

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>	... собеседования	
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 v2.1, от 2019/01/09

```

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{for loop}
62 \RequirePackage{datetime2}
63 \RequirePackage{microtype}
64 \relax
65 \tolerance{4000}

```

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

```

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

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

```

4.1 Трансокеанизация

Команда	Вывод
<code>\emptyset</code>	∅
<code>\le</code>	≤

<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^*

```

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

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

```

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

```

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

```

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

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

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

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

`lemma (env.)` 117 \newtheorem{theorem}{Теорема}

`lemma* (env.)` 118 \newtheorem*{theorem*}{Теорема}

`proposition (env.)` 119 \newtheorem{lemma}{Лемма}

`proposition* (env.)` 120 \newtheorem*{lemma*}{Лемма}

`corollary (env.)` 121 \newtheorem{proposition}{Утверждение}

`corollary* (env.)` 122 \newtheorem*{proposition*}{Утверждение}

`definition (env.)` 123 \newtheorem{corollary}{Следствие}

`definition* (env.)` 124 \newtheorem*{corollary*}{Следствие}

`note (env.)` Попробую сделать крутые указания к задаче. Более того окружение само знает, что оно — указание к последней задаче.

```
125 \newtheoremstyle{mynote}{0pt}{0pt}{\normalfont}{\itshape}{\bfseries}{\thmname{#1}\thmnumber{#2}\textit{ к задаче \theproblem{}}}{\thmname{#1}\thmnumber{#2}\textit{ к задаче \theproblem{}}}
126 \thmname{#1}\thmnumber{#2}\textit{ к задаче \theproblem{}}
127 }
128 \theoremstyle{mynote}
129 \newtheorem*{note}{Указание}
```

`problem (env.)` Новый стиль для `amsthm` теорем приследует следующую цель. Хотим отрисовывать звёздочку

`problem* (env.)` около сложной задачи и кружочек около обязательной. Для этого будем использовать аргумент,

`example (env.)` который обычно используют для названия теорем!

`example* (env.)` Можно использовать и для примеров.

```
130 \newtheoremstyle{problem}{0pt}{0pt}{\normalfont}{\bfseries}{\itshape}{\thmname{#1}\thmnumber{#2}\textit{#3}}
131 \theoremstyle{problem}
132 \newtheorem{example}{Пример}
133 \newtheorem*{example*}{Пример}
134 \newtheorem{problem}{Задача}
135 \newtheorem*{problem*}{Задача}
136 \newtheorem*{definition}{Определение}
137 \renewcommand \theproblem {%
138 \listok \listokenum.\@arabic\c@problem
139 \common&!listok\@arabic\c@problem
140 }
141 \renewcommand \theexample {%
142 \listok \listokenum.\@arabic\c@example
143 \common&!listok\@arabic\c@example
144 }
145 \renewcommand \thetheorem {%
146 \listok \listokenum.\@arabic\c@theorem
147 \common&!listok\@arabic\c@theorem
148 }
```

`\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`

```
149 \*listok
150 \newcommand \problems{\@startsection{section}{1}{\z@}{-3.5ex \@plus -1ex \@minus -.2ex}{2.3ex \@plus .2ex}{\normalfont}
151 \newcommand \theme{\@startsection{subsection}{2}{\z@}{-3.25ex \@plus -1ex \@minus -.2ex}{1.5ex \@plus .2ex}{\normalfont}
152
153 \listok}
```

`varab (env.)` Окружение `varab{<amount>}` необходимо для набора таблиц различного размера. `amount` задаёт количество столбцов.

```
154 \newenvironment{varab}[1]
155 {
156 \begin{tabular}{*{#1}{c} |c| } \hline
157 }{
158 \end{tabular}
159 }
```

`\conduit` `\conduit` печатает кондуит. 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.

```

160 <*ustn>
161 \newcounter{colidx}
162 \newcommand \conduit {
163
164 \vspace*{\fill}
165 \begin{center}
166 \begin{Large}
167 \begin{vartab}{\c@problem}
168 \hline
169 \forloop{colidx}{1}{\not{\value{colidx} > \c@problem}}{\; \arabic{colidx} \; & \; \; \$\Sigma$ \; \; \hline
170 \forloop{colidx}{1}{\not{\value{colidx} > \c@problem}}{\&} \; \hline
171 \forloop{colidx}{1}{\not{\value{colidx} > \c@problem}}{\&} \; \hline
172 \forloop{colidx}{1}{\not{\value{colidx} > \c@problem}}{\&} \; \hline
173 \forloop{colidx}{1}{\not{\value{colidx} > \c@problem}}{\&} \; \hline
174 \forloop{colidx}{1}{\not{\value{colidx} > \c@problem}}{\&} \; \hline
175 \forloop{colidx}{1}{\not{\value{colidx} > \c@problem}}{\&} \; \hline
176 \end{vartab}
177 \end{Large}
178 \end{center}
179 }
180 </ustn>
181 <*listok | test>
182 \newcommand \conduit [4]{
183 \begin{center}
184 \begin{tabular}{|c|c|c|c|}
185 \hline
186 \multicolumn{4}{|c|}{Критерии оценок} \; \; \hline
187 \hline
188 <<5>> & <<4>> & <<3>> & <<2>> \; \; \hline
189 #1 задач & #2 задач & #3 задач & #4 задач \; \; \hline
190 \end{tabular}
191 \end{center}
192 }
193 </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: а. , б. , в. , ...

`probparts` Inline версия `probenum`, но делает так: (а) , (б) , (в) , ...

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

```

194 <*common>
195 \AddEnumerateCounter{\Asbuk}{\@Asbuk}{bl}
196 \AddEnumerateCounter{\asbuk}{\@asbuk}{w}
197 \setlist[itemize]{nosep, nolistsep}
198 \newlist{probenum}{enumerate}{1}
199 \newlist{probparts}{enumerate*}{1}
200 \setlist[enumerate]{nosep, nolistsep}
201 \setlist[probenum]{nosep, nolistsep, label = \textbf{(\asbuk*)}}
202 \setlist[probparts]{nosep, nolistsep, label = \textbf{(\asbuk*)}}
203 \newenvironment{multienum}[1]
204 {
205 \begin{probenum}
206 \begin{multicols}{#1}
207 }{
208 \end{multicols}
209 \end{probenum}
210 }
211 \endinput

```

