

Оформление отчета о НИР в LaTeX

Date Сб 09 Ноябрь 2013 Редакция Вт 23 Август
2016 Категория comp Теги TeX / LaTeX / отчет о НИР / формулы / таблицы / графика

- Стандарт
- Приступим
- Шрифт
- Размер страницы
- Поля
- Междустрочный интервал
- Абзацный отступ
- Нумерация страниц
- Заголовки
- Содержание
- Математика
- Рисунки и таблицы
- Списки
- Сноски
- Листинги
- Приложения
- Переносы
- Полезные мелочи
- Ссылки

Оформление отчетов о научно-исследовательской работе (НИР) регламентируется стандартами ГОСТ 7.32-2001 (Россия, Беларусь) и ДСТУ 3008-95 (Украина). Далее мы будем придерживаться требований ДСТУ, но большинство рассматриваемых приемов подойдет и для ГОСТ, возможно, после небольшой переработки.

Стандарты оформления отчетов используются при подготовке многих других видов документов, в частности, курсовых и дипломных работ. К тому же, толкования требований стандарта в разных организациях различаются. Поэтому мы стремились к тому, чтобы создать некий "конструктор", из которого пользователь с минимальными знаниями LaTeX сможет выделить необходимые ему элементы оформления.

Стандарт

Тексты ДСТУ 3008-95 из Интернет отличаются деталями, в частности, в примерах оформления списков. Поэтому мы используем скан бумажной версии стандарта.

Оформление текста стандарта иногда ошибочно принимают за образец оформления отчета. В действительности, стандарт и отчет — разные виды документов, и образцами могут служить только примеры из текста стандарта.

Приступим

Предполагается, что читатель:

- установил дистрибутив TeX (например, TeXLive);
- представляет себе, что такое класс, пакет, преамбула документа, окружение — т.е., знаком с LaTeXом в объеме первого дня из книги "Сверстай диплом красиво: LaTeX за три дня";
- использует pdfLaTeX (хотя, после небольшой переделки, все это можно использовать и в XeTeX).

Запишем простейший документ LaTeX:

```
\documentclass{extreport}
\usepackage[utf8]{inputenc}
\usepackage[T2A]{fontenc}
\usepackage[russian]{babel}

\begin{document}

Текст.

\end{document}
```

В качестве кодировки документа выбрана UTF8. Указать другую кодировку можно с помощью опций `cp1251`, `koi8-r` или `cp866`. Основной язык документа — русский. Можно задать список используемых в документе языков, например: `english`, `ukrainian`, `russian`.

Шрифт

Стандарт требует, чтобы шрифт был высотой не менее 1.8 мм. По сложившейся традиции текст отчета набирается шрифтом размера 14 пунктов. Укажем этот размер при помощи опции `14pt` класса `extreport`

```
\documentclass[14pt]{extreport}
```

Класс документа `extreport` ("расширенный" отчет) выбран в качестве базового как раз потому, что поддерживает 14-пунктовый основной шрифт, чего нет у "обычного" отчета (класса `report`).

Размер страницы

Стандарт предполагает оформление отчета на листах формата A4. Добавим соответствующую опцию

```
\documentclass[a4paper, 14pt]{extreport}
```

Поля

Величина полей страницы должна составлять: верхнего, левого и нижнего — не менее 20 мм, правого — не менее 10 мм. Чтобы обеспечить нужные размеры, воспользуемся пакетом **geometry**:

```
% поля:
\usepackage[left=2.5cm, right=1.5cm, top=2.5cm, bottom=2.5cm]{geometry}
```

В нашем случае отступ от правого края страницы равен 15 мм, от остальных — 25 мм. Параметр `hmargin` пакета `geometry` задает размеры горизонтальных полей, а `vmargin` — вертикальных. Так что в нашем случае удобнее записать:

% поля:

```
\usepackage[left=2.5cm, right=1.5cm, vmargin=2.5cm]{geometry}
```

Междустрочный интервал

Число строк на странице не должно превышать 40. Эксперименты с текстом показали, что при указанных выше отступах, требования стандарта удовлетворяются при одинарном машинописном интервале.

Самостоятельно проверить это можно с помощью пакета `lipsum`, вставляющего в документ бессмысленный текст (lorem ipsum...) для заполнения места.

Изменить междустрочный интервал можно командой `\linespread`. Так, `\linespread{1.25}` соответствует полуторному интервалу.

Абзацный отступ

По умолчанию TeX делает первую строку текста, следующую после заголовка структурного элемента, лишенной абзацного отступа. Таковы британские полиграфические традиции. Изменить эту установку поможет пакет `indentfirst`

```
\usepackage{indentfirst} % отделять первую строку раздела абзацным отступом
```

Стандарт требует, чтобы величина абзацного отступа была равна пяти знакам, не уточняя, что это за знаки. Сделаем абзацный отступ равным по ширине пяти строчным буквам основного шрифта

```
\setlength\parindent{5ex}
```

`ex` — единица измерения, равная ширине латинской буквы 'x' шрифта по умолчанию (Computer Modern Roman, 14 пт).

Нумерация страниц

Номер страницы отчета проставляется арабскими цифрами в правом верхнем углу страницы. В классе `extreport` номера ставятся внизу и по центру страницы. Исправим это, изменив с помощью пакета `fancyhdr` стиль колонтитула:

```
\usepackage{fancyhdr}

\pagestyle{fancy}
\fancyhf{}
\rhead{\thepage}
\renewcommand{\headrulewidth}{0pt}
```

Здесь мы: очищаем колонтитулы (иначе номер страницы продолжит отображаться внизу), указываем номер страницы `\thepage` в правой части верхнего колонтитула (`\rhead`) и убираем горизонтальную линию, разделяющую колонтитул и основной текст (делаем ее толщину равной нулю).

Страницы с заголовками раздела `chapter` имеют отдельный стиль — `plain`, который настраивается аналогично

```
\fancyheadstyle{plain}{  
  \fancyhf{}  
  \rhead{\thepage}}
```

Заголовки

Заголовки разделов выполняются прописными буквами по центру, а заголовки пунктов и подпунктов — с абзачного отступа строчными буквами, начиная с первой прописной. Точки после номеров структурных элементов не ставятся.

Для форматирования заголовков используем пакет `titlesec`. Оформление заголовка настраивается командой `\titleformat`. У нее есть следующие параметры:

- первый — уровень настраиваемого заголовка (например, `chapter` или `section`);
- второй (необязательный) — форма заголовка. Описывает глобальный стиль его размещения — будет ли он «висеть» по центру, печататься перед текстом, входить в текст, располагаться на полях, или еще как-то иначе;
- третий параметр — команды, вызывающиеся перед печатью всего заголовка;
- четвертый параметр — оформление метки;
- пятый параметр — расстояние между меткой и текстом заголовка (горизонтальное или вертикальное, в зависимости от формы);
- шестой параметр — команды, вызывающиеся перед печатью текста заголовка;
- седьмой — команды, вызывающиеся после печати текста заголовка.

Еще одна необходимая команда — `\titlespacing*` — устанавливает отступы слева, перед, после и справа от заголовка.

Настроим заголовок раздела (`chapter`):

```
\usepackage{titlesec}  
  
\titleformat{\chapter}[block]  
  {\filcenter}  
  {\thechapter}  
  {1em}  
  {\MakeUppercase}}  
  
\titlespacing*{\chapter}{0pt}<s>30pt{*4}
```

Расстояние между меткой и текстом заголовка задаем в единицах шрифта: `em` — ширина буквы 'М' основного шрифта документа. `'*n'` в `\titlespacing` обозначает расстояние в `n` букв 'x' по высоте или ширине.

Настройка заголовка пункта (`section`):

```
\titleformat{\section}
{}
{\thesection}
{1ex}{}

\titlespacing*{\section}{\parindent}{*4}{*4}
```

В результате получим заголовок пункта, начинающийся с абзацного отступа и содержащий висящую метку (номер пункта)

1.1 Заголовок пункта. Текст текст текст текст текст текст текст текст текст текст текст текст текст текст текст текст текст текст

Если нужно, чтобы заголовок имел вид обычного абзаца, запишем его стиль так

```
\titleformat{\section}[block]
{\hspace{\parindent}}
{\thesection}
{1ex}{}

\titlespacing*{\section}{0pt}{*4}{*4}
```

1.1 Заголовок пункта. Текст текст текст текст текст текст текст текст текст текст текст текст текст текст текст текст текст текст

Заголовки подпунктов оформляются аналогично.

Кроме того, могут понадобиться заголовки 4-го уровня. Их можно печатать прямо в тексте и не помещать в оглавление. В начале нужно "скомандовать" классу `extreport` нумеровать заголовки 4-го уровня (по умолчанию нумеруются только заголовки первых трех уровней)

```
\setcounter{secnumdepth}{4}
```

Опция `runin` указывает, что заголовок будет помещаться в основном тексте.

```
\titleformat{\subsubsection}[runin]
{}
{\thesubsubsection}
{1ex}{}[.]

\titlespacing*{\subsubsection}{\parindent}{*4}{1ex}
```

Вот что получается в результате

2.1.1.1 Скорость сближения. Скорость сближения (closing velocity) \vec{v}_c двух объектов можно найти, проектируя скорости движения каждого объекта (\vec{v}_a и \vec{v}_b) на направление между этими объектами $\vec{x}_b - \vec{x}_a$ (рис. 2.2)

Нам понадобятся также заголовки без нумерации. Заголовки разделов, которые не помещают в содержание отчета (например, реферат), создаются с помощью команды `\chapter*`. Для остальных, в частности для введения и выводов, напишем макрос, который добавляет к `\chapter*` возможность помещать текст заголовка в содержание

```
\newcommand\chap[1]{%
  \chapter*{#1}%
  \addcontentsline{toc}{chapter}{#1}}
```

Содержание

Содержание по умолчанию называется "Оглавление". Исправим это

```
\addto{\captionsrussian}{\renewcommand*\contentsname{Содержание}}
```

Перед текстом заголовка главы в содержании делается отступ. Стандарт никак это не регламентирует, но традиции оформления отчетов в конкретной организации могут этого не допускать. Отменяем этот отступ с помощью пакета **titletoc**:

```
\usepackage{titletoc}

\dottedcontents{chapter}[1.6em]{}{1.6em}{1pc}
```

Строки заголовков в содержании превращаются в гиперссылки после подключения пакета **hyperref**:

```
\usepackage[hidelinks]{hyperref} % гиперссылки в содержании
```

Опция **hidelinks** нужна, чтобы скрыть рамку, окружающую гиперссылку.

Математика

Обычно с отображением формул TeX замечательно справляется безо всякой настройки. Однако, даже здесь могут возникнуть "хитрые" ситуации. Например, система уравнений

```
\begin{equation}
\begin{array}{lcl}
x &=& \frac{1}{2}y, \\
y &=& \frac{x}{z}, \\
z &=& \frac{3}{y}x.
\end{array}
\end{equation}
```

отображается как

$$\begin{array}{lcl} x & = & \frac{1}{2}y, \\ y & = & \frac{x}{z}, \\ z & = & \frac{3}{y}x. \end{array} \quad (1)$$

Дроби явно маловаты. Чтобы решить эту проблему, подключим пакеты от *American Mathematical Society*, содержащие многочисленные настройки и символы, используемые при оформлении математических текстов

```
\usepackage{amsmath,amssymb,amsfonts,amsthm} % математические дополнения от A
```

С помощью `\dfrac` из **amsmath** можно увеличить цифры в дробях

$$\begin{aligned} x &= \frac{1}{2}y, \\ y &= \frac{x}{z}, \\ z &= \frac{3}{y}x. \end{aligned} \tag{1}$$

Пространство по вертикали между уравнениями изменяется командой `\arraystretch`

```
\begin{equation}
\renewcommand{\arraystretch}{2.}
\begin{array}
...
\end{array}
\end{equation}
```

$$\begin{aligned} x &= \frac{1}{2}y, \\ y &= \frac{x}{z}, \\ z &= \frac{3}{y}x. \end{aligned} \tag{1}$$

Будучи помещенной в окружение (например, в `equation`), `\arraystretch` действует только на данный объект (систему уравнений). Если же поместить команду в преамбулу, то она будет действовать глобально — на весь документ. С помощью `\arraystretch` можно также регулировать пространство по вертикали в массивах и таблицах.

Вместо `array` для записи систем уравнений можно воспользоваться окружением `aligned` из пакета **amsmath**

```
\begin{equation}
\begin{aligned}
x &= \frac{1}{2}y, \ \backslash\backslash \\
y &= \frac{x}{z}, \ \backslash\backslash \\
z &= \frac{3}{y}x. \ \backslash\backslash
\end{aligned}
\end{equation}
```

Отрегулировать пробел до и после выключной формулы можно, переопределив окружение `equation`

```
\let\oldequation=\equation
\let\endoldequation=\endequation
\renewenvironment{equation}{\vspace{3pt}\begin{oldequation}}{\end{oldequation}}
```

В данном случае мы добавляем по 3 пункта к пробелам, которыми TeX по умолчанию окружает формулу.

Кроме того, подключим пакет **icomma**, позволяющий использовать запятую в качестве разделителя целой и дробной частей числа в формулах (в англоязычной литературе таким разделителем является точка):

```
\usepackage{icomma}
```

Рисунки и таблицы

Для размещения в тексте рисунков и таблиц обычно используются окружения `figure` и `table`.

Чтобы настроить формат подписей подключим пакеты **caption** и **subcaption**. Последний нужен для создания рисунков, состоящих из нескольких взаимосвязанных частей.

```
\usepackage[tableposition=top,singlelinecheck=false]{caption}
\usepackage{subcaption}

\DeclareCaptionLabelFormat{gostfigure}{Рисунок #2}
\DeclareCaptionLabelFormat{gosttable}{Таблица #2}
\DeclareCaptionLabelSeparator{gost}{~---~}
\captionsetup{labelsep=gost}
\captionsetup*[figure]{labelformat=gostfigure}
\captionsetup*[table]{labelformat=gosttable}
\renewcommand{\thesubfigure}{\asbuk{subfigure}}
```

Опция `tableposition` пакета **caption** управляет размещением заголовка относительно таблицы. В данном случае заголовок будет расположен над таблицей.

Однострочный заголовок отображается иначе, чем многострочный. Первый обычно размещается по центру, второй — как обычный абзац. Чтобы отменить проверку на "однострочность" и отображать все заголовки по правилам, действующим для многострочных заголовков, устанавливаем опцию `singlelinecheck=false`.

С помощью команды `\DeclareCaptionLabelFormat` настраивается оформление подписи, которое затем будет использоваться для всех рисунков и таблиц документа. Первый параметр (`#1`) этой команды, определяющий стандартный текст метки, мы не используем; параметр `#2` задает номер рисунка/таблицы.

`\DeclareCaptionLabelSeparator` настраивает разделитель между меткой и текстом подписи. В нашем случае это тире.

Команда `\captionsetup` устанавливает настройки оформления подписей к объектам (рисункам, таблицам):

```
\captionsetup[имя_объекта]{настройки}
```


Настройки могут относиться ко всем объектам (в этом случае квадратные скобки не ставятся), к отдельным типам объектов (`figure`, `table`), а также к разновидностям объекта (например, к продолжению таблицы, начатой на предыдущей странице).

Обычный `\captionsetup` выдает предупреждение, если стиль для какого-то типа объектов не использовался в документе. Например, настройки `\captionsetup[table]` заданы, а ни одного окружения `table` в документе нет. Существует «звездный» вариант команды — `\captionsetup*` — который этого не делает, поэтому мы используем именно его.

Отдельно стоит прокомментировать последнюю строку. Части составного рисунка, созданные с помощью `subcaption`, требуется нумеровать строчными буквами кириллицы. Кириллическое представление счетчиков (по аналогии с известными представлениями `\arabic`, `\roman`, `\latin` и др.) реализовано в пакете `babel` и называется `\asbuk` — для строчных букв и `\Asbuk` — для прописных.

Пример таблицы:

```
\begin{table}[h!]
  \caption{Первые искусственные спутники Земли}
  \centering
  \begin{tabular}{l | l | l |}
    \hline
    ИСЗ & Дата запуска & Масса, кг \\ \hline
    Спутник-1 & 4 октября 1957 & 83,6 \\ \hline
    Спутник-2 & 3 ноября 1957 & 508,3 \\ \hline
    Эксплорер-1 & 1 февраля 1958 & 21,5 \\ \hline
  \end{tabular}
\end{table}
```

Таблица 1.1 — Первые искусственные спутники Земли

ИСЗ	Дата запуска	Масса, кг
Спутник-1	4 октября 1957	83,6
Спутник-2	3 ноября 1957	508,3
Эксплорер-1	1 февраля 1958	21,5

Увеличим вертикальный просвет между строками таблицы с помощью

```
\renewcommand{\arraystretch}{1.3}
```

Таблица 1.1 — Первые искусственные спутники Земли

ИСЗ	Дата запуска	Масса, кг
Спутник-1	4 октября 1957	83,6
Спутник-2	3 ноября 1957	508,3
Эксплорер-1	1 февраля 1958	21,5

Согласно стандарту, горизонтальные и вертикальные линии, разграничивающие строки таблицы, а также линии, ограничивающие таблицу слева, справа и снизу, можно не проводить, если их отсутствие не затрудняет пользование таблицей. Поэтому может оказаться полезным пакет **booktabs**, который как раз и предполагает подобный подход к оформлению таблиц.

Стоит отметить использование команд `\multirow` и `\multicolumn` для создания сложных таблиц. Они соответствуют параметрам HTML-тега `<table>` `rowspan` и `colspan`. Для использования `\multirow` необходимо подключить пакет **multirow**

```
\usepackage{multirow} % улучшенное форматирование таблиц
```

```
\begin{table}[ht]
\caption{Пример таблицы с более сложным форматированием}
\label{tab_weight}
\centering
\begin{tabular}{|c|c|c|c|c|c|c|c|c|}
\hline \multirow{2}{*}{Параметр  $x_i$ } & \multicolumn{4}{c|}{Параметр  $x_i$ } & \multicolumn{4}{c|}{Параметр  $x_i$ } \\
\multicolumn{2}{c|}{Первый шаг} & \multicolumn{2}{c|}{Второй шаг} & \multicolumn{2}{c|}{Третий шаг} & \multicolumn{2}{c|}{Четвертый шаг} & \\
\cline{2-9} &  $x_1$  &  $x_2$  &  $x_3$  &  $x_4$  &  $w_i$  &  $K_{\text{B}}$  &  $w_i$  &  $K_{\text{B}}$  \\
\hline  $x_1$  & 1 & 1 & 1.5 & 1.5 & 5 & 0.31 & 19 & 0.32 \\
\hline  $x_2$  & 1 & 1 & 1.5 & 1.5 & 5 & 0.31 & 19 & 0.32 \\
\hline  $x_3$  & 0.5 & 0.5 & 1 & 0.5 & 2.5 & 0.16 & 9.25 & 0.16 \\
\hline  $x_4$  & 0.5 & 0.5 & 1.5 & 1 & 3.5 & 0.22 & 12.25 & 0.20 \\
\hline \multicolumn{5}{c|}{Итого:} & 16 & 1 & 59.5 & 1 \\
\hline
\end{tabular}
\end{table}
```

Таблица 1.1 — Пример таблицы с более сложным форматированием

Параметр x_i	Параметр x_j				Первый шаг		Второй шаг	
	X_1	X_2	X_3	X_4	w_i	$K_{\text{в}i}$	w_i	$K_{\text{в}i}$
X_1	1	1	1.5	1.5	5	0.31	19	0.32
X_2	1	1	1.5	1.5	5	0.31	19	0.32
X_3	0.5	0.5	1	0.5	2.5	0.16	9.25	0.16
X_4	0.5	0.5	1.5	1	3.5	0.22	12.25	0.20
Итого:					16	1	59.5	1

Многострочный заголовок таблицы отображается так

Таблица 1.1 — Текст текст текст текст текст текст текст текст текст текст
текст текст текст текст текст текст текст текст

Его можно оформить с висящей меткой

```
\captionsetup*[table]{labelformat=gosttable, format=hang} % с висящей меткой
```

Таблица 1.1 — Текст текст текст текст текст текст текст текст текст текст
текст текст текст текст текст текст текст текст

или с абзацным отступом

```
\captionsetup*[table]{labelformat=gosttable, margin={5ex,0pt}, indentation=-5ex}
```

Таблица 1.1 — Текст текст текст текст текст текст текст текст текст
текст текст текст текст текст текст текст текст текст текст

Рисунки вставляются в документ с помощью окружения `figure`:

```

\begin{figure}[ht]
\centering
В окружение \texttt{figure} можно помещать не только рисунки.

$$\begin{array}{l}
A \rightarrow T \\
B \rightarrow A \\
T \rightarrow B
\end{array}$$

\caption{Это не формула, а рисунок}
\label{fig:sample}
\end{figure}

```

В окружение `figure` можно помещать не только рисунки.

$$\begin{array}{l}
A \rightarrow T \\
B \rightarrow A \\
T \rightarrow B
\end{array}$$

Рисунок 1.1 — Это не формула, а рисунок

```

\begin{figure}[h]
\centering
\includegraphics[scale=0.75]{chick.png}
\caption{Тестовый рисунок <<Цыпа>>}
\label{fig:image}
\end{figure}

```

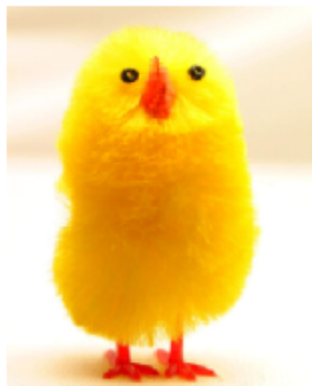


Рисунок 1.1 — Тестовый рисунок «Цыпа»

Заголовки рисунков и таблиц настраиваются аналогично

```

\captionsetup*[figure]{labelformat=gostfigure, justification=centering} % Вь

```

При размещении рисунков и таблиц возникает следующая проблема: стандарт требует, чтобы они помещались непосредственно после упоминания в тексте или на следующей странице, тогда как TeX стремится обеспечить максимальное полиграфическое качество документа. В результате рисунок (таблица) может появиться в документе через несколько страниц после упоминания, если с точки зрения качества заполнения страницы так будет лучше. Все дело в том, что окружения `figure` и `table` — "плавающие", т.е. меняющие свое положение в документе. Стандарт же, по сути, требует зафиксировать расположение рисунка (таблицы).

Добиться этого можно, отказавшись от использования `figure` и `table`. Заголовки рисунков (таблиц) создаются командой `\captionof` из пакета `caption`

```
\usepackage{caption}
...
\begin{center}
\includegraphics{imagefile}
\captionof{figure}{Заголовок}
\end{center}
```

В этом случае желательно помочь TeX'у, и поместить рисунок вблизи от того места документа, где он должен появиться. Напоминает работу в Word'e, не правда ли?

Рисунки можно размещать на отдельных страницах. Для этого в преамбуле документа следует записать:

```
\renewcommand\floatpagefraction{0.01}

\makeatletter
\renewcommand*\fps@figure{p}
\@fpsep\textheight
\makeatother
```

Вставлять такие рисунки нужно с помощью окружения `figure` без опций, т.е. без `[htbp]` или чего-то подобного.

При этом рисунки могут быть вставлены в документ в конце главы, в десятках страниц от места их упоминания в тексте (проблема "плавучести" `figure` никуда не исчезла). Исправить положение можно, уменьшая параметр `\floatpagefraction`. Установив расстояние между плавающими объектами `\@fpsep` (а рисунок `figure` — это плавающий объект) большим или равным высоте текста `\textheight`, мы добьемся того, что на странице не появится другой рисунок.

`\makeatletter` и `\makeatother` нужны для работы со специальным символом '@' (at). Первая превращает его в обычную букву, после чего можно переопределять команды вроде `\@fpsep`. Вторая вновь превращает '@' в спецсимвол.

Длинные таблицы, занимающие несколько страниц, удобно создавать с помощью пакета `longtable`.

Можно разбивать длинные таблицы вручную, оформляя каждую как `table`. В этом случае для продолжений таблицы нужно создать отдельный стиль заголовка

```
\DeclareCaptionLabelFormat{continued}{Продолжение таблицы~#2}
\captionsetup*[ContinuedFloat]{labelformat=continued}
```

Следующий пример показывает, как реализовать этот подход:

```

\begin{table}
\caption{Заголовок таблицы}
\centering ТАБЛИЦА
\end{table}

\begin{table}
\ContinuedFloat
\caption{}
\centering ТАБЛИЦА
\end{table}

```

Таблица 1 — Заголовок таблицы

ТАБЛИЦА

Продолжение таблицы 1

ТАБЛИЦА

Пакет **threeparttable** поможет при создании примечаний и сносок к таблицам. Пакет **makecell** позволит создать уникальное оформление ячеек.

Списки

Требования к спискам в стандарте следующие:

- нумерованные списки на первом уровне помечаются как 'а)', 'б)', 'в)'... На втором — как '1)', '2)', '3)';
- нумерованные списки помечаются дефисами.

Установим метки пунктов в соответствии с этими требованиями:

```
% маркированные списки
\renewcommand{\labelitemi}{</s>}
\renewcommand{\labelitemii}{<s>}

% нумерованные списки
\renewcommand{\labelenumi}{\asbuk{enumi}}
\renewcommand{\labelenumii}{\arabic{enumii}}
```

Согласно стандарту, списки первого уровня детализации должны начинаться с абзацного отступа, второго и последующих уровней — с отступа относительно предыдущего уровня. Величина этого последнего отступа в стандарте не задана.

LaTeX делает в списках `itemize` и `enumerate` специальный отступ по вертикали между пунктами списка и между пунктом списка и соседним абзацем. Чтобы убрать эти отступы, воспользуемся пакетом `enumitem` с опцией `nosep`, отключающей все дополнительные отступы. Однако `enumitem` ничего не "знает" о счетчиках `asbuk`, так что нужно зарегистрировать их как новые счетчики командой `\AddEnumerateCounter`. Третий параметр этой команды — пример самой широкой метки (пакет будет ориентироваться на эту ширину при печати).

```
\usepackage{enumitem}
\makeatletter
\AddEnumerateCounter{\asbuk}{\@asbuk}{ю}}
\makeatother
\setlist{nosep, leftmargin=\parindent}
```

Опция `leftmargin` устанавливает левую границу текста списка равной отступу в начале абзаца.

Рассмотрим следующий пример:

```
Это текст абзаца, который не вмещается в одну строку, поэтому он располагается
\begin{itemize}
\item Это ...
\begin{itemize}
\item Это ...
\item Это ...
\end{itemize}
\item Это ...
\end{itemize}
```

Каждый пункт списка содержит текст, задача которого — занимать, по крайней мере, две строки. Сделаем аналогично еще два списка, вложив нумерованный список внутрь маркированного, и нумерованные списки — друг в друга.

С указанными выше настройками списков, получим:

Это текст абзаца, который не вмещается в одну строку, поэтому он располагается на нескольких строках.

- Это текст абзаца, который не вмещается в одну строку, поэтому он располагается на нескольких строках.
 - Это текст абзаца, который не вмещается в одну строку, поэтому он располагается на нескольких строках.
 - Это текст абзаца, который не вмещается в одну строку, поэтому он располагается на нескольких строках.
- Это текст абзаца, который не вмещается в одну строку, поэтому он располагается на нескольких строках.

Это текст абзаца, который не вмещается в одну строку, поэтому он располагается на нескольких строках.

- Это текст абзаца, который не вмещается в одну строку, поэтому он располагается на нескольких строках.
 - а) Это текст абзаца, который не вмещается в одну строку, поэтому он располагается на нескольких строках.
 - б) Это текст абзаца, который не вмещается в одну строку, поэтому он располагается на нескольких строках.
- Это текст абзаца, который не вмещается в одну строку, поэтому он располагается на нескольких строках.

Это текст абзаца, который не вмещается в одну строку, поэтому он располагается на нескольких строках.

- а) Это текст абзаца, который не вмещается в одну строку, поэтому он располагается на нескольких строках.
 - 1) Это текст абзаца, который не вмещается в одну строку, поэтому он располагается на нескольких строках.
 - 2) Это текст абзаца, который не вмещается в одну строку, поэтому он располагается на нескольких строках.
- б) Это текст абзаца, который не вмещается в одну строку, поэтому он располагается на нескольких строках.

Иногда требуется выравнивание не по тексту пункта, а по метке. При этом пункт списка должен вести себя как обычный абзац (не выравниваться по первой строке, а тянуться вплоть до левой границы текста). Вот что имеется в виду:

В этом списке пункты выравниваются по первой строке.

- Это текст пункта, который не вмещается в одну строку, поэтому он располагается на нескольких строках.
- Это текст пункта, который не вмещается в одну строку, поэтому он располагается на нескольких строках.

Здесь пункты ведут себя как обычные абзацы.

- Это текст пункта, который не вмещается в одну строку, поэтому он располагается на нескольких строках.
- Это текст пункта, который не вмещается в одну строку, поэтому он располагается на нескольких строках.

Сделать это можно с помощью опции **wide** :

```
% убираем дополнительные отступы
% и настраиваем пункты для всего списка
\setlist{nosep,wide}
% для пунктов 2-го уровня
% отступ метки от края равен двум абзацным отступам
\setlist[2]{labelindent=2\parindent}
```

Кроме того, мне не нравится использование дефисов в качестве маркеров, и я заменяю их на "минус":

```
\renewcommand{\labelitemi}{-</s>
\renewcommand{\labelitemii}{--}
```

В сумме эти изменения дают следующий результат:

Это текст абзаца, который не вмещается в одну строку, поэтому он располагается на нескольких строках.

– Это текст абзаца, который не вмещается в одну строку, поэтому он располагается на нескольких строках.

– Это текст абзаца, который не вмещается в одну строку, поэтому он располагается на нескольких строках.

– Это текст абзаца, который не вмещается в одну строку, поэтому он располагается на нескольких строках.

– Это текст абзаца, который не вмещается в одну строку, поэтому он располагается на нескольких строках.

Это текст абзаца, который не вмещается в одну строку, поэтому он располагается на нескольких строках.

– Это текст абзаца, который не вмещается в одну строку, поэтому он располагается на нескольких строках.

а) Это текст абзаца, который не вмещается в одну строку, поэтому он располагается на нескольких строках.

б) Это текст абзаца, который не вмещается в одну строку, поэтому он располагается на нескольких строках.

– Это текст абзаца, который не вмещается в одну строку, поэтому он располагается на нескольких строках.

Это текст абзаца, который не вмещается в одну строку, поэтому он располагается на нескольких строках.

а) Это текст абзаца, который не вмещается в одну строку, поэтому он располагается на нескольких строках.

1) Это текст абзаца, который не вмещается в одну строку, поэтому он располагается на нескольких строках.

2) Это текст абзаца, который не вмещается в одну строку, поэтому он располагается на нескольких строках.

б) Это текст абзаца, который не вмещается в одну строку, поэтому он располагается на нескольких строках.

Сноски

Сноски обозначаются надстрочными знаками в виде арабских цифр со скобкой

% номер сноски со скобкой

```
\renewcommand*{\thefootnote}{\arabic{footnote}}}
```

Текст сноски должен отделяться от основного текста линией шириной 30—40 мм. Ширина (width) и высота (height) этой линии (или, если угодно, её длина и толщина) задаются командой `\footnoterule`. Установим ширину линии сноски равной 40 мм:

```
\renewcommand{\footnoterule}{%  
  \kern -3pt  
  \hrule width 40mm height .4pt  
  \kern 2.6pt  
}
```

Листинги

Листинги программ оформляются с помощью пакета **minted**

```
\usepackage{minted}
```

Для использования **minted** нужно вызывать **pdflatex** с опцией **-shell-escape**.

С настройками по умолчанию листинг программы на C++ выглядит следующим образом

```
int main(int argc, char** argv)  
{  
    glutInit(&argc, argv);  
    TimingData::init();  
  
    // #1: Инициализация и создание окна GLUT  
    app = getApplication();  
    createWindow(app->getTitle());  
  
    app->initGraphics();  
  
    // #2: Регистрация функций-обработчиков событий  
    glutReshapeFunc(reshape);  
    glutDisplayFunc(display);  
    glutIdleFunc(update);  
  
    // #3: Запуск основного цикла GLUT  
    glutMainLoop();  
  
    // Очистка ресурсов  
    TimingData::deinit();  
}
```

Стиль подсветки синтаксиса можно установить командой

```
\usemintedstyle{ИМЯ_СТИЛЯ}
```

Если документ предполагается печатать на черно-белом принтере, то подсветку синтаксиса можно отключить

```
\usemintedstyle{bw} % убираем подсветку синтаксиса в minted
```

Настройка размера шрифта и семейства шрифтов (по умолчанию это машинописные шрифты семейства `tt`, но могут потребовать и такое)

```
\begin{minted}[fontsize=\small, fontfamily=rm]{cpp}
...
\end{minted}
```

Чтобы получить тот же результат, не указывая опции в каждом окружении, можно написать в преамбуле

```
\newminted{cpp}[fontsize=\small, fontfamily=rm]
```

Тогда в тексте исходный код помещаем в окружение

```
\begin{cppcode}
...
\end{cppcode}
```

Обратите внимание: именно `cppcode`, а не `cpp`.

Листинги могут использоваться в качестве иллюстраций, и помещаться в окружение `figure`

```
\begin{figure}[h]
\centering
\begin{minted}{c++}
...
\end{minted}
\caption{Подпись к листингу}
\label{fig:listing}
\end{figure}
```

По умолчанию листинги выравниваются по левому краю, что для иллюстраций неприемлемо. Так происходит потому, что `minted` использует для оформления листингов окружение `Verbatim`. Заменяв `Verbatim` на окружение `BVerbatim` из пакета `fancyvrb`, добиваемся центрирования следующим образом:

```
\usepackage{fancyvrb}

\RecustomVerbatimEnvironment{Verbatim}{BVerbatim}{}

```

Приложения

Если приложения оформляется как продолжение отчета, то оно должно начинаться с новой страницы. Приложение должно иметь заголовок, напечатанный вверху строчными буквами, начиная с первой прописной, симметрично относительно текста страницы. Посередине строки над заголовком строчными буквами с первой прописной должно быть напечатано слово "Приложение" и большая буква, обозначающая приложение.

Для оформления приложений подключим пакет `appendix`

```
\usepackage[title,titletoc]{appendix}
```

Создадим стиль раздела приложения и назовем его `\append`

```
\newcommand{\empline}{\mbox{}\\} % пустая строка

\newcommand{\append}[1]{
  \clearpage
  \stepcounter{chapter}
  \begin{center}
    \chaptertitlename~\Asbuk{chapter}
  \end{center}
  \begin{center}{#1}\end{center}
  \empline
  \addcontentsline{toc}{chapter}{\Asbuk{chapter}\hspace{0.6em}~#1}}
```

Текст приложения включается в документ после библиографических ссылок при помощи окружения `appendices`

```
\begin{appendices}

\append{Заголовок приложения A}

Текст приложения.

\section{Заголовок пункта приложения}

Текст приложения.

\end{appendices}
```

Приложение А

Заголовок приложения А

Текст приложения.

А.1 Заголовок пункта приложения

Текст приложения.

Переносы

По стандарту, перенос слов не допускается в заголовках разделов.

Проще всего добиться этого, вставив перед потенциально переносимым словом разрыв строки `\linebreak` (этот вид разрыва сохраняет выравнивание строки по ширине). Побочным эффектом будет появление такого же разрыва в оглавлении документа, но его можно избежать, используя команду `\section` в форме

```
\section[Заголовок, помещаемый в оглавление]{Заголовок, помещаемый в основной текст}
```

Было:

0.1

Это наш первый документ LaTeX. В нем есть длинный-длинный-предлинный заголовок. При этом нужно еще сохранить выравнивание по ширине

Изменили заголовок:

```
\section[Это наш первый документ LaTeX. В нем есть длинный-длинный-предлинный заголовок]{Это наш первый документ LaTeX. В нем есть длинный-длинный-предлинный заголовок}
```

Получилось:

0.1

Это наш первый документ LaTeX. В нем есть длинный-длинный-предлинный заголовок. При этом нужно еще сохранить выравнивание по ширине

1

0.1

Это наш первый документ LaTeX. В нем есть длинный-длинный-предлинный заголовок. При этом нужно еще сохранить выравнивание по ширине

Другой вариант: переопределить заголовок с помощью пакета `titlesec` и включить в новое определение команду, запрещающую перенос.

Наименьшее количество букв в слове, которые можно переносить на следующую строку, задается параметром `\righthyphenmin`. Если сделать его достаточно большим, то TeX не сможет делать переносы, причем действовать этот запрет будет локально.

Пример определения заголовка:

```
\usepackage{titlesec}
\titleformat*{\section}{\sloppy\righthyphenmin62}
```

В нашем примере значение `\righthyphenmin` равно 62 буквам.

Иногда требование к переносам расширяют до полной отмены переносов в документе, сохраняя при этом выравнивание по ширине.

Полностью отключить переносы позволяет установка `\pretolerance=10000`. Однако при этом возникают незаполненные (underfull) или переполненные (overfull) строки. Поэтому нам понадобится задать еще два параметра: `\tolerance` и `\emergencystretch`.

`\tolerance` задает максимально допустимую меру разреженности строки. При разбиении абзаца на строки, TeX старается не создавать строк, мера разреженности которых больше, чем `\tolerance`. По умолчанию `\tolerance=200`. Нам придется ослабить требования к качеству строк — все равно они остаются весьма высокими.

Если при верстке абзаца не удастся избежать переполнения, то — при условии, что значение `\emergencystretch` отлично от нуля, — TeX делает еще одну попытку, при которой к «пределу растяжимости» каждой из строк прибавляется значение `\emergencystretch`.

Значения `\tolerance` и `\emergencystretch` подбираются экспериментально, например:

```
\pretolerance=10000
\tolerance=2000
\emergencystretch=10pt
```

Кроме того, полезно подключить пакет **`microtype`**. Он улучшает читаемость текста, управляя размерами пробелов между словами и прочими деликатными микротипографическими операциями. В результате использования **`microtype`** сокращается число переносов.

Полезные мелочи

По умолчанию, TeX следует английской полиграфической традиции, в которой после точки в конце предложения ставится двойной пробел. Отечественная полиграфия следует французской традиции одинарного пробела, поэтому в преамбуле следует указать команду `\frenchspacing`.

Чтобы иметь возможность искать и копировать из получившегося PDF-файла набранный кириллицей текст, используется пакет **`cmr`**.

```
\usepackage{cmr}
```

Важно: команда подключения **`cmr`** должна стоять в документе непосредственно после `\documentclass`.