

Praktische Informatik 1

Dokumente erstellen mit LaTeX

Thomas Röfer

Cyber-Physical Systems
Deutsches Forschungszentrum für
Künstliche Intelligenz

Multisensorische Interaktive Systeme
Fachbereich 3, Universität Bremen



TeX und LaTeX

- **TeX** ist ein Compiler, der einen Quelltext in ein typographisch professionell gesetztes Dokument übersetzt
- Er folgt dabei Regeln, die Verstöße gegen Layout-Regeln bewerten und sucht jeweils den besten Kompromiss
- **TeX** wurde 1979 von Donald E. Knuth entwickelt, um damit seine Bücher „The Art of Computer Programming“ zu schreiben
- **LaTeX** ist ein Makro-Paket von Leslie Lamport aus dem Jahr 1986, das die Erstellung strukturierter Dokumente auf Basis von **TeX** unterstützt

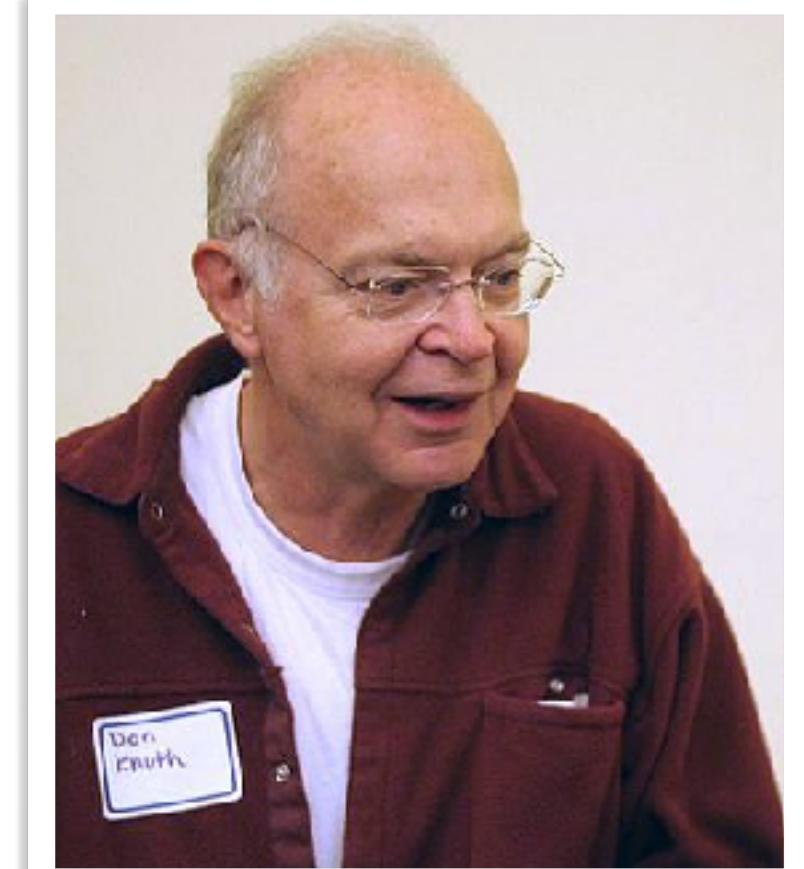
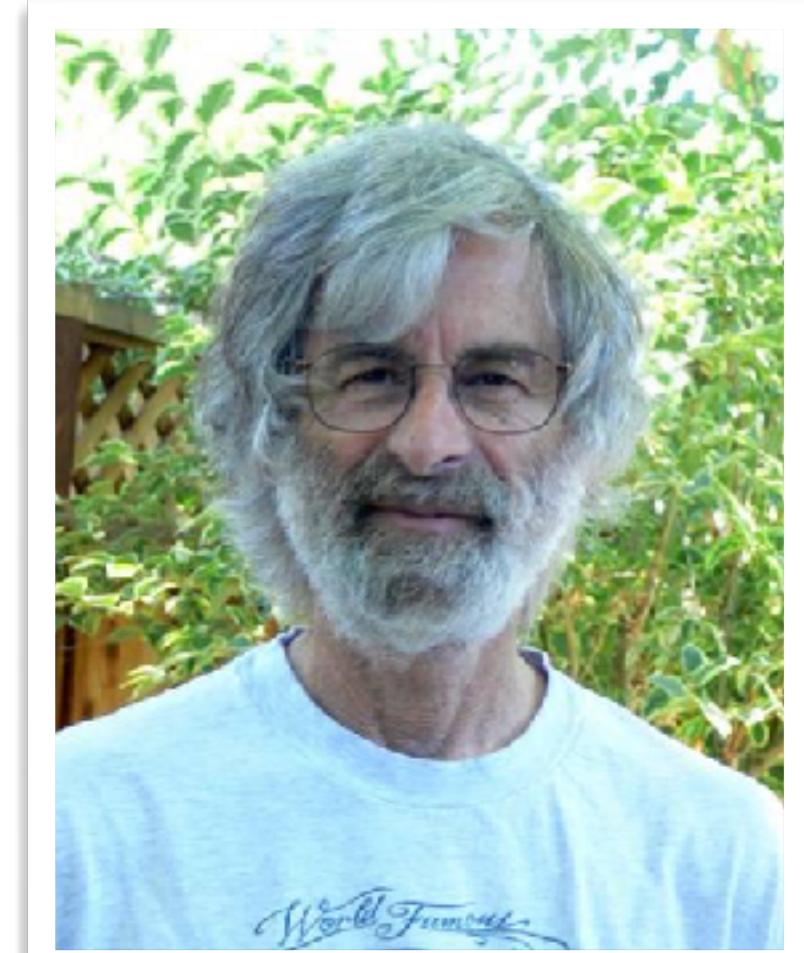


Foto aufgenommen von Jacob Appelbaum



Wissenschaftliches Papier: Demos

The screenshot shows the TeXstudio interface with the following details:

- Title Bar:** /Users/roefer/Documents/Lehre/Vorlesungen/PI-1/WiSe 21-22/pi1-2021.nosync/Beispiele/02/kicking.tex - TeXstudio
- Toolbar:** Standard file operations (New, Open, Save, Print, etc.) and document navigation.
- Left Margin:** Shows the current line number (1) and page number (1 von 12).
- Document Area:** The code editor window contains the LaTeX source code for a paper. The code includes:
 - \documentclass{UnCS}
 - \bibliographystyle{splncs03}
 - \usepackage[utf8]{inputenc}
 - \usepackage{graphicx}
 - \usepackage{url}
 - \usepackage{amsmath}
 - \usepackage{xcolor}
 - \usepackage{overpic}
 - \newcommand{\degree}{\text{\textdegree}}
 - \title{Online Generated Kick Motions for the NAO Balanced Using Inverse Dynamics}
 - \author{Felix Wenk\inst{1} \and Thomas Röfer\inst{2}}
 - \institute{Universität Bremen, Fachbereich 3 -- Mathematik und Informatik,\\ Postfach 330 440, 28334 Bremen, Germany \\ E-Mail: \texttt{fwenk@informatik.uni-bremen.de} \and Deutsches Forschungszentrum für Künstliche Intelligenz,\\ Cyber-Physical Systems, Enrique-Schmidt-Str. 5, 28359 Bremen, Germany \\ E-Mail: \texttt{thomas.roefer@dfki.de}}
 - \begin{document}
 - \maketitle
 - \begin{abstract}
- Preview Area:** The right side shows the generated PDF output. The title page reads:

Online Generated Kick Motions for the NAO Balanced Using Inverse Dynamics

Felix Wenk¹ and Thomas Röfer²

¹ Universität Bremen, Fachbereich 3 – Mathematik und Informatik,
Postfach 330 440, 28334 Bremen, Germany
E-Mail: fwenk@informatik.uni-bremen.de

² Deutsches Forschungszentrum für Künstliche Intelligenz,
Cyber-Physical Systems, Enrique-Schmidt-Str. 5, 28359 Bremen, Germany
E-Mail: thomas.roefer@dfki.de

Abstract. One of the major tasks of playing soccer is kicking the ball. Executing such complex motions is often solved by interpolating key-frames of the entire motion or by using predefined trajectories of the limbs of the soccer robot. In this paper we present a method to generate the trajectory of the kick foot online and to move the rest of the robot's body such that it is dynamically balanced. To estimate the balance of the robot, its Zero-Moment Point (ZMP) is calculated from its movement using the solution of the Inverse Dynamics. To move the ZMP, we use either a Linear Quadratic Regulator on the local linearization of the ZMP or the Cart-Table Preview Controller and compare their performances.
- Status Bar:** Zeile: 1 Spalte: 0 Modus: Einfügen, LT, <none>, UTF-8, Fertig, Automatisch, Page Number (1), 100% zoom.

Reservierte Zeichen

- Bestimmte Zeichen braucht TeX, um die eigenen Strukturen zu erkennen. Als Text können sie meistens mit vorangestelltem \ eingegeben werden (z.B. \%)
- \: Leitet Makros ein. Ersatz: **\textbackslash** oder **\backslash** (Mathematikmodus)
- %: Leitet Kommentare ein
- { ... }: Gruppiert Text, z.B. für Makroparameter
- \$: Schaltet zwischen Normal- und Mathematikmodus um
- &: Trennt Zellen in tabellenartigen Strukturen
- ~: Untrennbares Leerzeichen. Ersatz: **\textasciitilde** oder **\sim** (Mathematikmodus)
- Manche andere Zeichen können auch nicht direkt verwendet, aber durch ein Makro erzeugt werden

Befehle

- Beginnen mit `\`, dahinter kommt ihr Name
- Können zwingende und optionale Parameter haben
 - Zwingende Parameter sind entweder nur ein Zeichen, ein LaTeX-Befehl oder ein Block `{ ... }`
 - Optionale Parameter werden in eckige Klammern `[...]` eingefasst
 - Legale Werte für die Parameter sind pro Befehl individuell verschieden
 - Oft ist es auch einfach Freitext

```
\item
\item[a]
\frac{4}{8}
\frac{15}{16}
\newcommand\xyz{XYZ}
\begin{figure}[th]
\includegraphics[width=12cm]{bild.pdf}
```

Präambel

- Die **Präambel** steht vor dem eigentlichen Text, der zwischen **\begin{document}** und **\end{document}** steht
- **\documentclass{ ... }** muss ganz am Anfang stehen und legt den Typ des Dokuments fest und damit seine Formatierung (z.B. **article**, **report**, **book**, bei uns **pi1**)
- **\usepackage** bindet Pakete ein, die bestehende Befehle ändern oder neue hinzufügen (kann mehrfach genutzt werden)
- Des Weiteren können Makros (um)definiert werden, Parameter der Vorlage gesetzt oder verändert werden uvm.

Nützliche Pakete

- **\usepackage[utf8]{inputenc}**: Mit UTF-8 kodierter Text wird verstanden (Umlaute, **ß** usw.)
- **\usepackage[ngerman]{babel}**: Deutsche Texte, deutsche Silbentrennung und typographische Anführungszeichen mit „“ und „“
 - Erzeugen von Umlauten mit "a, "o, "u, "A, "O, "U und **ß** mit "s
- **\usepackage{graphicx}**: Bilder einbinden
- Diese drei werden von der Dokumentenklasse **pit** bereits eingebunden

Inhaltsstruktur

- Der Text wird in Abschnitte aufgeteilt, indem Überschriften definiert werden
 - **\chapter{Kapitelüberschrift}**: Beginnt neues Kapitel auf neuer Seite (**report**, **book**)
 - **\section{Überschrift}**, **\subsection{...}**, **\subsubsection{...}**: Abschnitte unterschiedlicher Hierarchie-Ebenen. **\section*{...}** usw. unterdrückt Nummerierung. **pi1** kennt noch **\skipsection**, um eine Aufgabe zu überspringen
 - **\paragraph{Spitzmarke}**, **\subparagraph{...}**
- Daraus kann mit **\tableofcontents** ein Inhaltsverzeichnis generiert werden
- Absätze werden durch eine **leere Zeile** beendet

Schriftauszeichnungen

- **\emph{...}**: Text so hervorheben, wie die Ersteller:in der Dokumentenklasse dies vorgesehen hat (meist *kursiv*)
- **\textbf{...}**: Fett (**boldface**)
- **\textit{...}**: Kursiv (*italics*)
- **\texttt{...}**: Feste Zeichenbreite (**typewriter**)
- Können gemischt werden, wenn der Schriftschnitt verfügbar ist
(\textbf{\textit{...}})

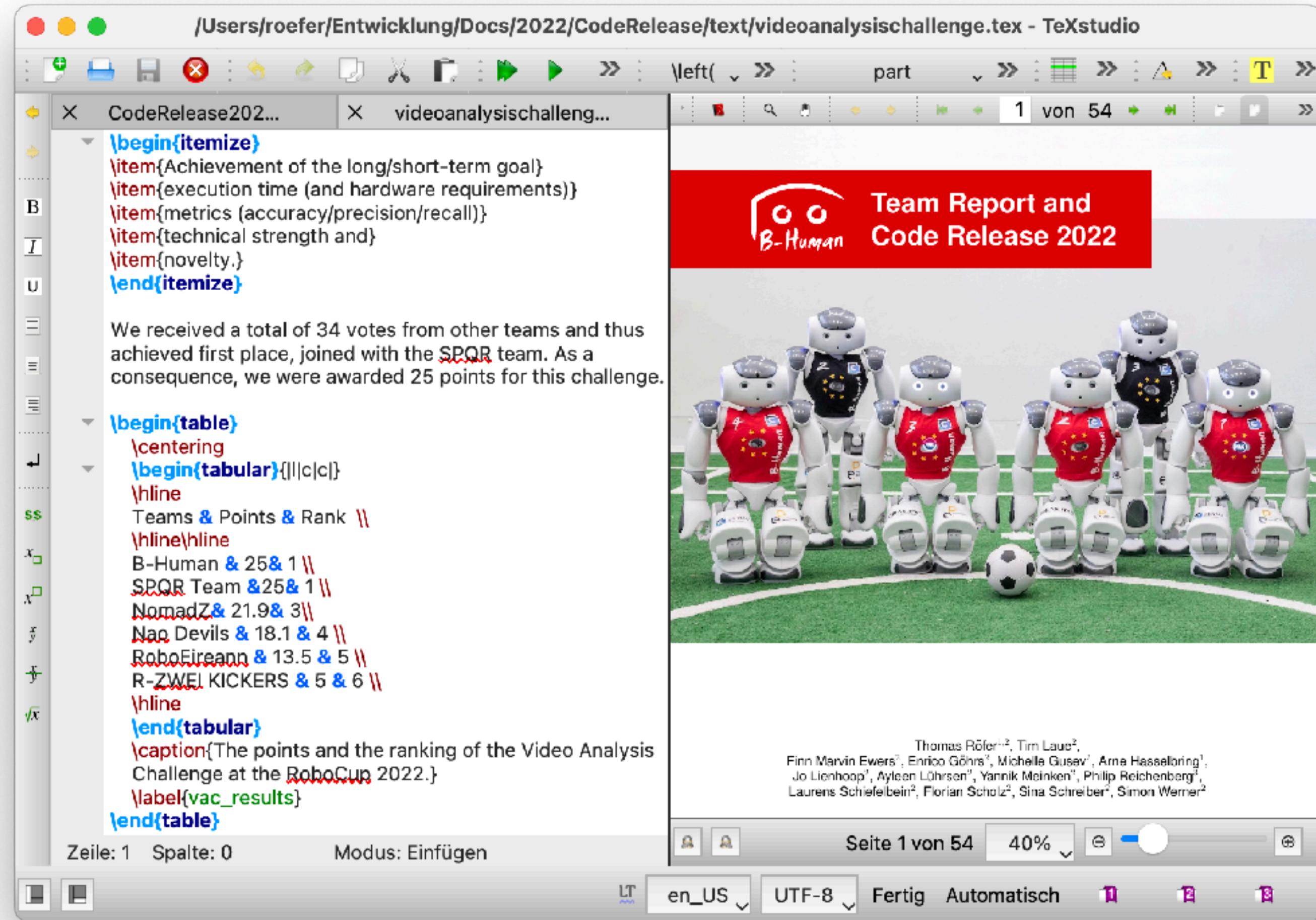
Titelseite bzw. -bereich

- Direkt auf **\begin{document}** folgt oft **\maketitle**
- Erzeugt bei **article** einen Titelbereich und bei **report** und **book** eine Titelseite
- Verwendet in der Präambel definierte **\title{...}**, **\author{...}** und **\date{...}**
- Achtung: Bei **pi1** werden die Parameter direkt übergeben:
\maketitle{\Übungsnummer\Tutor:\in\Bearbeiter:\in}

Graphiken einbinden

- **\usepackage{graphicx}** in Präambel (ist in **pi1** enthalten)
- **\includegraphics[Optionen]{Dateiname}**
 - **width**=Breite, z.B. **width=5cm** oder **width=0.8\textwidth**
 - **height**=Höhe, z.B. **height=4cm** oder **height=0.3\textheight**
 - **scale**=Faktor, z.B. **scale=0.5**
 - **angle**=Winkel, z.B. **angle=90**
- Mehrere Optionen sind möglich und werden durch Kommas getrennt
- **pdfLaTeX** unterstützt die Grafikformate **.jpg** (Fotos), **.png** (z.B. Screenshots) und **.pdf** (Zeichnungen)

Umgebungen: Demo



The screenshot shows the TeXstudio interface with a LaTeX document open. The file path is `/Users/roefer/Entwicklung/Docs/2022/CodeRelease/text/videoanalysischallenge.tex`. The left pane displays the LaTeX code, and the right pane shows the generated PDF output.

Code Preview:

```
\begin{itemize}
\item{Achievement of the long/short-term goal}
\item{execution time (and hardware requirements)}
\item{metrics (accuracy/precision/recall)}
\item{technical strength and}
\item{novelty.}
\end{itemize}

\begin{table}
\centering
\begin{tabular}{||c|c||}
\hline
Teams & Points & Rank \\
\hline\hline
B-Human & 25 & 1 \\
SPQR Team & 25 & 1 \\
NomadZ & 21.9 & 3 \\
Nap Devils & 18.1 & 4 \\
RoboFireann & 13.5 & 5 \\
R-ZWEI KICKERS & 5 & 6 \\
\hline
\end{tabular}
\caption{The points and the ranking of the Video Analysis Challenge at the RoboCup 2022.}
\label{vac_results}
\end{table}
```

PDF Preview:

The PDF page features a red header with the B-Human logo and the text "Team Report and Code Release 2022". Below the header is a photograph of several white NAO robots standing on a green soccer field, each wearing a red vest with the B-Human logo. At the bottom of the page, there is a list of names and their roles.

Thomas Röfer^{1,2}, Tim Laus²,
Finn Marvin Ewers², Enrico Göhrs², Michelle Gusev¹, Anna Hasselbring¹,
Jo Lienhopp¹, Ayleen Löhrensen¹, Yannik Meinken¹, Philip Reichenberg¹,
Laurens Schiefelebein², Florian Scholz², Sina Schreiber¹, Simon Werner²

Umgebungen

- **\begin{umgebung} ... \end{umgebung}** definiert einen Bereich im Dokument, in dem eigene Regeln gelten (anderes Layout, andere oder zusätzliche Befehle)
- **document, abstract**: Dokument und Zusammenfassung
- **itemize, enumerate, description**: Aufzählungen
- **tabular, tabbing**: Tabellen
- **figure, table**: Rahmen für „fließende“ Abbildungen und Tabellen
- **verbatim, listinglst**: Roher Text, Quelltext-Listing
- **equation, align**: Gleichungen

Umgebungen: Aufzählungen

- **\begin{itemize} ... \end{itemize}** definiert Bulletpoint-Liste
- **\begin{enumerate} ... \end{enumerate}** definiert nummerierte Aufzählung
- **\begin{description} ... \end{description}** definiert Liste aus Begriffen mit ihrer Definition (**\item[Begriff]** Definition)
- **\item** Definiert einen Aufzählungspunkt
- Können geschachtelt werden

```
\begin{itemize}  
\item A  
\item B  
\item C  
\end{itemize}
```

- A
- B
- C

```
\begin{enumerate}  
\item A  
\item B  
\item C  
\end{enumerate}
```

1. A
2. B
3. C

Tabellen

- **\begin{tabular}{Spalten} ... \end{tabular}** erzeugt eine Tabelle
- Spalten können linksbündig (**l** oder **p{feste breite}**), rechtsbündig (**r**) oder zentriert (**c**) sein
- **|** in der Spaltendefinition fügt senkrechte Linien ein, **\hline** im Tabellenrumpf horizontale
- **&** trennt Zellen, **\n** trennt Zeilen

```
\begin{tabular}{|p{1cm}||rc|rc||}
\hline
Name & Pkt. & Note & Pkt. & Note & best. \\
\hline
Peter-Paul & 70\% & 2,7 & 45\% & 5 & nein \\
\hline
Maria & 100\% & 1 & 100\% & 1 & ja \\
\hline
\end{tabular}
```

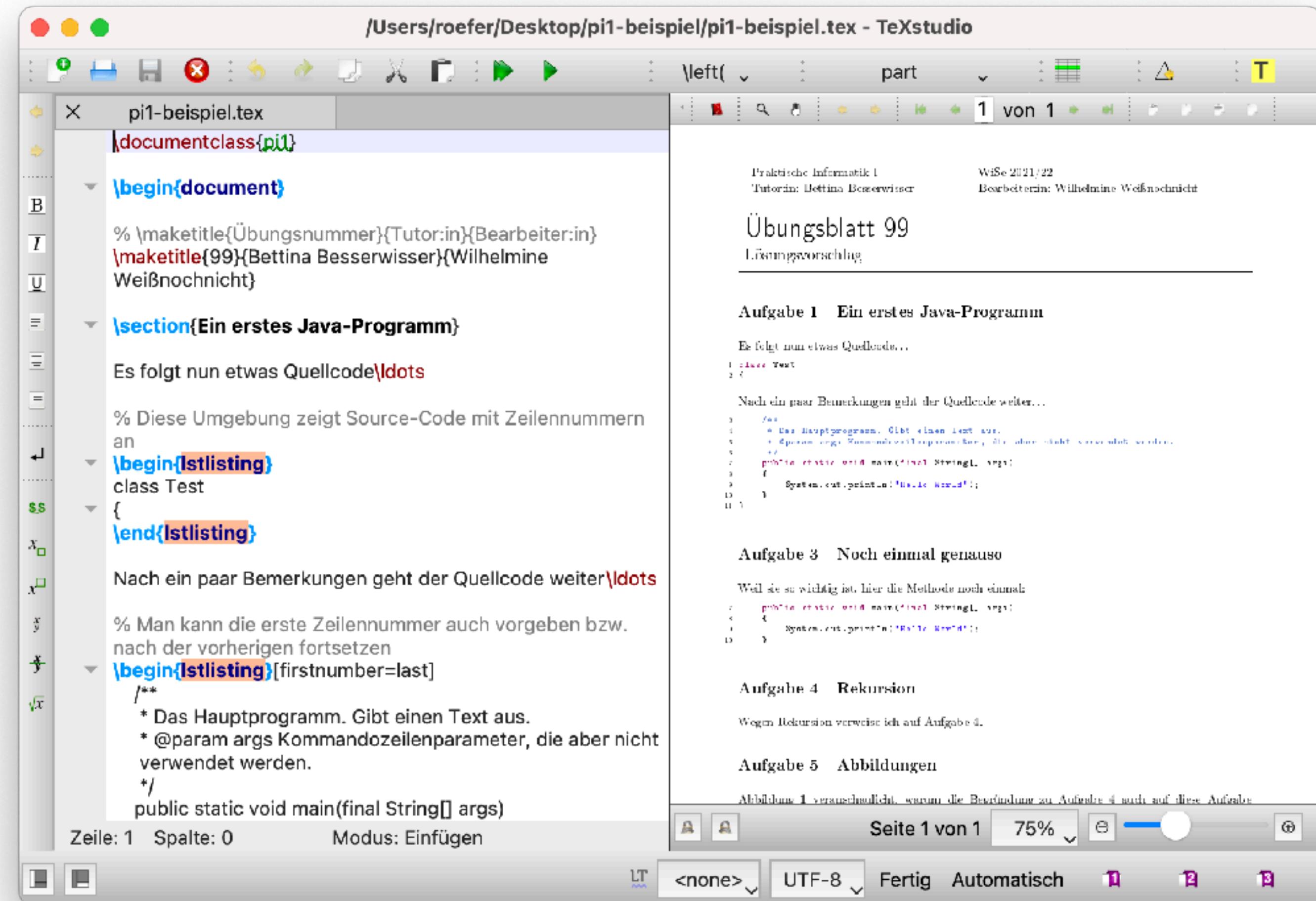
Name	Pkt.	Note	Pkt.	Note	best.
Peter-Paul	70%	2,7	45%	5	nein
Maria	100%	1	100%	1	ja

Fließende Abbildungen und Tabellen

- **\begin{figure}**[Platzierung] ... **\end{figure}**: Rahmen für Abbildungen, die LaTeX an eine passende Stelle platziert
- **\begin{table}**[Platzierung] ... **\end{table}**: Rahmen für Tabellen, die LaTeX an eine passende Stelle platziert
- Gewünschte Platzierung kann hier (**h**), am oberen (**t**) und unteren (**b**) Seitenrand bzw. auf einer eigenen Seite (**p**) sein, wobei mehrere angegeben werden können oder eine mit nachgestelltem **!** forciert werden kann
- **\caption{...}** definiert Beschriftung

```
\begin{figure}[ht]
\centering
\includegraphics[width=0.1\textwidth]{LaTeXLogo.png}
\caption{Das \LaTeX-Logo als Bild}
\end{figure}
```

PI-1-Vorlage: Demo



The screenshot shows the TeXstudio interface with a LaTeX document open. The left pane displays the source code, and the right pane shows the generated PDF output.

Source Code (pi1-beispiel.tex):

```
\documentclass{pi1}

\begin{document}

% \maketitle{Übungsnummer}{Tutor:in}{Bearbeiter:in}
\maketitle{99}{Bettina Besserwisser}{Wilhelmine Weißnochricht}

\section{Ein erstes Java-Programm}

Es folgt nun etwas Quellcode\ldots

% Diese Umgebung zeigt Source-Code mit Zeilennummern
% an
\begin{lstlisting}
class Test
{
\end{lstlisting}

Nach ein paar Bemerkungen geht der Quellcode weiter\ldots

% Man kann die erste Zeilennummer auch vorgeben bzw.
% nach der vorherigen fortsetzen
\begin{lstlisting}[firstnumber=last]
/**
 * Das Hauptprogramm. Gibt einen Text aus.
 * @param args Kommandozeilenparameter, die aber nicht
 * verwendet werden.
 */
public static void main(final String[] args)

```

Zeile: 1 Spalte: 0 Modus: Einfügen

PDF Output:

Praktische Informatik 1 WiSe 21/22
Tutorin: Bettina Besserwisser Bearbeiterin: Wilhelmine Weißnochricht

Übungsblatt 99
Lösungsvorschlag

Aufgabe 1 Ein erstes Java-Programm

Es folgt nun etwas Quellcode...

```
1 class Test
2 {
3
4     /*
5      * Das Hauptprogramm. Gibt einen Text aus.
6      * @param args Kommandozeilenparameter, die aber nicht verwendet werden.
7      */
8     public static void main(final String[] args)
9     {
10         System.out.println("Hello World!");
11     }
12 }
```

Nach ein paar Bemerkungen geht der Quellcode weiter...

```
1 /**
2  * Das Hauptprogramm. Gibt einen Text aus.
3  * @param args Kommandozeilenparameter, die aber nicht verwendet werden.
4  */
5 public static void main(final String[] args)
6 {
7     System.out.println("Hello World!");
8 }
```

Aufgabe 3 Noch einmal genauso

Weil sie so wichtig ist, hier die Methode noch einmal:

```
1 /**
2  * Das Hauptprogramm. Gibt einen Text aus.
3  * @param args Kommandozeilenparameter, die aber nicht verwendet werden.
4  */
5 public static void main(final String[] args)
6 {
7     System.out.println("Hello World!");
8 }
```

Aufgabe 4 Rekursion

Wegen Rekursion verweise ich auf Aufgabe 4.

Aufgabe 5 Abbildungen

Abbildung 1 veranschaulicht, warum die Bezeichnung zu Aufgabe 4 auch auf diese Aufgabe

Seite 1 von 1 75% Fertig Automatisch 1 2 3

Quelltext-Listings

- **\usepackage{listings}** in Präambel (ist in **pi1** enthalten)
- **\begin{lstlisting}[firstnumber=Anzeigezeile]** ... **\end{lstlisting}**: Quelltexte einbinden, Zeilen durchnummerieren und Syntax hervorheben (Zeilenangabe ist optional, **firstnumber=last** setzt vorherige Zählung fort)
- **\lstinputlisting[firstnumber=Anzeigezeile,firstline=Dateizeile,lastline=Dateiz
eile]{Datei}**: Einbinden eines Quelltextes aus einer Datei (alle Zeilenangaben sind optional)

Es folgt nun etwas Quellcode\ldots

```
\begin{lstlisting}
class Test
{
\end{lstlisting}
```

Es folgt nun etwas Quellcode\ldots

```
1 class Test
2 {
```

Mathematikmodus

- TeX kennt Textmodus und Mathematikmodus
- $\$ \dots \$$ wechselt innerhalb einer Zeile in den Mathematikmodus, wobei Formeln eher kompakt gesetzt werden
- **\begin{equation} ... \end{equation}** erzeugt einen Absatz, in dem eine Formel steht (großzügiger gesetzt und nummeriert)
- Viele Befehle funktionieren nur im Mathematikmodus
- \wedge stellt hoch, $_$ stellt tief, mit besonderen Ergebnissen nach bestimmten Zeichen
- TeX kennt sehr viele Zeichen: <http://detexify.kirelabs.org/classify.html>

Die gaußsche Summenformel $\sum_{k=1}^n k = \frac{n^2 + n}{2}$
hier gleich nochmal:
$$\begin{aligned} \sum_{k=1}^n k &= \frac{n^2 + n}{2} \\ \end{aligned}$$

Die gaußsche Summenformel $\sum_{k=1}^n k = \frac{n^2 + n}{2}$ hier gleich nochmal:

$$\sum_{k=1}^n k = \frac{n^2 + n}{2} \quad (1)$$

Querverweise

- **\label{Name}** definiert eine Marke, auf die aus dem Text verwiesen werden kann
 - Der Name muss eindeutig sein
 - **\label{...}** muss hinter dem Befehl stehen, auf den verwiesen werden soll, z.B.
\section{...}\label{...}
- **\ref{Name}** verweist auf eine Marke und setzt die Gliederungsnummer des Elements ein, das vor der Marke verwendet wurde
- **\pageref{Name}** setzt stattdessen die Seitenzahl ein
- Verweisen werden kann auf Überschriften, Abbildungen, Tabellen und Formeln

Literaturverzeichnis

- **\bibliographystyle{ ... }** legt in der Präambel den Stil der Literaturliste fest (**plain**, **abbrv**, **alpha**, **unsrt**, **apalike**)
 - Für deutsche Literaturlisten wird **\usepackage{bibgerm}** verwendet und üblicherweise **ger** vor den Stil geschrieben, z.B. **\bibliographystyle{gerplain}**
- Eine **.bib**-Datei enthält alle zitierbaren Literatureinträge, wobei jeder ein eindeutiges Kürzel hat, z.B. **@inproceedings{mueller-11, ...**
 - **Autor:innen** und **Editor:innen** werden durch **and** voneinander getrennt
- **\cite{Kürzel}** zitiert den Literatureintrag aus dem Text, z.B. **\cite{mueller-11}**
- **\bibliography{Dateiname}** erzeugt eine Literaturliste aus allen zitierten Einträgen, wobei der Dateiname die Endung **.bib** nicht enthält

Literaturverzeichnis

```
@inproceedings{mueller-11,  
author = {Judith Müller and Tim Laue and Thomas Röfer},  
title = {Kicking a Ball -- Modeling Complex Dynamic Motions for Humanoid Robots},  
editor = {Javier Ruiz-del-Solar and Eric Chown and Paul G. Ploeger},  
booktitle = {RoboCup 2010: Robot Soccer World Cup XIV},  
publisher = {Springer},  
volume = {6556},  
pages = {109-120},  
series = {Lecture Notes in Artificial Intelligence},  
year = {2011}  
}
```

1. Müller, J., Laue, T., Röfer, T.: Kicking a ball – modeling complex dynamic motions for humanoid robots. In: del Solar, J.R., Chown, E., Ploeger, P.G. (eds.) RoboCup 2010: Robot Soccer World Cup XIV. Lecture Notes in Artificial Intelligence, vol. 6556, pp. 109–120. Springer (2011)

TeX-Dokumente bauen

- TeX besteht aus mehreren Compilern, die möglicherweise mehrfach in bestimmter Reihenfolge aufgerufen werden müssen (z.B. **pdfLaTeX**, **BibTeX**, **pdfLaTeX**, **pdfLaTeX**)
- Besser: Build-System nutzen
 - In einigen LaTeX-Editoren bereits integriert (z.B. in TeXstudio)
 - Ansonsten gibt es z.B. **latexmk** oder **texify**, die sich in Editoren einbinden lassen
- In TeX-Dateien ist die Groß-/Kleinschreibung von Dateinamen wichtig (**Windows** und **macOS** ignorieren sie, **Linux** aber nicht). Außerdem müssen **relative Pfade** verwendet werden
- Für PI-1 muss der LaTeX-Editor **UTF-8** als Zeichenkodierung verwenden (bei TeXstudio Standard)

Übungsblatt 1

- **zip**-Datei herunterladen und auspacken
- **loesung01.tex** so ergänzen, dass die erzeugte pdf-Datei so aussieht wie Vorgabe
 - Keinen Java-Quellcode tippen, sondern aus **Primes.java** einbinden
 - Verweise nutzen
 - **Referenzen.bib** für Literaturliste nutzen

Übungsblatt 1

Abgabe: nein

Aufgabe 1 Einmal mit L^AT_EX, bitte

Erweitert die Datei *loesung01.tex* so, dass L^AT_EX eine PDF-Ausgabedatei erzeugt, die der zweiten Seite dieses Übungsblatts gleicht. Einzig die Felder <Tutor:in> und <Bearbeiter:in> sollen durch die für euch passenden Daten ersetzt werden.

Folgende Konstrukte sollen eingesetzt werden:

- Hauptüberschriften
- Normaler Text
- Einbinden von externen Quelltextabschnitten. Verweist hierzu auf die beigelegte Datei *Primes.java*¹.
- Fetter Text
- Hervorgehobener Text. Dieser erscheint mit der verwendeten Vorlage kursiv.
- Querverweise
- Formelsatz
- Literaturverweis und Literaturliste mit BibTeX. Bindet hierzu die beigelegte Literaturdatei *Referenzen.bib* ein und verwendet den Literaturstil *gerplain*.
- Fließende Umgebung mit Beschriftung
- Tabellsatz

Tipp. Beachtet, dass der Quelltext für Querverweise mehrfach übersetzt werden muss. Für das Literaturverzeichnis muss zusätzlich auch noch BibTeX ausgeführt werden (*pdflatex*, *bibtex*, *pdflatex*, *pdflatex*). Verwendet ihr *TeXstudio*, werden diese Schritte automatisch ausgeführt.

¹Wer möchte, kann das enthaltene Programm tatsächlich in BlueJ ausprobieren. Dazu die Datei *package.bluej* mit eben diesem öffnen. Über das Kontextmenü von *Primes* könnt ihr *void primes()* ausführen. Der gebogene Pfeil rechts unten im BlueJ-Fenster beendet das Programm wieder. Nichts davon ist aber für das Bearbeiten dieses Übungsblatts notwendig.