

Семейство классов listok *

Илья Райко
rayko_i@179.ru

25 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>... собеседования</u>	
Задача	\begin{problem}[Text]			
	Задача num.problem Text	Задача problem Text		
Пример	\begin{example}[Text]			
	Пример num.example Text	Пример example Text		
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 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, includeheadfoot]{geometry}
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}
```

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

```

11 \setlength\paperwidth {148mm}
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     \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 `rsur`, который не обновляется с конца 2000-х годов. И тут нас постигает совсем проблема: `rsur` 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[utf8]{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, amsfonts, amsthm}
57 \RequirePackage{mathtools, mathabx}
58 \RequirePackage{epstopdf}
59 \RequirePackage{graphicx}
60 \RequirePackage{for loop}
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}{Листок №\listok@num}
66 <test>\newcommand{\listok@name}{Вариант №\listok@num}
67 <written>\newcommand{\listok@name}{}
68 <ustn>\newcommand{\listok@name}{Фамилия, Имя}
69 \renewcommand{\maketitle}{%
70   \begin{center}%
71     {\Large \textit{\textbf{\underline{%
72 <listok>\@title % Лорен ипсум
73 <test>\@title. \listok@name % Контрольная по Лорен ипсум. Вариант №N
74 <written>Письменное собеседование
75 <ustn>Устное собеседование
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 Трансокеанизация

Команда	Вывод
\emptyset	∅
\le	≤

<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>\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>	Text°
<code>Text\point</code>	Text°
<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{#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 связи. Поэтому команды переопределены и ставят `\;` после связи.

К тому же доопределён макрос `\existssone`.

Команды `\existssum` и `\forallssum` задают символы кванторов в старом смысле.

```

108 \let\existssum\exists
109 \renewcommand*\exists[1]{\existssum #1\;~}
110 \newcommand*\existssone[1]{\existssum! #1\;~}
111 \let\forallssum\forall
112 \renewcommand*\forall[1]{\forallssum #1\;~}

```

4.3 Остальное

`\closure` В «The Comprehensive L^AT_EX Symbol List» Скотт Пакин, рассуждая о различиях `\bar` и `\closure`, рекомендует промежуточный вариант, который придумал Энрико Грегорио и опубликовал во втором издании своего «Appunti di programmazione in L^AT_EX e T_EX», который дословно воспроизведён тут

```

113 \newcommand{\closure}[2][3]{%
114 {} \mkern#1mu\overline{\mkern-#1mu#2}}

abstract (env.) Abstract имеет смысл использовать под разные комментарии. Текст печатается по центру
курсивом.
115 \renewenvironment{abstract}{\quotation\itshape\centering}\endquotation}

theorem (env.) Для вёрстки теорем, лемм, утверждений, следствий, определений и замечаний определены
theorem* (env.) окружения. Вариант со звёздочкой делает нумерованную теорему, etc.
lemma (env.) 116 \newtheorem{theorem}{Теорема}
lemma* (env.) 117 \newtheorem*{theorem*}{Теорема}
proposition (env.) 118 \newtheorem{lemma}{Лемма}
proposition* (env.) 119 \newtheorem*{lemma*}{Лемма}
corollary (env.) 120 \newtheorem{proposition}{Утверждение}
corollary* (env.) 121 \newtheorem*{proposition*}{Утверждение}
corollary* (env.) 122 \newtheorem{corollary}{Следствие}
definition (env.) 123 \newtheorem*{corollary*}{Следствие}
definition* (env.)
note (env.) Попробую сделать крутые указания к задаче. Более того окружение само знает, что оно —
problem (env.) указание к последней задаче. Новый стиль для amsthm теорем приследует следующую
problem* (env.) цель. Хотим отрисовывать звёздочку около сложной задачи и кружочек около обязательной.
example (env.) Для этого будем использовать аргумент, который обычно используют для названия теорем!
example* (env.) Можно использовать и для примеров.
124 \newtheoremstyle{problem}{0pt}{0pt}{\normalfont}{\bfseries}{.}{ }{\thmname{#1}\thmnumber{ #2}\textit{#3}}
125 \theoremstyle{problem}
126 \newtheorem{example}{Пример}
127 \newtheorem*{example*}{Пример}
128 \newtheorem{problem}{Задача}
129 \newtheorem*{problem*}{Задача}
130 \newtheorem*{definition}{Определение}
131 \renewcommand \theproblem {%
132 \listok \listok@num.\@arabic\c@problem
133 \common&!\listok\@arabic\c@problem
134 }
135 \renewcommand \theexample {%
136 \listok \listok@num.\@arabic\c@example
137 \common&!\listok\@arabic\c@example
138 }
139 \renewcommand \thetheorem {%
140 \listok \listok@num.\@arabic\c@theorem
141 \common&!\listok\@arabic\c@theorem
142 }

\problems \problems есть переписка с \section* из класса article. \theme — аналог \subsection* из класса
\theme article. Рекомендуется использовать, если задачи явно сгруппированы по какой-то теме.
Команды \problems и theme есть только в классе listok
143 \*listok
144 \newcommand \problems{\@startsection{section}{1}{\z@}{-3.5ex \@plus -1ex \@minus -.2ex}{2.3ex \@plus .2ex}{\normalfont}
145 \newcommand \theme{\@startsection{subsection}{2}{\z@}{-3.25ex \@plus -1ex \@minus -.2ex}{1.5ex \@plus .2ex}{\normalfont}
146
147 \*listok

varab (env.) Окружение varab{\langle amount \rangle} необходимо для набора таблиц различного размера. amount
задаёт количество столбцов.
148 \newenvironment{varab}[1]
149 {
150 \begin{tabular}{*{#1}{|c| |c| } \hline
151 }{
152 \end{tabular}
153 }

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

```

```

155 \newcounter{colidx}
156 \newcommand \conduit {
157
158 \vspace*{\fill}
159 \begin{center}
160 \begin{Large}
161 \begin{vartab}{\c@problem}
162 \hline
163 \forloop{colidx}{1}{\not{\value{colidx} > \c@problem}}{\; \arabic{colidx} \; & \; \Sigma \; \; \hline
164 \forloop{colidx}{1}{\not{\value{colidx} > \c@problem}}{\& \; \hline
165 \forloop{colidx}{1}{\not{\value{colidx} > \c@problem}}{\& \; \hline
166 \forloop{colidx}{1}{\not{\value{colidx} > \c@problem}}{\& \; \hline
167 \forloop{colidx}{1}{\not{\value{colidx} > \c@problem}}{\& \;
168 \forloop{colidx}{1}{\not{\value{colidx} > \c@problem}}{\& \;
169 \forloop{colidx}{1}{\not{\value{colidx} > \c@problem}}{\& \; \hline
170 \end{vartab}
171 \end{Large}
172 \end{center}
173 }
174 \</ustn>
175 \<listok | test>
176 \newcommand \conduit [4]{
177 \begin{center}
178 \begin{tabular}{|c|c|c|c|}
179 \hline
180 \multicolumn{4}{|c|}{Критерии оценок} \\\
181 \hline
182 <<5>> & <<4>> & <<3>> & <<2>> \\\
183 #1 задач & #2 задач & #3 задач & #4 задач \\\
184 \end{tabular}
185 \end{center}
186 }
187 \</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.

```

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

```