocode}
\DeclareDefaultOption{\PassOptionsToClass{\CurrentOptionKey}{article}}
```

Processes the Options and loades article

```
\ProcessKeyvalOptions*
\ifhwa@twoside
\PassOptionsToClass{twoside}{article}
\else
\PassOptionsToClass{oneside}{article}
\fi
\ifhwa@twocolumn
\PassOptionsToClass{twocolumn}{article}
\else
\PassOptionsToClass{onecolumn}{article}
\fi
\LoadClass{article}
```

Loads required Packages

```
\RequirePackage{suffix}
\RequirePackage{fancyhdr}
\RequirePackage{ifthen}
\RequirePackage{translations}
\RequirePackage{amssymb}
```

```
\ifhwa@listings
\RequirePackage{listings}
\lstset{
  frame = single,
  breaklines = true,
  postbreak=\raisebox{0ex}[0ex][0ex]{\ensuremath{\hookrightarrow\space}},
  basicstyle=\scriptsize
}
\else
\empty
\fi

% Make sure that this is the last Package loaded
\ifthenelse{\equal{\hwa@design}{v2}}{
  \RequirePackage{geometry}
  \ifhwa@twocolumn
  \geometry{top=2cm, bottom=2cm, left=2cm,
    headsep=14pt,hmarginratio={1:1}}
  \else
  \geometry{top=2cm, bottom=2cm, width=35em,
    headsep=14pt,hmarginratio={4:3}}
  \fi
}{
  \ifthenelse{\equal{\hwa@design}{v1}}{
    \empty
  }{
    \ClassError{HomeworkAssignment}{Value ’\hwa@design’ for key ’design’
    is not known}{The option design takes an argument to set the
    Pagestyle to the one of a previous version. Acceptable values are
    ’v1’, or ’v2’}
  }
}
```

8.2 Translations

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

```
\DeclareTranslationFallback{aufgabe}{Aufgabe}
\DeclareTranslationFallback{loesung}{L\"osung}
\DeclareTranslationFallback{beweis}{Beweis}
\DeclareTranslationFallback{uebungsgruppe}{\"Ubungsgruppe}
\DeclareTranslationFallback{abgabe}{Abgabe}
\DeclareTranslationFallback{zuZeigen}{Zu zeigen}
\DeclareTranslationFallback{gegeben}{Gegeben}
\DeclareTranslationFallback{falls}{falls}
\DeclareTranslationFallback{Falls}{Falls}

\DeclareTranslation{German}{aufgabe}{Aufgabe}
\DeclareTranslation{German}{loesung}{L\"osung}
\DeclareTranslation{German}{beweis}{Beweis}
\DeclareTranslation{German}{uebungsgruppe}{\"Ubungsgruppe}
```

```
\DeclareTranslation{German}{abgabe}{Abgabe}
\DeclareTranslation{German}{zuZeigen}{Zu zeigen}
\DeclareTranslation{German}{gegeben}{Gegeben}
\DeclareTranslation{German}{falls}{falls}
\DeclareTranslation{German}{Falls}{Falls}

\DeclareTranslation{English}{aufgabe}{Problem}
\DeclareTranslation{English}{loesung}{Solution}
\DeclareTranslation{English}{beweis}{Proof}
\DeclareTranslation{English}{uebungsgruppe}{Tutorial}
\DeclareTranslation{English}{abgabe}{Deadline}
\DeclareTranslation{English}{zuZeigen}{To show}
\DeclareTranslation{English}{gegeben}{Given}
\DeclareTranslation{English}{falls}{if}
\DeclareTranslation{English}{Falls}{If}
```

8.3 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 (4.1). Also inserts the Titlepage.

```

\fancypagestyle{firstpage}{
  %
  \fancyhf{}
  % clear all six fields
  \renewcommand{\headrulewidth}{.7pt}
  \renewcommand{\footrulewidth}{0pt}
  \fancyfoot[R]{\thepage}
  \fancyhead[L]{\hwa@tutorium}
  \fancyhead[R]{\@date } }
\fancypagestyle{followingpage}{
  %
  \fancyhf{}

  \ifthenelse{\equal{\hwa@design}{v2}}{
    \ifhwa@twoside % IF

    \fancyhead[RO]{\@author}
    \fancyhead[LO]{\hwa@kurs\\
      \hwa@tutorium}
    \fancyhead[LE]{\GetTranslation{abgabe}: \hwa@abgabe}
    \fancyfoot[RO,LE]{\thepage}

    \else %ELSE

    \fancyhead[R]{\hwa@kurs\\
      \@author}
    \fancyhead[L]{\hwa@tutorium\\
   \GetTranslation{abgabe}: \hwa@abgabe}
  \fancyfoot[R]{\thepage}
  \fi %ENDIF
 }{
  \ifthenelse{\equal{\hwa@design}{v1}}{
   \fancyhead[RE,LO]{\@author}
   \fancyhead[LE,RO]{\hwa@kurs\\
    \GetTranslation{abgabe}: \hwa@abgabe}
   \fancyfoot[RE,LO]{\thepage}
  }{
   \ClassError{HomeworkAssignment}{Value '\hwa@design' for key 'design'
    is not known}{The option design takes an argument to set the
    Pagestyle to the one of a previous version. Acceptable values are
    'v1', or 'v2'}
  }
 }


 \renewcommand{\headrulewidth}{0.7pt}
 \renewcommand{\footrulewidth}{0pt} } \pagestyle{followingpage}
\AtBeginDocument{ \thispagestyle{firstpage}
 \setlength{\headheight}{25pt} }
```

8.4 Internal commands

8.4.1 Counter–Commands

Counter--Commands

These are used to output the Exercise numbers in the desired style

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

8.4.2 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 8.4.3 for example usement.

```
\newcommand{\hwa@parseCounterStyle}[3]{
 \ifthenelse{\equal{#1}{arabic}}{ \renewcommand{#2}{\arabic{#3}} }{
  \ifthenelse{\equal{#1}{roman}}{ \renewcommand{#2}{\roman{#3}} }{
   \ifthenelse{\equal{#1}{alph}}{ \renewcommand{#2}{\alph{#3}} }{
    \ifthenelse{\equal{#1}{Alph}}{ \renewcommand{#2}{\Alph{#3}} }{
     \ifthenelse{\equal{#1}{Roman}}{
      \renewcommand{#2}{\Roman{#3}} }{
      \ClassError{HomeworkAssignment}{Invalid Value #1 for
       option Counter-Styling}{Possible Values are alph,
       arabic, Arabic, roman or Roman.}  } } } } } }
```

8.4.3 Counter–Commands II

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

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

8.5 Commands

\subject Defines \kurs. \subject equals \kurs

```
\newcommand{\hwa@kurs}{?\GetTranslation{subject}?}
\newcommand{\subject}[1]{\renewcommand{\hwa@kurs}{#1}}
\newcommand{\kurs}[1]{\subject{#1}}
```

\tutorial Defines \tutorial. \tutorium equals \tutorial

```
\newcommand{\hwa@tutorium}{?\GetTranslation{uebungsgruppe}?}
\newcommand{\tutorial}[1]{\renewcommand{\hwa@tutorium}{#1}}
\newcommand{\tutorium}[1]{\tutorial{#1}}
```

\deadline Defines \deadline. \abgabe equals \deadline

```
\newcommand{\hwa@abgabe}{\today}
\newcommand{\deadline}[1]{\def\hwa@abgabe{#1}}
\newcommand{\abgabe}[1]{\deadline{#1}}
```

\maketitle Overrides maketitle.

```

\renewcommand{\maketitle} {
  \twocolumn[{%
    \begin{centering}
      \huge{\textbf{\hwa@kurs}} \vspace{.25cm} {\hrule height 2pt}
      \vspace{.25cm} \large
      \GetTranslation{abgabe}: \hwa@abgabe\\
      \vspace{.5cm} \hrule \vspace{.25cm}
      \normalsize{\@author}\\
      \vspace{.25cm} \hrule \vspace{.25cm} \normalsize
    \end{centering}
  }]
}
```

Defines and initialize all counters.

```
\newcounter{problem} \setcounter{problem}{0}
\newcounter{subproblem}[problem] \setcounter{subproblem}{0}
\newcounter{subsubproblem}[subproblem] \setcounter{subsubproblem}{0}

```

Defines ‘plain’ sectioning-commands. See 3.2 for more informations.

```
\newcommand{\problem}[1]{\@startsection{problem}%Name
  {1}%Level
  {\z@}%indent
  {-2em \@plus -1em \@minus -1em}%beforeskip
  {1ex \@plus .5ex}%afterskip
  {\normalfont\Large\bfseries}%style
  *{#1} \addcontentsline{toc}{section}{#1} }

\newcommand{\subproblem}[1]{\@startsection{subproblem}%Name
  {2}%Level
  {\z@}%indent
  {-1em \@plus -.5em \@minus -.5em}%beforeskip
  {.5ex \@plus .5ex}%afterskip
  {\normalfont\large\bfseries}%style
  *{#1} \addcontentsline{toc}{subsection}{#1} }

\newcommand{\subsubproblem}[1]{\@startsection{subsubproblem}%Name
  {3}%Level
  {\z@}%indent
  {-.5em}%beforeskip
  {.5em}%afterskip
  {\normalfont\bfseries}%style
  *{#1} }

\newcommand{\solution}[1][]{\@startsection{solution}%Name
  {4}%Level
  {\parindent}%indent
  {-.1em}%beforeskip
  {\z@}%afterskip
  {\normalfont\bfseries}%style
  *{\GetTranslation{loesung}\ifthenelse{\equal{#1}{}} {} { #1}:~~ } }

\newcommand{\proof}[1][]{\@startsection{proof}%Name
  {4}%Level
  {\parindent}%indent
  {-.1em}%beforeskip
  {\z@}%afterskip
  {\normalfont\bfseries}%style
  *{\GetTranslation{beweis}\ifthenelse{\equal{#1} {} } {} { #1}:~~ } }

\newcommand{\toShow}[1][]{\@startsection{to show}%Name
  {4}%Level
  {\parindent}%indent
  {-.1em}%beforeskip
  {\z@}%afterskip
  {\normalfont\bfseries}%style
  *{\GetTranslation{zuZeigen}\ifthenelse{\equal{#1} {} } {} { #1}:~~ } }

\newcommand{\given}[1][]{\@startsection{given}%Name
  {4}%Level
  {\parindent}%indent
  {-.1em}%beforeskip
  {\z@}%afterskip
  {\normalfont\bfseries}%style
  *{\GetTranslation{gegeben}\ifthenelse{\equal{#1} {} } {} { #1}:~~ } }

```

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

```
\newcommand{\newproblem}[1][]{\stepcounter{problem}
  \ifthenelse{\equal{#1}{}} { } {\setcounter{problem}{#1}}
  \problem{\GetTranslation{aufgabe} \hwa@problemno} }

\newcommand{\newsubproblem}[1][]{\stepcounter{subproblem}
  \ifthenelse{\equal{#1}{}} { } {\setcounter{subproblem}{#1}}
  \subproblem{\GetTranslation{aufgabe} \hwa@problemno{}.\hwa@subproblemno} }

\newcommand{\newsubsubproblem}[1][]{\stepcounter{subsubproblem}
  \ifthenelse{\equal{#1}{}} { } {\setcounter{subsubproblem}{#1}}
  \subsubproblem{\hwa@subsubproblemno)} }

```

`End of Proof`

```
\newcommand{\QED}{\begin{flushright}
    \textit{QED}
  \end{flushright}
}
\newcommand{\EOP}{\begin{flushright}
    $\square$
  \end{flushright}
}
\newcommand{\eop}{\hfill$\blacksquare$}
```

`t demonstrandum at iucundum est`

```
\newcommand{\QNED}{\begin{flushright}
    $\triangle$
  \end{flushright}
}
\newcommand{\qned}{\hfill$\triangle$}
```

The following Macros are all stolen (and adapted) from occloxium (see 3.4)

`Math Common Set Symbols`

```
\newcommand{\N}{\ensuremath{\mathbb{N}}}
\newcommand{\Z}{\ensuremath{\mathbb{Z}}}
\newcommand{\R}{\ensuremath{\mathbb{R}}}
\newcommand{\Q}{\ensuremath{\mathbb{Q}}}
\newcommand{\C}{\ensuremath{\mathbb{C}}}
\newcommand{\F}{\ensuremath{\mathbb{F}}}
% The last one is mine
\newcommand{\Primes}{\ensuremath{\mathbb{P}}}
```

`Mathematical Functions`

```
\newcommand{\GL}{\ensuremath{\text{GL}}}
```

```
\newcommand{\id}{\ensuremath{\text{id}}}
\newcommand{\diff}[1]{\ensuremath{\frac{d}{d#1}}}
\newcommand{\dx}{\:dx}
```

Mathematical Functions

```
\newcommand{\divides}{\ensuremath{\ |\ }}
\newcommand{\property}{\ensuremath{\ |\ }}

\newcommand{\Var}{\ensuremath{\text{Var}}}
\newcommand{\Perm}{\ensuremath{\text{Perm}}}
\newcommand{\MComb}{\ensuremath{\text{MComb}}}
\newcommand{\Comb}{\ensuremath{\text{Comb}}}

\renewcommand{\dim}[1][]{\ensuremath{\text{dim}_{#1}\ }}
\renewcommand{\Im}{\ensuremath{\text{Im}\ }}
\newcommand{\modulo}[2]{#1/\!\!#2\:}

\newcommand{\Pot}{\ensuremath{\text{Pot}}}
\newcommand{\Map}{\ensuremath{\text{Map}}}

\newcommand{\excup}{\ensuremath{\stackrel{.}{\cup}}}

\newcommand{\falls}{\text{\ \GetTranslation{falls}}\ }

\newcommand{\Bin}{\ensuremath{\text{Bin}\ }}
\renewcommand{\char}{\ensuremath{\text{char}\ }}
```

Math Big Quantors

```
\let\oforall\forall
\let\oexists\exists
\renewcommand{\forall}{\ensuremath{\hskip 2pt \oforall \hskip 2pt}}
\renewcommand{\exists}{\ensuremath{\hskip 2pt \oexists \hskip 2pt}}
\newcommand{\bigforall}{\mbox{\raisebox{-2pt}[\height][\depth]{\Large $\mathsurround4pt\forall$
\newcommand{\bigexists}{\mbox{\raisebox{-2pt}[\height][\depth]{\Large $\mathsurround4pt\exists$
```

The End

```
\endinput
```