

Стилевой файл kprj, описание применения

Введение

Стилевой файл `kprj.sty` предназначен для оформления в среде \LaTeX отчетов (расчетно-пояснительных записок) по лабораторным работам, курсовым работам и проектам, а также выпускным квалификационным работам бакалавров, магистров, специалистов, научно-квалификационным работам аспирантов.

Стиль `kprj` не использует специальных шрифтовых настроек, однако переопределяет команды заголовков и формирует специальные страницы (титульный лист, задание, календарный план), включаемые в соответствующий документ. Вся необходимая информация указывается в преамбуле файла `Tex`. Далее используются стандартная команда генерации титульного листа `\maketitle` и стандартные команды заголовков `\section`, `\subsection`.

Стилевой файл рассчитан на класс `article`. Типичная команда

```
\documentclass[12pt,a4paper]{article}
```

Конкретный вариант документа указывается специальной командой в преамбуле. Также в преамбуле указываются и другие данные для титульного листа: название документа, автор, руководитель, факультет, кафедра и т.д.

1. Выбор типа документа

Для определения типа документа в преамбуле указывается специальная команда:

- `\LabWork` — отчет по лабораторной работе;
- `\KursWork` — расчетно-пояснительная записка по курсовой работе;
- `\KursProject` — расчетно-пояснительная записка по курсовому проекту;
- `\VKR` — расчетно-пояснительная записка по выпускной квалификационной работе бакалавра, магистра или специалиста;
- `\NKR` — расчетно-пояснительная записка по научно-квалификационной работе аспиранта.

Тип документа влияет на вид титульного листа и обеспечивает генерацию дополнительных страниц (задание, календарный план).

2. Титульный лист

Для формирования титульного листа необходимо обеспечить нужную информацию:

- `\title{<Название>}` — наименование работы;
- `\author{<И.О. Фамилия>}` — автор работы;
- `\group{<группа>}` — название группы;
- `\faculty{<факультет>}` — факультет;
- `\chair{<кафедра>}` — название кафедры (например: Математическое моделирование);
- `\chief{<И.О. Фамилия>}` — инициалы и фамилия научного руководителя (инициалы впереди);
- `\consultant{<И.О. Фамилия>}` — консультант (команда указывается, если консультант в работе имеется);

- `\workyear{<год>}` — год публикации работы;

Для титульного листа по лабораторной работе дополнительно указывается:

- `\discipline{<Название дисциплины>}` — наименование дисциплины;
- `\LabWorkNo{<номер>}` — номер лабораторной работы (необязательно);
- `\variant{<номер>}` — номер варианта лабораторной (необязательно).

Для титульного листа по курсовой работе (проекту) дополнительно указывается:

- `\discipline{<Название дисциплины>}` — наименование дисциплины.

Для титульного листа по выпускной (научной) квалификационной работе дополнительно указывается:

- `\inspector{<И.О. Фамилия>}` — инициалы и фамилия нормоконтролера.

Все эти данные указываются в преамбуле документа. Фактическая генерация страниц происходит по команде `\maketitle`, которую надо указать в начале документа (сразу после команды `\begin{document}`).

3. Задание на выполнение работы и календарный план

Задание на выполнение работы формируется для всех типов документа, кроме лабораторных. Для курсовых работ и проектов оно занимает одну страницу, для квалификационных работ две.

Для формирования задания необходимы дополнительные данные:

- `\authorfull{<Фамилия Имя Отчество>}` — фамилия, имя, отчество автора работы (полностью в указанном порядке);
- `\facultyshort{<аббрев.>}` — аббревиатура факультета (например: ФН);
- `\chairshort{<аббрев.>}` — аббревиатура кафедры (например: ФН-12);
- `\chairhead{<И.О. Фамилия>}` — инициалы и фамилия заведующего кафедрой;
- `\trend{<текст>}` — направленность работы (учебная, исследовательская, практическая, производственная и др.);
- `\themesource{<текст>}` — источник тематики работы (кафедра, предприятие, НИР).

Все эти данные могут быть опущены, тогда в бланк задания их следует вписать ручкой.

Календарный план формируется в случае выпускной (научной) квалификационной работы. Дополнительных данных для его формирования не требуется.

4. Содержание и заголовки

Содержание вставляется в документ стандартной командой `\tableofcontents`. Этот раздел автоматически получает заголовок «Содержание». Пользователь может изменить заголовок командой `\setcontentsname`, например:

```
\setcontentsname{Оглавление}
```

Также можно изменить оформление содержания командами:

- `\contsectionfont` — шрифт для имен разделов (например `\bfseries`);
- `\contsubsectionfont` — шрифт для имен подразделов;
- `\setcounter{secnumdepth}{1}` — установить глубину (количество уровней заголовков) в содержании;
- `\SkipContentsbreak` — подавить разрыв страниц после содержания.

Стиль рассчитан на заголовки трех уровней, которые указываются командами `\section`, `\subsection`, `\subsubsection`. Можно изменять расстояние перед заголовком, после заголовка и шрифт заголовка. По умолчанию действуют определения:

```
\def\secsepbefore{3.5ex plus 1ex minus .2ex}
\def\secsepafter{2.3ex plus.2ex}
\def\sectionfont{\large\bf}

\def\subsecsepbefore{3ex plus 1ex minus .2ex}
\def\subsecsepafter{2ex plus 0.3ex minus 0.2ex}
\def\subsectionfont{\bf}

\def\subsubsecsepbefore{3ex plus 1ex minus .2ex}
\def\subsubsecsepafter{2ex plus 0.3ex minus 0.2ex}
\def\subsubsectionfont{\it}
```

Уравнения и рисунки по умолчанию нумеруются сплошной нумерацией. Можно указать нумерацию в пределах раздела командами `\equationnumbering{section}` `\figurenumbering{section}`.

5. Рисунки и таблицы

Стиль `kprj` не имеет специальных команд для подключения в текст и размещения рисунков, а также для оформления таблиц. Используются стандартные механизмы. Стиль автоматически подключает пакет `graphicx`.

Для размещения рисунка можно использовать конструкцию

```
\begin{figure}[ht]
\centerline{\includegraphics[scale=0.4]{emblema}}
\vskip2mm
\caption{Эмблема}\label{fig1}
\end{figure}
```

В команде `\includegraphics` можно не указывать расширение файла, если в преамбуле эти расширения были описаны. Пример такого описания:

```
\DeclareGraphicsExtensions{.eps,.bmp}
```

Следует иметь в виду, что типы подключаемых графических файлов должны соответствовать компилятору TeX. Стандартный компилятор, который создает DVI-файл, поддерживает большинство графических форматов (кроме PDF). В то же время PDFTeX поддерживает лишь форматы `.jpg`, `.png`, `.pdf`. Стиль поддерживает условную конструкцию, в которой команды выбираются в зависимости от типа компилятора:

```
\ifpdftex
\DeclareGraphicsExtensions{.pdf}
\else
\DeclareGraphicsExtensions{.eps}
\fi
```

TeX может считывать размеры графического файла из самого файла только для файлов формата `.eps`. В остальных случаях необходимо указывать фактические размеры файла. Это можно сделать с помощью параметров `natwidth` и `natheight`, например:

```
\includegraphics[natwidth=60mm,natheight=40mm,scale=0.4]{pngfig}
```

Фактические размеры рисунка вычисляются исходя из разрешения 72 dpi (точки на дюйм), т.е. указание размера в 1 дюйм (2.54 см) означает, что рисунок имеет 72 пикселя по соответствующему направлению.

PDFTeX умеет вычислять размеры рисунков, поэтому для этого компилятора указывать размеры необязательно.

Таблицы в документ включаются аналогично рисункам с помощью окружения `table`. При этом следует иметь в виду, что если подпись к рисункам размещается после рисунка, то у таблицы название, оно размещается перед таблицей, например:

```
\begin{table}[ht]
  \caption{Таблица с данными}\label{tab1}
  \begin{tabular}{|c|c|c|}
    ...
  \end{tabular}
\end{table}
```

6. Дополнительные возможности

Кавычки «лапки» (как в этой фразе) в текст можно вставлять либо командами `\lk` и `\rk`, либо командами `\al` и `\ar`. После этих команд желательно ставить пустые фигурные скобки, например: `\lk{}лапки\rk{}`. Это позволяет отделить команду левой кавычки от идущего далее слова, а за командой правой кавычки сохранить пробел (вспомним, что после команды TeX пробелы поглощаются).

Для использования в тексте полужирного курсива можно использовать комбинацию `\bfseries\itshape` (соответствующий фрагмент надо заключать в фигурные скобки). Стиль предоставляет команду `\bi` как эквивалент указанной комбинации. При этом команда `\bi` может использоваться и в формулах, но при этом работает не как переключатель, а как команда с параметром.

В стилевом файле реализован механизм набора текста «в разрядку». По команде `\looser{⟨число⟩}{⟨текст⟩}` в `⟨текст⟩` после каждого символа вставляется пробел величиной `⟨число⟩em`. Во втором параметре можно использовать фигурные скобки, тогда фрагмент в фигурных скобках воспринимается как один символ (фигурные скобки защищают от вставки такого пробела). Особенностью команды также является работа с пробелами. При использовании `\looser` в параметрах команд (например в заголовках) пробелы следует защищать фигурными скобками, например: `\section{\looser{0.2}{Мой{}заголовок}}`. Важной возможностью команды `\looser` является небольшое растяжение или сжатие текста при его укладке в строку. Коэффициент растяжения или сжатия (первый параметр команды) при этом не должен превышать 0.01, чтобы такое растяжение/сжатие не было заметно визуально.

В стилевом файле есть также команда `\mspsh`, которая регулирует пробелы в формулах. Пробелы в формулах имеют растяжимость и сжимаемость (в терминологии TeX это клей). Если внутри «долларов» задать команду `\mspsh{Ч1}{Ч2}{Ч3}`, где Ч1, Ч2, Ч3 — числа (коэффициенты), то в формуле пробелы вокруг отношений (типа равенства), операций, после запятых скорректируются: нормальный пробел умножится на Ч1, растяжимость станет равной нормальному (прежнему) пробелу, умноженному на Ч2, а сжимаемость — на Ч3. Например, после команды `\mspsh{0.7}{0.3}{0.4}` каждый пробел в формуле станет равным $0.7x$ plus $0.3x$ minus $0.4x$, где x — величина исходного нормального значения пробела. Команда `\msps{⟨число⟩}` есть эквивалент команды `\mspsh{⟨число⟩}{0}{0}`.

Как известно, TeX автоматически расставляет переносы в словах. Однако если в слове имеется дефис (или другие специальные символы), в этом слове механизм автоматического переноса отключается. Можно либо вручную указать точки возможного разрыва слова с помощью команды `\-`, либо вставить перед дефисом одну из команд `\nf` и `\nb`. В первом случае разрыв слова на дефисе допускается, во втором нет (как в конструкции ε -окрестность).

В списковых конструкциях (окружения `itemize` и `enumerate` довольно трудно регулировать формат (в первую очередь пробелы перед списком и между элементами). Эти пробелы устанавливаются параметрами `\topsep`, `\itemsep`, `\parsep` (соответственно пробел перед списком, пробел между элементами списка, пробел между абзацами в списке). Устанавливать их перед окружением нет смысла, поскольку в преамбуле окружения они переопределяются, внутри окружения они тоже не действуют, поскольку используются только в преамбуле для регулировки ряда внутренних параметров низкого уровня. Выход из ситуации обеспечивает команда `\userpar`. Например команда

```
\userpar{\topsep=0pt\itemsep=0pt\parsep=0pt}
```

снимает в окружении все вертикальные пробелы. Подобная команда кодируется перед окружением (вне его), причем она действует только на одно окружение, поскольку в конце окружения ее действие аннулируется.