

Форматирование в системе L^AT_EX

Пример файла

Листов 31

Аннотация

Настоящий документ является примером \LaTeX файла. \LaTeX (произносится «латех») — замечательная альтернатива программам *Microsoft Word*, *LibreOffice* и другим текстовым редакторам, поскольку:

- относится к свободно распространяемому ПО;
- предназначен для работы в любых операционных системах; при переносе текста с одного компьютера на другой форматирование сохраняется, даже если компьютеры работают под разными операционными системами;
- позволяет автору сосредоточиться на содержании текста, принимая на себя заботы по его оформлению;
- превосходит остальные издательские системы по качеству текстов, содержащих математические формулы.

\LaTeX не является монолитной программой, а состоит из набора пакетов. Дистрибутивы и документация к \LaTeX находятся на страничке проекта: <http://www.ctan.org>.

Также существуют онлайн редакторы \LaTeX , например, *Overleaf* (<https://ru.overleaf.com>). В данном редакторе предусмотрена автоматическая подсказка и подстановка команд, что позволяет работать с ним уже на начальном этапе знакомства с \LaTeX .

СОДЕРЖАНИЕ

1. Как происходит работа с \LaTeX	5
1.1. Пример \TeX файла	6
1.2. Пакеты \LaTeX	6
2. Набор текста	7
2.1. Русификация \LaTeX	7
2.2. Шрифты	7
2.2.1. Выделение текста	7
2.2.2. Размер шрифта	7
2.3. Пробелы	8
2.4. Абзацы	9
2.5. Спецсимволы	10
2.6. Дефисы и тире	10
2.7. Многоточие	10
2.8. Подчеркивание, рамки	10
3. Списки	11
3.1. Ненумерованный список	11
3.2. Нумерованный список	11
3.2.1. Как исключить букву из нумерованного буквенного списка	11
3.3. Многоуровневый список	12
4. Формулы	14
5. Заголовки	16
6. Перекрестные ссылки	17
7. Рисунки	18
7.1. Процедура <code>figure</code>	18
8. Таблицы	21
8.1. Процедура <code>tabular</code>	21
8.2. Процедура <code>table</code>	22
8.3. Дополнительные возможности	23

8.3.1. Пакет <code>multirow</code>	23
8.3.2. Пакет <code>array</code>	23
8.3.3. Пакет <code>makecell</code>	24
8.4. Многостраничные таблицы	24
9. Макет документа	26
10. Титульный лист	27
11. Конвертация <code>tex</code> файла в другие форматы	28
Приложение 1. Перечень принятых обозначений	29
Приложение 2. Перечень команд для задания символов	30

1. Как происходит работа с L^AT_EX

Для начала автор готовит файл с текстом в любом редакторе, оснащённом командами L^AT_EX (настоящий файл создан при помощи редактора T_EX Live). Такие файлы имеют расширение `tex`. Далее файл необходимо обработать при помощи компилятора.

Исходный файл в самом начале должен содержать команду

```
\documentclass[опции]{класс}
```

в которой `[опции]` является необязательным, а `{класс}` — обязательным аргументом, содержащим название класса печатного документа:

- `article` — статья или небольшой отчет;
- `letter` — письмо;
- `report` — статья, разбитая на главы или небольшая книга;
- `book` — книга;
- `proc` — текст в две колонки;
- `slides` — презентация.

Таким образом, команда `\documentclass` определяет основную структуру печатного документа.

Следующей обязательной командой является

```
\begin{document}
```

с которой начинается текст документа. Текст перед `\begin{document}` называется преамбулой и содержит команды настройки выбранного класса печатного документа, а также определения новых команд. Заканчивается текст документа командой

```
\end{document}
```

1.1. Пример \TeX файла

Теперь можно создать простейший \TeX файл:

```
\documentclass{article}
\begin{document}
Example
\end{document}
```

После компиляции появится печатная версия файла, на которую будет выведен результат компиляции — слово `Example`.

1.2. Пакеты \LaTeX

Пакет — это служебный текстовый файл, при помощи которого в документ вносятся дополнения и изменения. Пакеты загружаются в редактируемый документ путем добавления в преамбулу команды

```
\usepackage[опции]{имя пакета}
```

Опции и имена пакетов, если их несколько, перечисляются через запятую.

2. Набор текста

2.1. Русификация L^AT_EX

Чтобы перейти на русскоязычный шрифт, необходимо подключить несколько пакетов:

```
\documentclass{article}
\usepackage[T2A]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage[russian]{babel}
\begin{document}
```

Пример

```
\end{document}
```

В примере в исходный файл были включены:

- `fontenc` — пакет для выбора внутренней кодировки шрифтов L^AT_EX;
- `inputenc` — пакет для указания кодировки текста в исходном файле;
- `babel` — пакет поддержки языков.

2.2. Шрифты

2.2.1. Выделение текста

Для набора **полужирного** текста используется команда `\textbf{текст}` или декларация `{\bf текст}`.

Для *курсивного* начертания используется команда `\textit{текст}` или декларация `{\it текст}`.

Наконец, чтобы получить **моноширинный** шрифт, используется команда `\texttt{текст}` или декларация `{\tt текст}`.

2.2.2. Размер шрифта

Декларации для переключения размера шрифта приведены в таблице 1. Текст, размер которого необходимо изменить, помещается в фигурных скобках сразу после декларации.

Таблица 1 — Декларации, переключающие размер шрифта

Декларация	Название	Образец
<code>\tiny</code>	крошечный	Аа . . . яЯ
<code>\scriptsize</code>	индексный	Аа . . . яЯ
<code>\footnotesize</code>	подстрочный	Аа . . . яЯ
<code>\small</code>	маленький	Аа . . . яЯ
<code>\normalsize</code>	стандартный	Аа . . . яЯ
<code>\large</code>	большой	Аа . . . яЯ
<code>\Large</code>	огромный	Аа . . . яЯ
<code>\LARGE</code>	громадный	Аа . . . яЯ
<code>\huge</code>	грандиозный	Аа . . . яЯ
<code>\Huge</code>	колоссальный	Аа . . . яЯ

Чтобы задать основной размер шрифта, необходимо загрузить пакет `extsizes`, содержащий набор размеров шрифтов и указать размер шрифта в необязательном аргументе:

```
\usepackage[14pt]{extsizes}
```

2.3. Пробелы

Слова в тексте отделяются друг от друга пробелами, при этом не имеет значения, сколько пробелов набрано: ЛАТЭХ автоматически преобразует их в один. Чтобы вручную управлять горизонтальными промежутками, есть специальные команды, описанные в таблице 2.

Таблица 2 — Команды, задающие пробел

Команда	Вид пробела	
<code>\quad</code>	Пробел в 1em	
<code>\qqquad</code>	Пробел в 2em	
<code>\,</code>	«Тонкий пробел», или тонкая шпация	

Продолжение таблицы 2

Команда	Вид пробела	
<code>\:</code>	«Средний пробел»	
<code>\;</code>	«Толстый пробел»	
<code>\!</code>	«Отрицательный тонкий пробел»	
<code>~</code>	Неразрывный пробел	
<code>\/</code>	Корректирующий пробел (используется, например, при переходе от курсива к прямому шрифту)	
<code>\hspace{длина}</code>	Пробел заданной длины	
<code>\hfill</code>	Пробел бесконечно растяжимой длины	

2.4. Абзацы

Абзацы разделяются между собой пустыми строками; чтобы сформировать абзац для первой строки, необходимо загрузить пакет `indentfirst` (команда `\usepackage{indentfirst}` в преамбуле).

Если не найдена подходящая точка для переноса, текст может выйти за границу правого поля:

Здесь текст, который вышел за границу поля из-за команды: `\usepackage{indentf`

В этом случае необходима дополнительная защита от переполнения строки:

```
\begin{sloppypar}
текст абзаца
\end{sloppypar}
```

В случае, когда необходимо подавить абзацный отступ, удобно воспользоваться командой `\noindent`, которая располагается впереди текста абзаца:

```
\noindent текст абзаца
```

Чтобы завершить текущую страницу и начать с новой страницы, используется команда `\newpage`.

2.5. Спецсимволы

Следующие 10 символов:

{ } \$ & # % _ ^ ~ \

являются специальными; чтобы получить первые семь в печатной версии файла, необходимо в исходном тексте поставить перед соответствующим символом знак \ без пробела. Символы ^ и ~ получаются из комбинаций \^{ } и \~{ } в исходном файле, а символ \ — из команды \textbackslash.

2.6. Дефисы и тире

Для получения дефиса (используется в сложных составных словах, например, **северо-запад**), в исходном тексте набирается один знак -.

Для получения короткого тире (используется при задании числовых промежутков, например, 5–15), набираются два знака - (--).

Тире (—) используется в качестве знака препинания в предложениях и состоит из трех знаков - (---).

2.7. Многоточие

Для получения многоточия (...) используется команда \ldots.

Для заполнения пустого пространства в строке точками используется команда \dotfill:

2.8. Подчеркивание, рамки

Чтобы подчеркнуть слово, используется команда \underline{подчеркиваемый текст}.

Чтобы поместить текст в рамку, необходимо использовать команду \fbox{текст}.

3. Списки

3.1. Ненумерованный список

Для составления ненумерованного списка используется процедура

```
\begin{itemize}
\item
\item
...
\end{itemize}
```

3.2. Нумерованный список

Для составления нумерованного списка используется процедура

```
\begin{enumerate}
\item
\item
...
\end{enumerate}
```

По умолчанию нумерованный список является числовым. Чтобы преобразовать его в буквенный, необходимо указать в преамбуле (для всего документа) или непосредственно перед списком (команда действует до конца документа, если ее не перезаписать):

```
\renewcommand{\theenumi}{\asbuk{enumi}}
```

Теперь, чтобы заменить точку после буквы на круглую скобку, необходимо добавить команду

```
\renewcommand{\labelenumi}{\asbuk{enumi}})
```

3.2.1. Как исключить букву из нумерованного буквенного списка

Иногда требуется исключить из списка определенные цифры или буквы (например, по требованию ГОСТа необходимо исключать буквы «з» и «о» из русскоязычного буквенного списка). В этом случае необходимо:

- а) начать список, как обычно, с команды `\begin{enumerate}`;
- б) продолжить список;
- в) ...
- г) ...
- д) ...
- е) ...
- ж) ← закончить список на букве, предшествующей исключаемой (команда `\end{enumerate}`);
- и) начать новый список, дополнив его командой `\setcounter{enumi}{8}`, позволяющей указать, с какого элемента начнется нумерация (в данном случае с восьмого — буквы «и», так как в начале списка находится нулевой элемент);
- к) таким образом, буква «з» исключена из буквенного нумерованного списка;
- л) ...
- ...

3.3. Многоуровневый список

Исходный код для нумерованного многоуровневого списка выглядит следующим образом:

```
\begin{itemize}
\item
\item
  \begin{itemize}
    \item
    \item
    ...
  \end{itemize}
  ...
\end{itemize}
```

\LaTeX допускает четыре уровня вложенности списка:

- Первый уровень
 - Второй уровень
 - * Третий уровень
 - Четвертый (последний) уровень

Чтобы заменить метки, используемые процедурой `itemize` по умолчанию, с помощью команды `\renewcommand` переопределяются команды

<code>\labelitemi</code>	<code>\labelitemii</code>
<code>\labelitemiii</code>	<code>\labelitemiv</code>

Например, команда

```
\renewcommand{\labelitemi}{$\circ$}
```

заменяет элемент первого уровня на символ бинарных операторов \circ (`\circ`). Знаки $\$$ вокруг записи символа означают переход в математический режим форматирования, позволяющий использовать в тексте символы математических операций, а также формулы, о которых речь пойдет в следующем разделе.

Для нумерованного списка элементами второго уровня по умолчанию являются буквы английского алфавита, заключенные в круглые скобки. Их можно заменить на арабские цифры при помощи команды

```
\renewcommand{\theenumii}{\arabic{enumii}}
```

Для замены круглых скобок на закрывающую круглую скобку используется команда

```
\renewcommand{\labelenumii}{\arabic{enumii})}
```

4. Формулы

Чтобы использовать в тексте формулы, необходимо подключить в преамбуле пакет `amsmath`:

```
\usepackage{amsmath}
```

Для небольших формул, которые размещаются внутри абзаца, существует процедура форматирования математических формул `math` и три варианта обращения к ней:

```
\begin{math} ... \end{math}
\left( ... \right)
$ ... $
```

Процедура `equation` производит формулы, расположенные в отдельной строке, и автоматически нумерует их. Существует два варианта обращения к данной процедуре:

```
\begin{equation} ... \end{equation}
$$ ... $$
```

Чтобы отключить нумерацию, необходимо процедуру `equation` заменить на `equation*`.

В таблице 3 приведены команды для задания математических знаков, используемых в формулах.

Таблица 3 — Команды для задания математических знаков

Команда	Пример
<code>\sqrt[степень]{подкоренное выражение}</code>	$\sqrt[3]{a+b}$
<code>\frac{числитель}{знаменатель}</code>	$\frac{a+b}{c}$
<code>выражение^степень</code>	b^2
<code>\cos</code>	$\cos x$
<code>\sin</code>	$\sin x$
<code>\tg</code>	$\operatorname{tg} x$
<code>\ctg</code>	$\operatorname{ctg} x$

Продолжение таблицы 3

Команда	Пример
<code>\exp</code>	\exp
<code>\ln{выражение}</code>	$\ln 2$
<code>\log_степень выражение</code>	$\log_2 10$
<code>\lim_предел</code>	$\lim_{n \rightarrow \infty}$
<code>\min</code>	\min
<code>\max</code>	\max
<code>\sum_{нижний предел}^{верхний предел}</code>	$\sum_{i=1}^n$
<code>\int_{нижний предел}^{верхний предел}</code>	\int_0^∞
<code>(\big(\Big(\bigg(\Bigg(</code>	$((((($
<code>[\big[\Big[\bigg[\Bigg[</code>	$[[[[[$

5. Заголовки

Для класса документа вида `article` существуют следующие команды секционирования:

```
\section{Название раздела}  
\subsection{Название подраздела}  
\subsubsection{Название пункта}  
\paragraph{Название параграфа}
```

Каждая из приведенных выше команд начинает новую составную часть документа, то есть присваивает ей номер, печатает заголовок и вносит этот заголовок в оглавление.

Для печати оглавления используется команда `\tableofcontents`.

6. Перекрестные ссылки

Чтобы сослаться на раздел, рисунок, таблицу, формулу или пункт перечня, необходимо поставить метку:

```
\label{метка}
```

после команды именования соответствующего окружения. В случае таблиц или картинок такой командой является `\caption`, в случае пунктов перечня — `\item`.

Метка представляет любую комбинацию латинских букв, цифр и некоторых знаков препинания. В частности при составлении меток удобно использовать двоеточие (:) и дефис (-).

Когда ссылки идут через метку, то номер раздела (команда `\ref`) и номер страницы (команда `\pageref`) определяется L^AT_EX автоматически.

Например, для заголовка пункта 2.2.2. была поставлена метка

```
\label{sec:text:font:size}
```

где через двоеточие указаны наименования раздела, подраздела и, наконец, пункта. Для ссылки на данный пункт используется команда

```
\ref{sec:text:font:size}
```

7. Рисунки

Чтобы работать в L^AT_EX с графическими изображениями, необходимо загрузить пакет `graphicx` (см. 1.2.). Также предварительно нужно создать каталог(и) для рисунков и указать путь в преамбуле:

```
\graphicspath{{каталог 1/}{каталог 2/}...}
```

Расширения графических файлов задаются при помощи команды `\DeclareGraphicsExtensions{exts}`, где аргумент `exts` — список расширений имен файлов, перечисленных через запятую, например

```
\DeclareGraphicsExtensions{.png, .jpg}
```

Чтобы разместить в тексте изображение, используется команда `\includegraphics{}`, где в фигурных скобках указывается имя файла. Ниже приведен пример использования данной команды:

Исходный код

Кнопка `\includegraphics{compile}` запускает процесс компиляции.

Вывод на печать

Кнопка  запускает процесс компиляции.

7.1. Процедура figure

Для автоматизированного размещения рисунков существует процедура `figure`, рассматривающая рисунок как плавающий объект (если для объекта нет места на текущей странице, он переносится на следующую). Ниже приведен пример использования процедуры `figure`:

```
\begin{figure}[htb]
\centering % Размещение по центру
\includegraphics[width=\textwidth]{имя файла}
\caption{Название рисунка} % Подрисуночная подпись
\label{fig:label} % Метка
\end{figure}
```

Пример изображения представлен на рис. 1.



Рисунок 1 — Пример изображения

Необязательный аргумент процедуры `figure` задает способы размещения плавающего объекта:

- `h` — разместить по возможности здесь же;
- `t` — разместить в верхней части страницы;
- `b` — разместить в нижней части страницы;
- `r` — разместить на отдельной странице, где нет ничего кроме плавающих объектов.

Если указано несколько букв, возможен любой из предусматриваемых этими буквами вариантов размещения.

Команда `\caption{}` создает подпись к рисунку и печатает порядковый номер рисунка. Для работы с подписями к плавающим объектам необходимо загрузить пакет `caption` (см. 1.2.).

По умолчанию в \LaTeX подрисуночная подпись начинается со слова Рис., после которого следует двоеточие и название рисунка:

Рис.: Название рисунка

Чтобы представить подрисуночную подпись в виде

Рисунок — Название рисунка

необходимо в преамбуле добавить команды:

```
\DeclareCaptionLabelSeparator{dash}{ --- }  
\captionsetup{labelsep=dash}
```

и добавить опцию при подключении пакета `caption`:

```
\usepackage[figurename=Рисунок]{caption}
```

`\textwidth` — это переменная, хранящая значение длины, равной ширине текста. Размерами рисунка также можно управлять через параметр `scale`, например:

```
\includegraphics[scale=0.25]{имя файла}
```

8. Таблицы

8.1. Процедура `tabular`

Процедура `tabular` задает таблицу. В обязательном аргументе, помещаемом в фигурных скобках непосредственно после команды `\begin{tabular}`, задается последовательность букв (по одной на столбец), обозначающих выравнивание текста в ячейке:

`l` — выравнивание слева;

`c` — выравнивание по центру;

`r` — выравнивание справа.

`p{ширина}` — еще один способ задать столбец; означает абзац текста, ширина которого задается в фигурных скобках.

Символ `&` делит данные на ячейки. Разделительные линии между столбцами задаются с помощью вертикальной черты `|`. Горизонтальные линии создаются при помощи команды `\hline`. Строки в теле процедуры разделяются командой `\\`. Заканчивается процедура командой `\end{tabular}`.

Ниже приведен исходный код для создания таблицы в L^AT_EX:

```
\begin{tabular}{|l|l|} % Два столбца, выравнивание слева
\hline % Горизонтальная линия
{\bf Заголовок 1} & {\bf Заголовок 2} \\ % Заголовок
\hline
ячейка 1/1 & ячейка 1/2 \\ \hline % Тело таблицы
ячейка 2/1 & ячейка 2/2 \\ \hline
ячейка 3/1 & ячейка 3/2 \\ \hline
\end{tabular}
```

В результате компиляции вышеприведенного кода получится таблица:

Заголовок 1	Заголовок 2
ячейка 1/1	ячейка 1/2
ячейка 2/1	ячейка 2/2
ячейка 3/1	ячейка 3/2

8.2. Процедура table

Процедура `table` служит для задания названия таблицы, метки для ссылки, параметров размещения таблицы относительно остального текста и автоматической нумерации.

Для работы с подписью к таблице необходимо подключить пакет `caption` (см. 1.2.). Тогда команда `\captionsetup{singlelinecheck=off}` позволит выровнять по левому краю подпись, которая по умолчанию располагается в центре страницы; команды `\setlength{\abovecaptionskip}{...}` и `\setlength{\belowcaptionskip}{...}` позволят управлять вертикальными отбивками вокруг подписи.

```
\begin{table}[htb] % Размещение таблицы
\captionsetup{singlelinecheck=off} % Подпись выровнена по левому краю
\setlength{\abovecaptionskip}{0pt} % Отступ между названием и таблицей
\setlength{\belowcaptionskip}{5pt} % Отступ между текстом и названием
\caption{Название таблицы} % Название таблицы
\label{tab:label} % Метка
\begin{tabular}
...
\end{tabular}
\end{table}
```

Теперь после компиляции таблица будет пронумерована, и к ней добавятся название и метка для ссылки (см. таблицу 4).

Таблица 4 — Название таблицы

Заголовок 1	Заголовок 2
ячейка 1/1	ячейка 1/2
ячейка 2/1	ячейка 2/2
ячейка 3/1	ячейка 3/2

8.3. Дополнительные возможности

8.3.1. Пакет `multirow`

Пакет `multirow` (см. 1.2.) позволяет объединять ячейки таблицы. В аргументе команды `\multirow` задается, сколько ячеек нужно объединить, ширина конечной ячейки и непосредственно текст ячейки.

Команда `\cline{...}` позволяет прочертить горизонтальную линию не по всей ширине таблицы, а охватывая заданное число колонок. Номера первой и последней из колонок, которые необходимо соединить горизонтальной линией, указываются в фигурных скобках.

Ниже приведен исходный код для создания таблицы с объединенными ячейками.

```
\begin{tabular}{|c|c|c|}
\hline
\multirow{2}{3cm}{\hspace{0.25cm} Заголовок}
& \multicolumn{2}{c|}{Общий заголовок} \\ \cline{2-3}
& столбец 1 & столбец 2 \\ \hline
строка 1 & ячейка 1/1 & ячейка 1/2 \\ \hline
строка 2 & ячейка 2/1 & ячейка 2/2 \\ \hline
\end{tabular}
```

Результатом компиляции данного кода будет таблица:

Заголовок	Общий заголовок	
	столбец 1	столбец 2
строка 1	ячейка 1/1	ячейка 1/2
строка 2	ячейка 2/1	ячейка 2/2

8.3.2. Пакет `array`

При подключении пакета `array` (см. 1.2.) появляется простой способ борьбы с примыканием горизонтальных линеек к тексту: надо присвоить ненулевое значение параметру `\extrarowheight`. Это — величина, которая добавляется к высоте каждой строки таблицы, по умолчанию равная нулю. Для лучшего отделения линеек от текста хорошо присвоить ей значение 2–3 пункта:

```
\setlength{\extrarowheight}{3pt}
```

Также при подключении пакета `array` наряду с выражением `p{...}`, можно использовать выражения `m{...}` и `b{...}`. Как и `p{...}`, они указывают, что в колонке находится абзац текста ширины, заданной в фигурных скобках. Отличается способ выравнивания:

`p{...}` — выравнивание по верхней строке абзаца;

`b{...}` — выравнивание по нижней строке абзаца;

`m{...}` — абзац выравнивается по середине своей высоты.

8.3.3. Пакет `makecell`

Для создания многострочных ячеек в таблицах удобно использовать команду `\makecell`. Предварительно в преамбуле необходимо подключить пакет `makecell` (см. 1.2.).

8.4. Многостраничные таблицы

Для создания многостраничной таблицы необходимо подключить пакет `longtable` (см. 1.2.). Ниже находится пример исходного кода для размещения многостраничной таблицы:

```
\begin{longtable}[htb]{|m{5cm}|m{5cm}|} % Ширина колонок 5 см
\captionsetup{singlelinecheck=off}
\caption*{\hbox to 3.7in {\tablename \; \thetable{}} --- Название таблицы}
\label{tab:label} \\ % Метка таблицы
\hline % Горизонтальная линия
{Заголовок 1} & {Заголовок 2} \\
\hline
\endfirsthead % Конец заголовка на первой странице
\caption*{\hbox to 3.8in {Продолжение таблицы \thetable{}} \hfill }} \\
\hline
{Заголовок 1} & {Заголовок 2} \\
\hline
\endhead % Конец заголовка на последующих страницах
ячейка 1/1 & ячейка 1/2 \\ \hline
ячейка 2/1 & ячейка 2/2 \\ \hline
```



```
ячейка 3/1 & ячейка 3/2 \\ \hline
...
ячейка n/1 & ячейка n/2 \\ \hline
\end{longtable}
```

По умолчанию многостраничные таблицы размещаются по центру. Для выравнивания по левому краю необходимо до начала таблицы добавить команду

```
\LTleft=0pt
```

9. Макет документа

Формат листа можно задать в первой команде документа, например:

```
\documentclass[a4paper, times new roman, oneside]{article}
```

где `a4paper` — установка размера листа бумаги A4 (по умолчанию `letter`); `oneside` — форматирование документа для односторонней печати.

Для управления размерами макета через выставку полей необходимо загрузить пакет `vmargin` и указать в преамбуле значения полей:

```
\setmarginsrb{левое поле}{верхнее поле}{правое поле}{нижнее поле}{высота верхнего колонтитула}{расстояние от текста до верхнего колонтитула}{высота нижнего колонтитула}{расстояние от текста до нижнего колонтитула}
```

Конкретные значения аргументов команды `\setmarginsrb` могут быть следующими:

```
\setmarginsrb{20mm}{25mm}{10mm}{15mm}{0pt}{0mm}{14pt}{10mm}
```

Команда `\pagestyle{}` задает стиль страницы:

`empty` — страница выводится без каких-либо колонтитулов — только текст;

`plain` — выводится только номер страницы в нижнем колонтитуле;

`headings` — в верхнем колонтитуле выводится номер страницы и информация, определяемая классом документа.

10. Титульный лист

Титульный лист печатается при помощи команды `\maketitle`. Параметры титульного листа задаются в преамбуле:

```
\title{Название}
```

```
\author{Автор}
```

```
\date{Дата}
```

Чтобы очистить стиль титульной страницы, используется команда `\thispagestyle{empty}`.

11. Конвертация tex файла в другие форматы

Чтобы конвертировать файл формата `tex` в формат `html` (в кодировке `utf8`), нужно:

- а) скопировать `filename.tex` в папку `C:\Users\Username`;
- б) скопировать рисунки в папку `C:\Users\Username\Изображения`;
- в) набрать в командной строке

```
make4ht -u filename.tex
```

или

```
htlatex filename.tex
```

Чтобы создать файл с расширением `odt`, нужно:

- а) скопировать `filename.tex` в папку `C:\Users\Username`;
- б) скопировать картинки в папку `C:\Users\Username\Изображения`;
- в) набрать в командной строке

```
make4ht -f odt filename.tex
```

*ПРИЛОЖЕНИЕ 1**Справочное***Перечень принятых обозначений**

В тексте документа используются следующие соглашения:

- команды и вывод в печатную версию приведены **моноширинным** шрифтом (символы имеют равную ширину);
- текст после символа **%** в примерах исходного кода означает комментарий и выделен **красным** цветом.

ПРИЛОЖЕНИЕ 2

Справочное

Перечень команд для задания символов

Для получения необходимого символа необходимо задающую его команду поместить в процедуру форматирования математических операций (например, \$ \dots \$).

Символ	Команда
+	+
—	-
*	*
±	\pm
×	\times
÷	\div
○	\circ
●	\bullet
▷	\triangleright
◁	\triangleleft
▽	\bigtriangledown
△	\bigtriangleup
○	\bigcirc
<	<
>	>
=	=
:	:
≤	\le
≥	\ge
≠	\ne
≈	\approx
≪	\ll
≫	\gg

Продолжение перечня команд для задания символов

Символ	Команда
\parallel	<code>\parallel</code>
\rightarrow	<code>\to</code>
\leftarrow	<code>\gets</code>
\uparrow	<code>\uparrow</code>
\downarrow	<code>\downarrow</code>
∞	<code>\infty</code>
\S	<code>\S</code>