# Семейство классов listok \*

Илья Райко rayko\_i@179.ru

31 марта 2023 г.

#### Аннотация

Классы, используемый нашей командой для вёрстки листков.

### 1 Введение

На данный момент у нас есть классы listok, written, ustn, test. listok используется для вёрстки листков, 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}			
	$\underline{Title}$	$Title.\ Bapua$ ห $m$ $N\!\!\!^{\circ}$ ทน $m$	co	беседование
Задача	\begin{problem}[Text]			
	${f 3}$ ада ${f 4}$ а num.problem ${\it Text}$	${f 3}$ адача problem ${\it Text}$		
Пример	\begin{example}[Text]			
	$oldsymbol{\Pi}$ рим $oldsymbol{ ext{ep}}$ num.example $\mathit{Text}$	$oldsymbol{\Pi}$ рим $oldsymbol{e}$ ехамр le $Text$		
Теорема	\begin{theorem}[Text]			
	ightharpoonup num.theorem $(Text)$	<b>Теорема</b> 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 а4рарет всё просто — напечатает одну копию на листе A4, ориентированном вертикально. Ширина текста будет 179mm, а высота 267mm. Текст выровнен по центру (вертикально и горизонтально).

- $4 \langle *a4 \rangle$
- 5 NRequirePackage[a4paper, width = 179mm, height = 267mm, footskip = 0mm, centering, includeheadfoot]{geometry
- 6 \PassOptionsToClass{a4paper,oneside}{article}
- 7 (/a4)
- $\begin{tabular}{l} 8 $$ (common) \end{tabular} $$ (common) \end{tab$

Чуть сложнее с а5рарег. Он нужен для того, чтобы напечать 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}

<sup>\*</sup>Этот документ относится к listok v2.3, от <math>2023/03/31

```
\PassOptionsToClass{a5paper,oneside}{article}
12
13
      \RequirePackage{pgfpages}
      \RequirePackage{atbegshi}
14
      \setlength\hoffset{.4in}
15
      \setlength\oddsidemargin{-1in}
16
      \setlength\textwidth{130mm}
17
      \setlength\voffset{-25mm}
18
      \setlength\textheight{185mm}
19
      \pgfpagesuselayout{2 on 1}[a4paper,landscape,border shrink=5mm]
20
      \AtBeginShipout{%
21
          \pgfpagesshipoutlogicalpage{1}\copy\AtBeginShipoutBox%
22
          \pgfpagesshipoutlogicalpage{2}\box\AtBeginShipoutBox%
23
          \pgfshipoutphysicalpage%
24
      }
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 ОТ2 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 Т2. Проблемы в том, что был только один такой шрифт, в котором русские буквы имеют равную высоту. Этот шрифт распространялся пакетом рѕсуг, который не обновляется с конца 2000-х годов. И тут нас постигает совсем проблема: рѕсуг сегодня поддерживается только дистрибутивом MiKTEX.

Поэтому предлагается использовать расширенный движок LuaT<sub>E</sub>X. Если не вдаваться в подробности, то имеет место простая формула:

```
LuaT_EX = pdfT_EX + Поддержка Unicode и OpenType шрифтов+ 
 <math>+ e-T_EX + Omega + 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 pdfl₄теX).

Движок определяется автоматически по имени запускаемой программы.

```
\begin{array}{ll} 37 \land \mathsf{ifLuaTeX} \\ 38 \qquad \land \mathsf{input\{lua.clo\}} \\ 39 \ \big\langle \mathsf{/common} \big\rangle \end{array}
```

```
40 ⟨*lua⟩
41
       \RequirePackage[TU]{fontenc}
       \RequirePackage[lutf8]{luainputenc}
42
43
       \RequirePackage{fontspec}
       \setmainfont{CMU Serif}
44
45 \langle /lua \rangle
46 ⟨common⟩\else
48 \(\rangle pdf \) \(\rangle \rangle \text{quirePackage} \) (cmap)
49 (*common)
       \input{pdf.clo}
51 \fi
```

#### 4 Фишки

```
Далее идут подключённые пакеты
 52 \RequirePackage[russian]{babel}
53 \ \ensuremath{\mbox{\sc 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
Nlistok@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}{%}
        \begin{center}%
 72
        {\Large \textit{\textbf{\underline{%
 73
 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
        \end{center}
 79
 80 }
81 \DTMsetup{datesep=.}
 82 \DTMsetstyle{ddmmyyyy}
 83 \text{ renewcommand} \text{ the footnote} \{\text{nsymbol} \{\text{footnote}\}\}
 84 \renewcommand{\ensuremath{\columnwddhead}}{
 85
        \vbox{\hbox to\textwidth{\listok@name\hfil \strut
        \listok@date
 86
       }\hrule}
 87
 88 }
 89 \renewcommand{\@oddfoot}{}
```

### 4.1 Трансокеанизация

Команда	Вывод	
\emptyset	Ø	
\le	<b>≼</b>	
\ge	≽	
\epsilon	$\varepsilon$	
<b>\phi</b>	$\varphi$	
\N	N	
١Z	$\mathbb{Z}$	
\Q	Q	
\R	$\mathbb{R}$	
\Cx	$\mathbb{C}$	
∖Re z	$\operatorname{Re} z$	
∖Im z	$\operatorname{Im} z$	
\Zm{p}	$\mathbb{Z}_p$	
\Prob{Event}	$\mathbb{P}\left[Event\right]$	
$\P$	$\mathbb{P}_{\xi}\left[Event\right]$	
$\Expect{\xi}$	$\mathbb{E} \xi$	
\pt{Text}	Text°	
Text\point	Text°	
Text∖hard	Text*	

```
90 \renewcommand{\emptyset}{\varnothing}
    91 \renewcommand{\le}{\leqslant}
    92 \renewcommand{\ge}{\geqslant}
    93 \renewcommand{\epsilon}{\varepsilon}
    94 \renewcommand{\phi}{\varphi}
    99 \newcommand{\Cx}{\mathbb{C}}}
 100 \renewcommand{\Re}[1]{\operatorname{Re}{#1}}
101 \renewcommand{\Im}[1]{\operatorname{Im}{#1}}
102 \mbox{ newcommand} \mbox{\command} \mbox
104 \mbox{ } \mbox{mewcommand} \mbox{ProbV}[2]{\mathbb{P}_{\#1}\times [eft [ #2 \mbox{ } ]}
105 \DeclareMathOperator{\Expect}{\mathbb{E}}}
106 \ge 106 \le 106 
107 \newcommand{\pt}[1]{$\text{#1}^\circ$}
108 \rightarrow 108 \rightarrow 108
109 \mbox{newcommand{\hard}{${}^{ast$}}}
```

### 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.

K тому же доопределён макрос \existsone.

Команды vexistssym и vforallsym задают символы кванторов в старом смысле.

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

#### 4.3 Остальное

```
vctosure «Крышка» сверху. В «The Comprehensive IATEX 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 LATEX e TEX», который
                  дословно воспроизведён тут
                  115 \rightarrow 115 \rightarrow 115
                  116 {}\mkern#1mu\overline{\mkern-#1mu#2}}
   abstract (env.) Abstract имеет смысл использовать под разные комментарии. Текст печатается по центру курсивом.
                  117 \renewenvironment{abstract}{\quotation\itshape\centering}{\centquotation}
    theorem (env.) Для вёрстки теорем, лемм, утверждений, следствий и замечаний определены окружения. Вариант
   theorem* (env.) со звёздочкой делает ненумерованную теорему, etc.
      lemma (env.) 118 \newtheorem{theorem}{Teopema}
     lemma* (env.) 119 \newtheorem*{theorem*}{Teopema}
proposition (env.) 120 \mbox{ newtheorem{lemma}{Imma}}
proposition* (env.) ^{121} \newtheorem*{lemma*}{Лемма}
  corollary (env.) 122 \newtheorem{proposition}{Утверждение}
                  123 \newtheorem*{proposition*}{Утверждение}
 corollary* (env.) 124 \newtheorem{corollary}{Cледствие}
                  125 \newtheorem*{corollary*}{Следствие}
       note (env.) Попробую сделать крутые указания к задаче. Более того окружение само знает, что оно — указание
                  к последней задаче.
                  126 \text{ } \text{mewtheoremstyle\{mynote}\{0pt\}\{0pt\}\{\text{normalfont}\}\{\}\{\text{itshape}\}\{\}\{\}\}
                         \thmname{#1}\thmnumber{ #2}\textit{ к задаче \theproblem{}:}
                  128 }
                  129 \theoremstyle{mynote}
                  130 \newtheorem*{note}{Указание}
   solution (env.) На основе указания сделаны решения. Верстаются только если класс вызван с опцией solutions.
                  131 \iflistok@solutions
                  132 \newtheoremstyle{solution}{0pt}{0pt}{\normalfont}{}{\itshape}{}{ }{\%
                         \thmname{#1}\thmnumber{ #2}\textit{ задачи \theproblem{}:}
                  133
                  134 }
                  135 \theoremstyle{solution}
                  136 \newtheorem*{solution}{Решение}
                  137 \else
                  138 \excludecomment{solution}
                  139 \f i
                      Новый стиль для amsthm теорем приследует следующую цель. Хотим отрисовывать звёздочку
    problem (env.)
   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.
                  141 \theoremstyle{problem}
                  142 \newtheorem{example}{∏pumep}
                  143 \newtheorem*{example*}{∏pumep}
                  144 \newtheorem{problem}{Задача}
                  145 \newtheorem*{problem*}{Задача}
 definition (env.) В отличие от всех остальных окружений, definition — ненумерованный вариант, а definition* —
definition* (env.) нумерованный.
                  146 \mbox{newtheorem*{definition}{Oпределение}}
                  147 \newtheorem{definition*}{Определение}
                  148 \renewcommand \theproblem {%
                  149 (listok)
                               \listok@num.\@arabic\c@problem
                  150 \ \langle {\tt common} \& ! {\tt listok} \rangle \\ \verb|\| \texttt{Carabic} \\ \verb|\| \texttt{Ceproblem} \\
                  151 }
                  152 \renewcommand \theexample {%
                  153 (listok)
                               \listok@num.\@arabic\c@example
                  154 (common&!listok)\@arabic\c@example
                  155 }
```

```
156 \renewcommand \thetheorem {%
                      157 (listok)
                                         ∖listok@num.\@arabic\c@theorem
                      158 (common&!listok)\@arabic\c@theorem
     \problems \problems ecть 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
                      160 (*listok)
                      162 \newcommand \theme{\@startsection{subsection}{2}{\z@}{-3.25ex\@plus -1ex \@minus -.2ex}{1.5ex \@plus .2ex}{\normale \normale 
                      164 (/listok)
vartab (env.) Окружение vartab\{\langle amount \rangle\} необходимо для набора таблиц различного размера. amount задаёт
                      количество столбнов.
                      165 \newenvironment{vartab}[1]
                      166 {
                      167
                                 \left( x^{+1} \right) = \left( x^{+1} \right) 
                      168 }{
                                 \end{tabular}
                      169
                      170 }
       vconduit vconduit печатает кондуит. В случае устного собеседования — это табличка в пять строк и vceproblem
                      столбцов (то есть по одному столбцу на задачу). (Три оценки за каждую задачу и подпись). На
                     листочке и контрольной — табличка с критериями оценки.
                     171 (*ustn)
                      172 \newcounter{colidx}
                     173 \newcommand \conduit {
                     175 \vspace*{\fill}
                      176 \begin{center}
                      177 \begin{Large}
                     178 \begin{vartab}{\c@problem}
                     179 \hline
                      181 \forloop{colidx}{1}{\not{\value{colidx} > \c@problem}}{&} \white
                     182 \forloop{colidx}{1}{\not{\value{colidx} > \c@problem}}{&} \white
                      183 \forloop{colidx}{1}{\not{\value{colidx} > \c@problem}}{&} \w \hline
                      184 \forloop{colidx}{1}{\not{\value{colidx}} \rightarrow \cent{colidx} \
                      185 \forloop{colidx}{1}{\not{\value{colidx} > \c@problem}}{&} \\
                      187 \end{vartab}
                      188 \end{Large}
                     189 \end{center}
                     190 }
                      191 (/ustn)
                      192 (*listok | test)
                      193 \newcommand \conduit [4]{
                      194 \begin{center}
                      195 \begin{tabular}{|c|c|c|c|}
                      196 \hline
                      197 \multicolumn{4}{|c|}{Критерии оценок} \\
                      198 \hline
                      199 <<5>> & <<4>> & <<3>> & <<2>> \\ \hline
                      200 #1 задач & #2 задач & #3 задач & #4 задач \\ \hline
                      201 \end{tabular}
                     202 \end{center}
                     203 }
                     204 (/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)
```

```
ргоbепит Каждый пункт на новой строке, делает так: {\bf a.} , {\bf 6.} , {\bf 8.} , ...
      probparts Inline версия probenum, но делает так: (a) , (b) , (b) , ...
     multienum Есть один аргумент. Разбивает probenum на переданное число столбцов.
205 \langle *common \rangle
206 \verb| AddEnumerateCounter{\Asbuk}{\Ceasbuk}{b}
208 \setlist[itemize]{nosep, nolistsep}
209 \ge 100 = 209 \le 100 = 200 \le 100 = 200 \le 100 \le 100 = 100 \le 100 
210 \newlist{probparts}{enumerate*}{1}
211 \setlist[enumerate]{nosep, nolistsep}
212 \setlist[probenum]{\nosep, \nolistsep, \label = \textbf{(\asbuk*)}}
213 \texttt{\scale}[probparts] \{nosep, nolistsep, label = \texttt{\scale}(\texttt{\scale})\} \}
214 \newenvironment{multienum}[1]
215 {
                                \begin{probenum}
216
                                 \begin{multicols}{#1}
217
218 }{
                                \end{multicols}
219
220
                                \end{probenum}
221 }
222 \endinput
223 \langle /common \rangle
```