

Печатный документ

Структура

Преамбула

Преамбула — вводная часть исходного файла, предшествующая самому документу. Первая команда, которая может быть записана в преамбуле, это `\documentclass[опции]{класс}[дата]`. Обязательный аргумент *класс* определяет класс документа, а необязательный аргумент *опции* — дополнительные параметры, если опций несколько, то они разделяются запятыми. С помощью ещё одного необязательного аргумента *дата* можно указать дату выпуска наиболее старой версии класса, пригодного для компиляции. Дата записывается в формате гггг/мм/дд.

После определения класса документа можно подключить дополнительные пакеты с помощью команды `\usepackage[опции]{пакет}[дата]`. Эта команда действует аналогично предыдущей, только в обязательном аргументе можно указывать несколько пакетов, хотя иногда это вовсе неуместно (поскольку у разных пакетов разные опции).

Титульный лист

Для автоматической генерации титульного листа используется декларация `\maketitle`. Информация для заголовка берётся из команд:

`\title{Заголовок}` — **обязательна**;

`\author{Автор}` — **обязательна**, можно между «авторами» вставлять декларацию `\and`;

`\date{дата}` — если не указать, то ставит текущую дату.

В аргументах этих команд можно использовать команду `\thanks{примечания}`.

Для составления титульного листа вручную используется окружение `titlepage`, которое создаёт новую страницу, устанавливает счётчик страниц на 1, но не печатает номер страницы. На странице печатается текст, помещённый внутри окружения.

Аннотация (abstract)

Для аннотации используется окружение `abstract`, оно начинает новую страницу, если включена опция документа `titlepage`. Текст внутри окружения выделяется дополнительными отступами слева и справа.

Секционирование (рубрикация)

Большинство классов предназначены для написания статей, книг, журналов, сборников. Такие публикации требуют строгой структуризации текста. Для этого пользователю достаточно правильно использовать команды рубрикации, которые перечислены в таб. 1 по старшинству. Обязательный аргумент *загол.*

Таблица 1.

Стандартные команды рубрикации

Команда	Рубрика
<code>\part[альт.загол.]{загол.}</code>	Часть
<code>\chapter[альт.загол.]{загол.}</code>	Глава
<code>\section[альт.загол.]{загол.}</code>	Раздел
<code>\subsection[альт.загол.]{загол.}</code>	Подраздел
<code>\subsubsection[альт.загол.]{загол.}</code>	Подподраздел
<code>\paragraph[альт.загол.]{загол.}</code>	Параграф
<code>\subparagraph[альт.загол.]{загол.}</code>	Подпараграф

это заголовок рубрики, а необязательный *альт.загол.* используется в том случае, когда надо задать альтернативный заголовок, который будет фигурировать в содержании и в колонтитулах. Все эти рубрики имеют разновидность со звёздочкой (*). Такая команда создаёт рубрику без номера и, естественно, не имеет дополнительного аргумента, т. к. *рубрики без номера не появляются в содержании и колонтитулах.*

Аргумент, который передаётся в содержание (колонтитул), является подвижным, поэтому неустойчивые команды, используемые в таких аргументах, должны быть защищены с помощью декларации `\protect`.

Нумерация рубрик устроена иерархически, т. е. счётчику старшей рубрики подчинены счётчики младших рубрик. Исключение составляет `part`, эта рубрика необязательна и никак не влияет на нумерацию глав. В классах `article` и `proc` не определена рубрика `chapter`, для них самой старшей и обязательной рубрикой является `section`. Класс `letter` вообще не имеет рубрик. Счётчик `secnumdepth` — определяет уровень самого младшего раздела, использующего нумерацию.

Если в документе имеются приложения, то для них используется декларация `\appendix`. Она изменяет способ нумерации, а при использовании рубрики `chapter` пишет слово «Приложение» вместо «Глава». Вообще, все ключевые слова рубрик задаются следующими командами-логосами:

`\abstractname`, `\partname`, `\chaptername`, `\appendixname`, которые должны переопределяться в файле русификации.

Оглавление (содержание)

Оглавление, списки рисунков и таблиц делают соответствующие декларации:

`\tableofcontents` — оглавление;

`\listoffigures` — список рисунков;

`\listoftables` — список таблиц.

При этом создаются и используются служебные файлы¹ `.toc`, `.lof` и `.lot`, соответственно.

Для «ручной» коррекции содержаний используются следующие команды:

`\addcontentsline{файл}{формат}{запись}` — добавление *записи* в соответствующий *файл* в выбранном *формате*. Форматы могут быть следующими:

для файлов `.toc` — `part`, `chapter`, `section`, `subsection`, `subsubsection`, `paragraph`, `subparagraph`; для файлов `.lof` — `figure`;

для файлов `.lot` — `table`.

`\addtocontents{файл}{запись}` — добавление *записи* в соответствующий *файл*. Здесь в качестве записи можно использовать любой текст в т. ч. и форматизирующие команды.

Для внесения записи с номером (например, раздела) можно использовать команду `\numberline{номер}{запись}` с предварительной командой `\protect: \protect\numberline{номер}{запись}`.

Для составления оглавления полезно использовать счётчик `tocdepth`, который определяет уровень самого младшего раздела, входящего в оглавление.

ЗАПОМНИТЕ: *запись* в приведённых командах является подвижным аргументом!

Библиография (список литературы)

Список литературы формируется с помощью окружения `thebibliography`:

```
\begin{thebibliography}{текст}
  \bibitem[метка]{имя} данные
  \bibitem[метка]{имя} данные
  .
  .
  .
  \bibitem[метка]{имя} данные
\end{thebibliography}
```

Обязательный аргумент *текст* служит для задания левого отступа: сам текст не пишется, а добавляется отступ, равный ширине этого *текста*. Если *метка* в аргументе `\bibitem` не задана, то записи списка нумеруются порядковыми номерами. Для задания *имени* литературного источника можно использовать любые символы, кроме `'`, `,`.

¹ Служебные файлы создаются/обновляются ЛАТ_EXом по мере необходимости, если только, в преамбуле не стоит декларация `\nofiles`. Обычно эта декларация используется на самом последнем этапе подготовки документа, для внесения последних поправок (если в этом есть необходимость).

При записи данных литературного источника можно использовать декларацию `\newblock`, которую следует вставлять между логическими блоками. При использовании стандартных классов, в этом месте будет добавлен дополнительный пробел. При наличии опции класса `openbib` каждый блок будет напечатан с новой строки, а строки внутри блока наделены дополнительным левым отступом на величину `\bibindent`.

Цитирование литературы в тексте производится командой `\cite[текст]{имя}`, при этом, если указан *текст*, то он вставляется, через запятую, после индекса источника. Можно в одной команде указывать несколько источников, перечисляя их через запятую. Для цитирования литературы используется механизм перекрёстного цитирования (через вспомогательный файл `.aux`).

Механизм перекрёстного цитирования

TeX обладает уникальным механизмом перекрёстного цитирования, его отличает простота использования и высокая надёжность.

Для того, чтобы сослаться на ту или иную часть текста, необходимо поставить *метку* в этой части с помощью команды `\label{метка}`.

Для ссылок существуют две стандартные команды:

`\ref{метка}` — вставляет значение счётчика¹, изменённого перед соответствующей командой `\label`.

`\pageref{метка}` — вставляет номер страницы, на которой стоит соответствующая команда `\label`.

Цитирование литературы, описанное в предыдущем разделе, использует тот же механизм. Каждая команда `\label` (так же как и команда `\bibitem`) записывает во вспомогательный файл `.aux` при первой компиляции соответствующую команду. Во время второй компиляции необходимый номер считывается из этого файла и вставляется в `.dvi` файл. Поэтому, при изменении ссылки необходимо две компиляции.

ЗАПОМНИТЕ: метку надо ставить сразу же после помечаемого объекта!

Большие документы

Условная компиляция

Если в Вашем документе есть текст, который в некоторых случаях должен появиться при печати, а в некоторых — нет, то для этого можно задействовать **условную компиляцию**. Необходимый блок надо окружить командными скобками `if`: `\iftrue` ... `\fi` или `\iffalse` ... `\fi`. Первые командные скобки разрешают компиляцию, вторые — запрещают. Можно внутри скобок `if` использовать оператор `\else`, который имеет тот-же смысл (инверсия), что и в программировании.

Включение файлов

Большие документы неудобно писать в одном файле, т. к. при работе с большими файлами замедляется работа текстового редактора и увеличивается время компиляции².

Включать файлы в основной документ можно двумя способами. Можно использовать команду `\input{файл}`, где *файл* можно указать как локальное имя или как полное имя, при этом, для описания пути используется слэш `'/'` (как в ОС UNIX), если файл имеет расширение `.tex`, то его можно не писать. Действие этой команды аналогично тому, как если-бы вместо команды было написано содержимое файла.

Второй способ больше подходит для создания таких документов, как книг или больших отчётов. Документ разбивается на логически законченные части (обычно главы), которые записываются в отдельных файлах. Создаётся *главный документ*, который включает в себя преамбулу, окружение `document`, при необходимости — команды генерации содержаний и команды `\include{файл}` для всех используемых файлов. В преамбулу помещается команда `\includeonly{список файлов}`, в которой перечисляются те файлы, которые должны быть откомпилированы. Содержимое каждого откомпилированного файла будет начинаться с новой страницы, при этом, сохранится вся необходимая информация для перекрёстного цитирования.

Пример проекта (главного файла):

¹Например, номер раздела. О счётчиках подробнее будет рассказано ниже.

²Время компиляции увеличивается прямопропорционально размеру файла.

```
\documentclass[a4paper,openbib]{report}
\usepackage[cp1251]{inputenc}
\usepackage[russian]{babel}

\includeonly{%
%fonts,
%symbols,
%ex1,
%pages,
%ex2,
%format,
%ex3,
doc,
%ex4,
}

\begin{document}
  \tableofcontents

\include{com}
\include{boxes}
\include{fonts}
\include{symbols}
\include{errors}
\include{ex1}
\include{pages}
\include{msdos}
\include{ex2}
\include{format}
\include{ex3}
\include{doc}
\include{ex4}

\bibliographystyle{plain}
\bibliography{xbib}

\end{document}
```