

Семейство классов listok *

Илья Райко
rayko_i@179.ru

31 марта 2023 г.

Аннотация

Классы, используемый нашей командой для вёрстки листов.

1 Введение

На данный момент у нас есть классы listok, written, ustn, test. listok используется для вёрстки листов, written и ustn используются для вёрстки письменных и устных собеседований, test используется для контрольных работ.

Разница между ними описана в следующей табличке

	listok	test	written	ustn
Заголовок	\title{Title}			
	<u>Title</u>	<u>Title. Вариант №num</u>	<u>... собеседование</u>	
Задача	\begin{problem}[Text]			
	Задача num.problemText	Задача problemText		
Пример	\begin{example}[Text]			
	Пример num.exampleText	Пример exampleText		
Теорема	\begin{theorem}[Text]			
	Теорема num.theorem (Text)	Теорема theorem (Text)		

При этом у классов есть ещё некоторые индивидуальные особенности, которые будут описаны ниже.

2 Параметры классов

2.1 Формат бумаги

```
1 \common\RequirePackage{kvoptions}
2 \common\newcommand{\@paperfile}{}
3 \common\DeclareOption{a4paper}{\renewcommand{\@paperfile}{a4.clo}}
```

С a4paper всё просто — напечатает одну копию на листе А4, ориентированном вертикально. Ширина текста будет 179mm, а высота 267mm. Текст выровнен по центру (вертикально и горизонтально).

```
4 \*a4
5 \RequirePackage[a4paper, width = 179mm, height = 267mm, footskip = 0mm, centering, includeheadfoot]{geometry}
6 \PassOptionsToClass{a4paper, oneseide}{article}
7 \/a4
8 \common\DeclareOption{a5paper}{\renewcommand{\@paperfile}{a5.clo}}
```

Чуть сложнее с a5paper. Он нужен для того, чтобы напечатать два вертикальных А5 на одном горизонтальном А4. Признаться честно, я уже забыл, что делают конкретные команды, но это, кажется, работает.

```
9 \*a5
10 \setlength\paperheight{210mm}
11 \setlength\paperwidth{148mm}
```

*Этот документ относится к listok v2.3, от 2023/03/31

```

12 \PassOptionsToClass{a5paper,oneside}{article}
13 \RequirePackage{pgfpages}
14 \RequirePackage{atbegshi}
15 \setlength\hoffset{.4in}
16 \setlength\oddsidemargin{-1in}
17 \setlength\textwidth{130mm}
18 \setlength\voffset{-25mm}
19 \setlength\textheight{185mm}
20 \pgfpagesuselayout{2 on 1}[a4paper,landscape,border shrink=5mm]
21 \AtBeginShipout{%
22     \pgfpagesshipoutlogicalpage{1}\copy\AtBeginShipoutBox%
23     \pgfpagesshipoutlogicalpage{2}\box\AtBeginShipoutBox%
24     \pgfshipoutphysicalpage%
25 }
26 </a5>

```

2.2 Дата и номер листка

`date` задаёт дату листочка в углу. Сейчас значение по умолчанию — сегодня. Бейте меня палками, если до сих пор не взлетает.

```

27 <{*common}>
28 \DeclareStringOption[DTMToday]{date}

```

`num` — номер листочка. Влияет на номер задач.

```

29 \DeclareStringOption[1]{num}

```

`solutions` отвечает за вёрстку решений.

```

30 \DeclareBoolOption{solutions}

```

Все остальные опции передаются классу `article`.

```

31 \DeclareOption*{\PassOptionsToClass{\CurrentOption}{article}}
32 \ProcessOptions
33 \ProcessKeyvalOptions*
34 \LoadClass{article}
35 \input{\@paperfile}
36 \RequirePackage{iftex}

```

3 Движки

Вот тут начинается цирк с движками. Проблема заключается в том, что традиционный Web2C движок использует кодировки отличные от `utf8` (OT2, T2A, T2B, T2C, etc.) Чем это плохо? В OT2 был очень странный порядок сортировки кириллических букв (Э, Ю, Ж, Й, Ё, Я, А, Б, Ц, Д, Е, Ф и т.д.), что выливается в отвратительные списки и предметные указатели.

Даже игнорируя это факт, мы получаем две проблемы: старый Web2C движок генерирует странный слой текст в pdf (зюковки и кракозябры вместо русского текста), можно использовать только шрифты поддерживающие кодировки T2. Проблемы в том, что был только один такой шрифт, в котором русские буквы имеют равную высоту. Этот шрифт распространялся пакетом `rsrur`, который не обновляется с конца 2000-х годов. И тут нас постигает совсем проблема: `rsrur` сегодня поддерживается только дистрибутивом MiKTeX.

Поэтому предлагается использовать расширенный движок LuaTeX. Если не вдаваться в подробности, то имеет место простая формула:

$$\text{LuaTeX} = \text{pdfTeX} + \text{Поддержка Unicode и OpenType шрифтов} + \text{e-TeX} + \text{Omega} + \text{встроенный интерпретатор lua}$$

Корректная работа классов семейства `listok` гарантируется при использовании LuaTeX. В теории можно использовать и другие движки, но на свой страх и риск (на момент 19.01.2019 я не мог скомпилировать русский текст с pdfLaTeX).

Движок определяется автоматически по имени запускаемой программы.

```

37 \ifLuaTeX
38     \input{lua.clo}
39 </common>

```

```

40 <*lua>
41   \RequirePackage[TU]{fontenc}
42   \RequirePackage[lut8]{luainputenc}
43   \RequirePackage{fontspec}
44   \setmainfont{CMU Serif}
45 </lua>
46 <common>\else
47 <pdf>\RequirePackage[utf8]{inputenc}
48 <pdf>\RequirePackage{cmap}
49 <*common>
50   \input{pdf.clo}
51 \fi

```

4 Фишки

Далее идут подключённые пакеты

```

52 \RequirePackage[russian]{babel}
53 \RequirePackage[inline]{enumitem}
54 \RequirePackage{textcomp, multicol}
55 \RequirePackage{mathtext}
56 \RequirePackage[indentfirst}
57 \RequirePackage{amsmath, amssymb, amsfonts, amsthm}
58 \RequirePackage{mathtools, mathabx}
59 \RequirePackage{epstopdf}
60 \RequirePackage{graphicx}
61 \RequirePackage{forloop}
62 \RequirePackage{datetime2}
63 \RequirePackage{microtype}
64 \RequirePackage{comment}
65 \relax
66 \tolerance 4000

```

\listok@name задаёт имя листка, затем используется командой \maketitle для заголовка.

```

67 <listok>\newcommand{\listok@name}{Листок №\listok@num}
68 <test>\newcommand{\listok@name}{Вариант №\listok@num}
69 <written>\newcommand{\listok@name}{}
70 <ustn>\newcommand{\listok@name}{Фамилия, Имя}
71 \renewcommand{\maketitle}{%
72   \begin{center}%
73     {\Large \textit{\textbf{\underline{%
74 <listok>\@title % Лорен ипсум
75 <test>\@title. \listok@name % Контрольная по Лорен ипсум. Вариант №N
76 <written>Письменное собеседование
77 <ustn>Устное собеседование
78   }}}
79   \end{center}
80 }

81 \DTMsetup{datesep=.}
82 \DTMsetstyle{ddmmyyyy}
83 \renewcommand{\thefootnote}{\fnsymbol{footnote}}
84 \renewcommand{\@oddhead}{
85   \vbox{\hbox to\textwidth{\listok@name\hfil \strut
86     \listok@date
87   }}\hrule}
88 }
89 \renewcommand{\@oddfoot}{}

```

4.1 Транскодеанизация

Команда	Вывод
<code>\emptyset</code>	\emptyset
<code>\le</code>	\leq
<code>\ge</code>	\geq
<code>\epsilon</code>	ϵ
<code>\phi</code>	φ
<code>\N</code>	\mathbb{N}
<code>\Z</code>	\mathbb{Z}
<code>\Q</code>	\mathbb{Q}
<code>\R</code>	\mathbb{R}
<code>\C</code>	\mathbb{C}
<code>\Re z</code>	$\operatorname{Re} z$
<code>\Im z</code>	$\operatorname{Im} z$
<code>\Zm{p}</code>	\mathbb{Z}_p
<code>\Prob{Event}</code>	$\mathbb{P}[Event]$
<code>\ProbV{\xi}{Event}</code>	$\mathbb{P}_\xi[Event]$
<code>\Expect{\xi}</code>	$\mathbb{E}\xi$
<code>\pt{Text}</code>	$\operatorname{Text}^\circ$
<code>Text\point</code>	$\operatorname{Text}^\circ$
<code>Text\hard</code>	Text^*

```

90 \renewcommand{\emptyset}{\varnothing}
91 \renewcommand{\le}{\leqslant}
92 \renewcommand{\ge}{\geqslant}
93 \renewcommand{\epsilon}{\varepsilon}
94 \renewcommand{\phi}{\varphi}
95 \newcommand{\N}{\mathbb{N}}
96 \newcommand{\Z}{\mathbb{Z}}
97 \newcommand{\Q}{\mathbb{Q}}
98 \newcommand{\R}{\mathbb{R}}
99 \newcommand{\C}{\mathbb{C}}
100 \renewcommand{\Re}[1]{\operatorname{Re}{#1}}
101 \renewcommand{\Im}[1]{\operatorname{Im}{#1}}
102 \newcommand{\Zm}[1]{\mathbb{Z}_{#1}}
103 \renewcommand{\Prob}[1]{\mathbb{P}\left[ #1 \right]}
104 \newcommand{\ProbV}[2]{\mathbb{P}_{#1}\left[ #2 \right]}
105 \DeclareMathOperator{\Expect}{\mathbb{E}}
106 \newcommand{\wdt}{\widetilde}
107 \newcommand{\pt}[1]{\text{#1}^\circ}
108 \newcommand{\point}{\text{ }^\circ}
109 \newcommand{\hard}{\text{ }^*}

```

4.2 Переопределение кванторов

Проблема: команды `\exists` и `\forall` задают просто символ, а не оператор или что-то ещё. Из этого следует, что на печати надо самим заботиться о пробелах после кванторной связки. Поэтому команды переопределены и ставят `\;` после связки.

К тому же доопределён макрос `\existssone`.

Команды `\existssym` и `\forallsym` задают символы кванторов в старом смысле.

```

110 \let\existssym\exists
111 \renewcommand*{\exists}[1]{\existssym #1~\;}
112 \newcommand*{\existssone}[1]{\existssym! #1~\;}
113 \let\forallsym\forall
114 \renewcommand*{\forall}[1]{\forallsym #1~\;}

```


`\problems` `\problems` есть переписка с `\section*` из класса `article`. `\theme` — аналог `\subsection*` из класса `article`.
`\theme` Рекомендуется использовать, если задачи явно сгруппированы по какой-то теме. Команды `\problems` и `theme` есть только в классе `listok`

`vartab` (*env.*) Окружение `vartab{(amount)}` необходимо для набора таблиц различного размера. `amount` задаёт количество столбцов.

\conduit \conduit печатает кондуит. В случае устного собеседования — это табличка в пять строк и \cspproblem столбцов (то есть по одному столбцу на задачу). (Три оценки за каждую задачу и подпись). На листочке и контрольной — табличка с критериями оценки.

`enumitem` позволяет делать свои списки. Поэтому есть следующие списки:

itemize Каждый пункт на новой строке, делает так: \bullet , \bullet , \bullet , ...

probenum Каждый пункт на новой строке, делает так: **а.** , **б.** , **в.** , ...
 probparts Inline версия probenum, но делает так: **(а)** , **(б)** , **(в)** , ...
 multienum Есть один аргумент. Разбивает probenum на переданное число столбцов.

```

205 <*common>
206 \AddEnumerateCounter{\Asbuk}{\@Asbuk}{bl}
207 \AddEnumerateCounter{\asbuk}{\@asbuk}{yl}
208 \setlist[itemize]{nosep, nolistsep}
209 \newlist{probenum}{enumerate}{1}
210 \newlist{probparts}{enumerate*}{1}
211 \setlist[enumerate]{nosep, nolistsep}
212 \setlist[probenum]{nosep, nolistsep, label = \textbf{(\asbuk*)}}
213 \setlist[probparts]{nosep, nolistsep, label = \textbf{(\asbuk*)}}
214 \newenvironment{multienum}[1]
215 {
216   \begin{probenum}
217   \begin{multicols}{#1}
218 }{
219   \end{multicols}
220   \end{probenum}
221 }
222 \endinput
223 </common>
  
```