

Семейство классов listok *

Илья Райко
rayko_i@179.ru

5 ноября 2019 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}			
	<i>Title</i>	<i>Title. Вариант №num</i>	<i>... собеседование</i>	
Задача	\begin{problem}[Text]			
	Задача num.problem <i>Text</i>	Задача problem <i>Text</i>		
Пример	\begin{example}[Text]			
	Пример num.example <i>Text</i>	Пример example <i>Text</i>		
Теорема	\begin{theorem}[Text]			
	Теорема num.theorem (<i>Text</i>)	Теорема theorem (<i>Text</i>)		

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 A4, 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, includeheadf
6 \PassOptionsToClass{a4paper,oneside}{article}
7 </a4>
8 <common>\DeclareOption{a5paper}{\renewcommand{\@paperfile}{a5.clo}}
```

Чуть сложнее с a5paper. Он нужен для того, чтобы напечатать два вертикальных A5 на одном горизонтальном A4. Признаться честно, я уже забыл, что делают конкретные команды, но это, кажется, работает.

```
9 <*a5>
10 \setlength\paperheight {210mm}
11 \setlength\paperwidth {148mm}
12 \PassOptionsToClass{a5paper,oneside}{article}
13 \RequirePackage{pgfpages}
```

*Этот документ относится к listok v2.1, от 2019/01/09

```

14 \RequirePackage{atbegshi}
15 \setlength\hoffset{.4in}
16 \setlength\oddsidemargin{-1in}
17 \setlength\textwidth{130mm}
18 \setlength\hoffset{-25mm}
19 \setlength\textheight{185mm}
20 \pgfpagesuselayout{2 on 1}[a4paper,landscape,border shrink=5mm]
21 \AtBeginShipout{%
22 \pgfpageshipoutlogicalpage{1}\copy\AtBeginShipoutBox%
23 \pgfpageshipoutlogicalpage{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}

```

Все остальные опции передаются классу `article`.

```

30 \DeclareOption*{\PassOptionsToClass{\CurrentOption}{article}}
31 \ProcessOptions
32 \ProcessKeyvalOptions*
33 \LoadClass{article}
34 \input{\@paperfile}
35 \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 `rscur`, который не обновляется с конца 2000-х годов. И тут нас постигает совсем проблема: `rscur` 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 LuaLaTeX. 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.

```

36 \ifLuaTeX
37 \input{lua.clo}
38 </common>
39 <*lua>
40 \RequirePackage[TU]{fontenc}
41 \RequirePackage[lut8]{luainputenc}
42 \RequirePackage{fontspec}

```

```

43 \setmainfont{CMU Serif}
44 \lua
45 \common\else
46 \pdf\RequirePackage[utf8]{inputenc}
47 \pdf\RequirePackage{cmap}
48 \*common
49 \input{pdf.clo}
50 \fi

```

4 Фишки

Далее идут подключённые пакеты

```

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

```

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

```

65 \listok\newcommand{\listok@name}{\listok@num}
66 \test\newcommand{\listok@name}{\listok@num}
67 \written\newcommand{\listok@name}{}
68 \ustn\newcommand{\listok@name}{, }
69 \renewcommand{\maketitle}{%
70 \begin{center}%
71 {\Large \textit{\textbf{\underline{%
72 \listok\@title %
73 \test\@title. \listok@name %
74 \written
75 \ustn
76 }}}}}
77 \end{center}
78 }

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

```

4.1 Транскеанизация

Команда	Вывод
\emptyset	\emptyset
\le	\leq
\ge	\geq
\epsilon	ϵ
\phi	φ

<code>\N</code>	\mathbb{N}
<code>\Z</code>	\mathbb{Z}
<code>\Q</code>	\mathbb{Q}
<code>\R</code>	\mathbb{R}
<code>\Cx</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^*

```

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

```

108 \let\existssym\exists
109 \renewcommand*\exists[1]{\existssym #1~\;}
110 \newcommand*\existsone[1]{\existssym! #1~\;}
111 \let\forallssym\forall
112 \renewcommand*\forall[1]{\forallssym #1~\;}

```

4.3 Остальное

`abstract` Abstract имеет смысл использовать под разные комментарии. Текст печатается по центру курсивом.

```
113 \renewenvironment{abstract}{\quotation\itshape\centering}{\endquotation}
```

`theorem` Для вёрстки теорем, лемм, утверждений, следствий, определений и замечаний определены окружения.

`theorem*` Вариант со звёздочкой делает нумерованную теорему, etc.

```
lemma 114 \newtheorem{theorem}{ }
```

```
lemma* 115 \newtheorem*{theorem*}{ }
```

```
proposition 116 \newtheorem{lemma}{ }
```

```
proposition* 117 \newtheorem*{lemma*}{ }
```

`corollary`

`corollary*`

`definition`

```

118 \newtheorem{proposition}{      }
119 \newtheorem*{proposition*}{} 
120 \newtheorem{corollary}{        }
121 \newtheorem*{corollary*}{}     }
122 \theoremstyle{definition}
123 \newtheorem{note}{             }
124 \newtheorem*{note*}{}         }

problem Новый стиль для amsthm теорем пристраивает следующую цель. Хотим отрисовывать звёздочку около
problem* сложной задачи и кружочек около обязательной. Для этого будем использовать аргумент, который обычно
example используют для названия теорем!
example* Можно использовать и для примеров.

125 \newtheoremstyle{problem}{0pt}{0pt}{\normalfont}\{\bfseries\}.{ }\tname{#1}\tnumber{ #2}
126 \theoremstyle{problem}
127 \newtheorem{example}{       }
128 \newtheorem*{example*}{}    }
129 \newtheorem{problem}{      }
130 \newtheorem*{problem*}{}   }
131 \newtheorem*{definition}{} }
132 \renewcommand \theproblem {%
133 <listok> \listok@num.\@arabic@c@problem
134 <common&!listok>\@arabic@c@problem
135 }
136 \renewcommand \theexample {%
137 <listok> \listok@num.\@arabic@c@example
138 <common&!listok>\@arabic@c@example
139 }
140 \renewcommand \thetheorem {%
141 <listok> \listok@num.\@arabic@c@theorem
142 <common&!listok>\@arabic@c@theorem
143 }

\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

144 (*listok)
145 \newcommand \problemsf{@startsection[section]{1}{z@}{-3.5ex \@plus -1ex \@minus -.2ex}{2.3ex \
146 \newcommand \themef{@startsection[subsection]{2}{z@}{-3.25ex \@plus -1ex \@minus -.2ex}{1.5ex \
147
148 />}

varab Окружение varab{<amount>} необходимо для набора таблиц различного размера. amount задаёт количе-
ство столбцов.

149 \newenvironment{varab}[1]
150 {
151 \begin{tabular}{*{#1}{|c}| |c| } \hline
152 }{
153 \end{tabular}
154 }

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

155 (*ustn)
156 \newcounter{colidx}
157 \newcommand \conduit {
158
159 \vspace*{\fill}
160 \begin{center}
161 \begin{Large}
162 \begin{varab}{\c@problem}
163 \hline
164 \forloop{colidx}{1}{not{\value{colidx} > \c@problem}}{\; ; \; \; \$\Sigma$ \; \;
165 \forloop{colidx}{1}{not{\value{colidx} > \c@problem}}{&& \\ \hline
```

`\problems` — 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` 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 `\sc@problem` столбцов (то есть по одному столбцу на задачу). (Три оценки за каждую задачу и подпись). На листочке и контрольной — табличка с критериями оценки.

```

166 \forloop{colidx}{1}{\not\value{colidx} > \c@problem}{\&} \\\hline
167 \forloop{colidx}{1}{\not\value{colidx} > \c@problem}{\&} \\\hline
168 \forloop{colidx}{1}{\not\value{colidx} > \c@problem}{\&} \\\hline
169 \forloop{colidx}{1}{\not\value{colidx} > \c@problem}{\&} \\\hline
170 \forloop{colidx}{1}{\not\value{colidx} > \c@problem}{\&} \\\hline
171 \end{vartab}
172 \end{Large}
173 \end{center}
174 }
175 \</ustn>
176 \<*\listok | test>
177 \newcommand \conduit [4]{
178 \begin{center}
179 \begin{tabular}{|c|c|c|c|}
180 \hline
181 \multicolumn{4}{|c|}{ } \\\hline
182 \hline
183 <<5>> & <<4>> & <<3>> & <<2>> \\\hline
184 #1 & #2 & #3 & #4 \\\hline
185 \end{tabular}
186 \end{center}
187 }
188 \</listok | test>

```

enumitem позволяет делать свои списки. Поэтому есть следующие списки:

enumerate Каждый пункт на новой строке, делает так: 1. , 2. , 3. , ...

itemize Каждый пункт на новой строке, делает так: • , • , • , ...

enumerate*, itemize* Тоже, что и двое выше, inline версия (все пункты в одну строчку)

probenum Каждый пункт на новой строке, делает так: **a.** , **б.** , **в.** , ...

probparts Inline версия probenum, но делает так: **(a)** , **(б)** , **(в)** , ...

multienum Есть один аргумент. Разбивает probenum на переданное число столбцов.

```

189 \<*\common>
190 \AddEnumerateCounter{\Asbuk}{\@Asbuk}{ }
191 \AddEnumerateCounter{\asbuk}{\@asbuk}{ }
192 \setlist[itemize]{nosep, nolistsep}
193 \newlist{probenum}{enumerate}{1}
194 \newlist{probparts}{enumerate*}{1}
195 \setlist[enumerate]{nosep, nolistsep}
196 \setlist[probenum]{nosep, nolistsep, label = \textbf{\asbuk*}}
197 \setlist[probparts]{nosep, nolistsep, label = \textbf{(\asbuk)}}
198 \newenvironment{multienum}[1]
199 {
200 \begin{probenum}
201 \begin{multicols}{#1}
202 }{
203 \end{multicols}
204 \end{probenum}
205 }
206 \</endinput>
207 \</common>

```