

Рекомендации по подготовке статей на VIII международную конференцию «Компьютерные науки и информационные технологии»*

Оргкомитет конференции КНИТ-2018

knit2018@sgu.ru

*Саратовский Национальный Исследовательский Государственный Университет
имени Н.Г. Чернышевского, Саратов, Россия*

Рекомендации по подготовке статей в формате $\LaTeX 2_{\epsilon}$ с использованием стилевого файла `csit-2018.sty`.

Ключевые слова: материалы конференции, порядок оформления статей, набор в формате $\LaTeX 2_{\epsilon}$

Исходный текст статьи готовится и представляется в формате $\LaTeX 2_{\epsilon}$. Имя файла со статьёй — написанная латиницей фамилия первого автора статьи. Если автор представляет на конференцию несколько статей, к фамилии автора в названии нужно добавить номер статьи.

Весь текст статьи в формате материалов конференции должен занимать от двух до четырех страниц (для пленарных докладов — до шести страниц). Превышение этого объема приведет к увеличению размера оргвзноса.

Работу над статьёй удобно начинать с редактирования файла-образца `csit-2018-example.tex`.

Для опубликования статьи в сборнике необходимо предоставить исходный файл статьи в формате $\LaTeX 2_{\epsilon}$ (файл с расширением `.tex`), а также файлы всех подключаемых элементов текста (файлы изображений, включаемые в статью файлы исходного кода программ и т. п.).

1. Общий формат статьи и титульная информация

Текст файла со статьёй должен начинаться со строки

```
\documentclass[10pt]{article}
\usepackage{csit-2018}
\begin{document}
```

Команда `\usepackage` подключает стилевой файл `csit-2018.sty`, который должен располагаться в той же директории, что и сама статья.

Если статья написана по-английски, то это надо указать явно, сразу после `\begin{document}` (иначе не включатся английские переносы слов):

```
\English
```

Затем формируется заголовок статьи, включая ссылку на грант и аннотацию:

```
\title[Краткое название]{Полное название}
\author{Фамилия~И.\,0.}
\email{author@site.ru}
\organization{Город, Организация}
\abstract{Данная статья посвящена...}
\keywords{ключевое слово 1, ...}
\thanks{Работа выполнена при поддержке гранта...}
```

Все эти команды могут идти в произвольном порядке и должны завершаться командой

```
\maketitle
```

В случае, когда у статьи несколько авторов, в команде `\author` после фамилии и инициалов каждого автора необходимо поместить команду `\authorrefmark` с порядковым номером автора в качестве аргумента.

```
\author{Автор~И.\,0.\authorrefmark{1}, Соавтор~И.\,0.\authorrefmark{2}}
```

Иная расстановка инициалов, пробелов или запятых в аргументе команды `\author` может приводить к ошибкам в оглавлении и в авторском указателе.

В аргументе команды `\email` нужно перечислить через запятую электронные адреса всех авторов. Перед каждым электронным адресом нужно поместить команду `\authorrefmark` с порядковым номером соответствующего автора.

```
\email{\authorrefmark{1}author1@site.ru, \authorrefmark{2}author2@site.ru}
```

Конференция проводится при поддержке РФФИ.

В команде `\organization` указываются названия организаций авторов в порядке: полное название организации, город, страна.

Если соавторы из разных организаций, то в аргументе команды `\organization` нужно перечислить через запятую организации всех авторов. Перед названием организации каждого автора в команде `\organization` необходимо поместить соответствующую команду `\authorrefmark`.

```
\organization{\authorrefmark{1}НИИ-X, \authorrefmark{2}НИИ-Y, Москва, Россия}
```

Ссылка на грант(ы) оформляется как часть заголовка командой `\thanks` и выводится в виде сноски на первой странице статьи.

В команде `\abstract` задается аннотация к статье. Аннотация (не более 10 строк) не должна содержать ссылок на литературу, формул, таблиц, рисунков.

Команда `\keywords` задает набор (обычно 3–6) ключевых слов/словосочетаний. Список ключевых слов начинается со строчного символа. Точка в конце списка не ставится.

Текст статьи можно разбивать на разделы и параграфы командами

```
\section*{Название раздела}
\section{Название раздела}
\paragraph{Название параграфа.}
```

Команду `\section*{Название}` следует использовать для нумеруемых разделов статьи. В частности, если статья содержит разделы «Введение» и «Заключение», то они оформляются как нумеруемые. Заголовок раздела «Список литературы» генерируется автоматически.

Команда `\section{Название}` оформляет нумеруемый заголовок раздела. Номер генерируется автоматически, указывать его в тексте названия не следует.

В конце названий разделов точка не ставится. Название параграфа является частью первой строки абзаца; если это целое предложение, то точка ставится перед закрывающей фигурной скобкой.

Команды `\subsection`, `\subparagraph` рекомендуется не использовать, так как статьи имеют небольшой объём.

Не следует использовать иные способы секционирования статьи кроме вышеупомянутых.

Файл статьи должен заканчиваться командой

```
\end{document}
```

Убедительная просьба — не использовать команды сокращения вертикальных промежутков и другие способы искусственного уплотнения/разрежения текста.

2. Стандартные средства \LaTeX 'а

Нет особых ограничений на использование основных средств \LaTeX 'а [1, 2, 3, 4]. В статью можно вставлять формулы, таблицы, списки, рисунки, сноски, и т. д. Определения ссылок `\label` и команд `\newcommand`, `\renewcommand` действуют только внутри одной статьи; конфликты с чужими статьями исключены.

Стандартные пакеты, подключённые в стилевом файле `csit-2018.sty`: `algorithm`, `algorithmic`, `amssymb`, `amsmath`, `array`, `babel`, `balance`, `color`, `enumitem`, `epic`, `euscript`, `graphicx`, `ifthen`, `inputenc`, `listings`, `mathrsfs`, `multicol`, `multirow`, `pb-diagram`, `theorem`, `tikz`, `url`, `xy`. Этими пакетами можно пользоваться, не вызывая команду `\usepackage`.

Формулы внутри текста, даже очень короткие, необходимо окружать знаками доллара $\$$:

число $\$-3.14\$$	число -3.14 — верно
число -3.14	число -3.14 — неверно
объект $\sim\$\$$	объект x — верно
объект x	объект x — неверно

Выключные формулы без номера окружаются скобками `\[` и `\]`. Выключные формулы с номером окружаются командами `\begin{equation}` и `\end{equation}`. Команда `\label{name}` между ними даёт метку формулы. Русские буквы в именах меток `name` не допустимы. Метка позволяет ссылаться на формулу командой `\eqref{name}`, например команда `\eqref{eqCases}` даёт (1).

Списки оформляются стандартными окружениями `enumerate` или `itemize`.

Таблицы создаются окружением `tabular` и оформляются как плавающие с помощью окружения `table`. Подпись делается *над таблицей* командой `\caption`, см. таблицу 1. Команда `\label`, определяющая ссылку на номер таблицы, обязана идти после `\caption`. Шрифт в таблице должен быть уменьшен до `\small`. Если таблица не умещается по ширине текста, то можно уменьшить шрифт до `\footnotesize`, либо уменьшить интервалы между колонками: `\tabcolsep=2pt`.

Таблица 1. ПОДПИСЬ РАЗМЕЩАЕТСЯ НАД ТАБЛИЦЕЙ.

Задача	CCEL	boosting
Cancer	3.46 ± 0.37 (3.16)	4.14 ± 1.48
German	25.78 ± 0.65 (1.74)	29.48 ± 0.93
Hepatitis	18.38 ± 1.43 (2.87)	19.90 ± 1.80

Иллюстрации могут быть подготовлены в любом графическом формате, поддерживаемом последними версиями пакета MiKTeX , в частности, BMP, PNG, PDF или EPS. Не забудьте прислать графические файлы вместе с TeX -файлом! Имена графических файлов должны начинаться с фамилии первого автора латиницей, как и основной файл статьи.

Желательно, чтобы рисунки были чёрно-белыми или grayscale (оттенки серого). При чёрно-белой печати передача цвета плохо предсказуема. Растровые рисунки должны быть подготовлены с разрешением не менее 600 dpi. Рисунки вставляются командой `\includegraphics`. Если рисунок занимает по высоте более 1–2 см, то он оформляется как плавающая иллюстрация `figure`. Подпись делается *под рисунком* командой `\caption`, см. рис. 1.

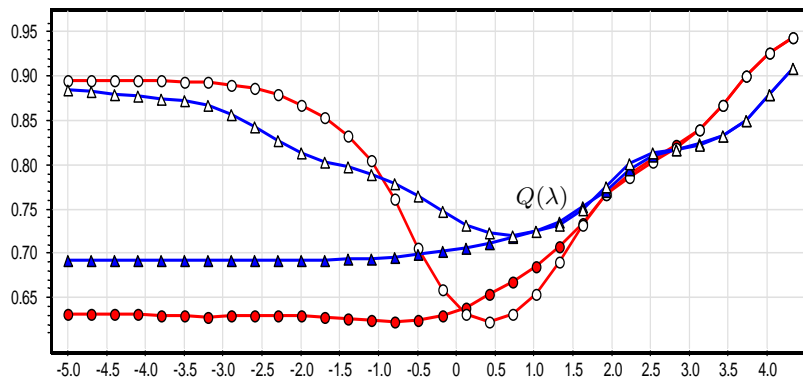


Рис. 1. Подпись должна размещаться под рисунком. **ВНИМАНИЕ!** Красные и синие линии при печати будут выглядеть как чёрные.

Определена команда `\XYtext(x,y){text}`, для надписей поверх рисунков. Например, так сделана надпись « $Q(\lambda)$ » на рис. 1. Координаты левого нижнего угла надписи (x, y) подбираются вручную относительно правого нижнего угла рисунка.

Запрещается использовать пакеты, размещающие рисунки сбоку или влияющие на двухколоночный режим: `multicol`, `floatfig`, `floatflt`, и т. п.

Сноски делаются командой `\footnote{text}`¹.

Программный код оформляется с помощью окружения `lstlisting`, которому, дополнительным параметром передается имя метки для ссылки и подпись к программному коду. Предварительной командой `\lstset` устанавливается язык программирования. Например, текст

```
\lstset{language=[Sharp]C} % Язык C#
\begin{lstlisting}[label=code:lab1, caption=Пример кода на C\#]
private Stream stream
{
    get
    {
        if (context.stream == null)
        {
            context.stream = new FileStream(context.path);
            reader = new StreamReader(context.stream);
        }
        return context.stream;
    }
}
\end{lstlisting}
```

приведет к получению кода, представленного листингом 1.

¹Текст сноски указывается в аргументе *text*.

Листинг 1. Пример кода на C#

```
private Stream stream
{
    get
    {
        if (context.stream == null)
        {
            context.stream = new FileStream(context.path);
            reader = new StreamReader(context.stream);
        }
        return context.stream;
    }
}
```

Извлечения из программного кода (имена переменных, фрагменты выражений), используемые в предложениях, оформляются с помощью команды `\lstinline|text|`, где фрагмент кода указывается в аргументе `text`, а вместо символов `|` может использоваться любой символ, не содержащийся в фрагменте. Например фрагмент `new StreamReader(context.stream)` вставлен в результате выполнения команды `\lstinline|new StreamReader(context.stream)|`.

Список литературы формируется окружением `thebibliography`. Каждая запись библиографии начинается командой `\bibitem{name}`. Метка `name` позволяет сослаться на данную запись командой `\cite{name}`. В ссылках разрешается указывать несколько меток через запятую: `\cite{name1,name2}`. Новая команда `\citenb` даёт ссылку без квадратных скобок, что позволяет делать интервалы; например, [1–4] было получено так: `[\citenb{VoronLatex}--\citenb{Lvovsky}]`. Русские буквы в именах меток `name` недопустимы. Записи сортируются по авторам в порядке русского, затем латинского алфавита.

Фамилии и инициалы авторов выделяются командой `\BibAuthor`, причем инициалы указываются после фамилии. Названия статей в сборниках выделяются командой `\BibTitle`. Если публикация существует только в электронном виде, веб-ссылка даётся командой `\BibUrl`.

Стоит придерживаться оформления, показанного на примере библиографического списка в файле `csit-2018-example.tex`. В остальном старайтесь придерживаться требований ГОСТ 7.80-00.

Глобальные ссылки. В стиле `csit-2018.sty` определены команды `\globallabel`, `\globalref`, `\globalpageref`, позволяющие сослаться из одной статьи на любое место в другой статье. Это полные аналоги стандартных команд `\label`, `\ref`, `\pageref`, но определяемые ими метки доступны во всём сборнике. Типичное применение этой возможности — указать в библиографии диапазон страниц другой статьи «в настоящем сборнике»:

```
C.\, \globalpageref{Kozlov:begin}--%
    \globalpageref{Kozlov:end}
```

Для каждой статьи в сборнике по умолчанию определены две метки `\globallabel{file:begin}` и `\globallabel{file:end}`, где `file` — имя `tex`-файла статьи, без указания расширения.

Ссылки на сайты делаются командой `\url`. При вёрстке документа в формате PDF ссылки становятся активными, хотя не подчёркиваются и не выделяются цветом. Пример: `\url{www.mmro.ru}`.

3. Математические обозначения

Следование приводимым ниже рекомендациям способствует большему единообразию в обозначениях и облегчает подготовку сборника.

Целочисленные интервалы обозначаются только как $1, \dots, n$. Варианты $\overline{1, n}$ или $1, \dots, i, \dots, n$ или $1, 2, \dots, n$ не допустимы. То же относится к векторам и спискам переменных вида x_1, \dots, x_n .

В качестве десятичного разделителя используется точка.

Числовые множества \mathbb{N} , \mathbb{Z} , \mathbb{R} , \mathbb{C} делаются командами `\NN`, `\ZZ`, `\RR`, `\CC`.

В стиле `csit-2018.sty` переопределены команды `\geq`, `\leq`, `\emptyset`, `\epsilon`, `\kappa`, `\phi` математических символов \geq , \leq , \emptyset , ε , \varkappa , φ .

Математические операторы `lim`, `inf`, `sup`, `min`, `max` переопределены так, что пределы всегда ставятся снизу, а не сбоку.

Определены математические операторы: `arg min`, `arg max`, `diag`, `sign`, `tr`, `const` командами `\argmin`, `\argmax`, `\diag`, `\sign`, `\Tr`, `\const`.

Команды `\myop` и `\mylim` производят новые операторы, не предусмотренные L^AT_EX'ом:

<code>\myop{Ker} f\$</code>	$\text{Ker } f$
<code>\$A_{\myop{Ker} f}\$</code>	$A_{\text{Ker } f}$
<code>\myop{Hom}_{\Phi}(A, B)\$</code>	$\text{Hom}_{\Phi}(A, B)$
<code>\mylim{Hom}_{\Phi}(A, B)\$</code>	$\text{Hom}_{\Phi}(A, B)$

Для выделения векторных и матричных величин прямым жирным шрифтом предусмотрена команда `\vec{формула}`.

Линейная алгебра:

<code>\rank A</code>	$\text{rank } A$
<code>\Tr A</code>	$\text{tr } A$
<code>\diag (d_1, \dots, d_n)</code>	$\text{diag}(d_1, \dots, d_n)$
<code>A\T</code>	A^T
<code>u\T F\T F u</code>	$u^T F^T F u$
<code>\vec x</code>	\mathbf{x}
<code>\Omega \neq \vec{\Omega}</code>	$\Omega \neq \boldsymbol{\Omega}$
<code>e^{-\vec{x}\T\Sigma x}</code>	$e^{-\mathbf{x}^T \Sigma \mathbf{x}}$ (верно)
<code>e^{-x\T\Sigma x}</code>	$e^{-x^T \Sigma x}$ (неверно)

Теория вероятностей:

<code>\Prob\{x\colon x\in A\}</code>	$P\{x: x \in A\}$
<code>\Expect \xi</code>	$E\xi$
<code>\Var \xi</code>	$D\xi$
<code>\Normal(\mu, \Sigma)</code>	$\mathcal{N}(\mu, \Sigma)$
<code>p(x\cond y)</code>	$p(x y)$

В условных вероятностях команда `\cond` даёт правильные пробелы вокруг вертикальной черты.

Теория вычислительной сложности:

<code>\P</code>	P
<code>\NP</code>	NP
<code>\DTIME</code>	$DTIME$
<code>\MaxSNP</code>	Max-SNP
<code>\Apx</code>	Apx
<code>\PC</code>	PC
<code>\MinPC</code>	$MINPC$
<code>\threeSAT</code>	$3SAT$
<code>\GapSAT</code>	$GAP-3SAT$

Легко определять собственные такие команды для новых классов сложности и задач, например, класс NP и задача $MINPC$ были определены так:

```
\def\NP{\CCfont{NP}}
\def\MinPC{\CPfont{MinPC}}
```

Все эти команды могут употребляться как внутри формул, так и непосредственно в тексте.

Для оформления условных конструкций пользуйтесь стандартным окружением `cases`. Текст внутри формул выводится командой `\text`:

$$y(x, \alpha) = \begin{cases} -1, & \text{если } f(x, \alpha) < 0; \\ +1, & \text{если } f(x, \alpha) \geq 0. \end{cases} \quad (1)$$

```
\begin{equation}\label{eqCases}
y(x, \alpha) = \begin{cases}
-1, & \text{\text{если } } f(x, \alpha) < 0; \ \backslash \\
+1, & \text{\text{если } } f(x, \alpha) \geq 0.
\end{cases}
\end{equation}
```

Чтобы размер скобок соответствовал размеру обрамляемой формулы, пользуйтесь командами `\left` и `\right`. Однако в простых случаях эти команды не нужны и только загромождают текст. Лучше записать $f(x_i)$, чем $f\left(x_i\right)$ — результат в обоих случаях будет одинаков.

Для вставки матрицы в строку текста $\begin{pmatrix} a & b & c \\ 1 & 2 & 3 \end{pmatrix}$ используйте окружение `smallmatrix`.

Для разбиения длинных формул на несколько строк используйте окружения `align`, `gather`, `multline` или `split`. Окружение `array` для этой цели не подходит.

Окружения типа теорем. Следующие окружения выводят заключённый в них текст *наклонным шрифтом*: `Def` или `Definition` — Определение, `Theorem` — Теорема, `Lemma` — Лемма, `State` — Утверждение, `Corollary` — Следствие.

Следующие окружения выводят заключённый в них текст обычным шрифтом: `Axiom` — Аксиома, `Problem` — Задача, `Example` — Пример, `Remark` — Замечание, `Hypothesis` — Гипотеза.

Окружение `Proof` — Доказательство, выводит заключённый в него текст обычным шрифтом, а в конце последней строки выводит знак `QED`.

Алгоритмы оформляются в стиле псевдокода с помощью плавающего окружения `Algorithm`. Внутри окружения `Algorithm` стандартные ключевые слова `\IF`, `\FOR`, `\WHILE`, и др., дают, соответственно, **если**, **для**, **пока**, и т. д. Шаги алгоритма нумеруются автоматически, и на них можно ссылаться, см. важный шаг 5 алгоритма 1.

Алгоритм 1 Показаны все команды управления.

Вход: x, y ;
Выход: $z = F(x, y)$;
 1: инициализация: $b := a$;
 2: для $i = 1, \dots, n$
 3: для всех $w \in W$ таких, что $w \geq 0$
 4: **повторять**
 5: важный шаг: вычисление вектора u_i ;
 6: **пока** $\|u_i - u_{i-1}\| > \varepsilon$;
 7: **если** $a > 0$ **то**
 8: **пока** $W \neq \emptyset$
 9: $W := W - \{a\}$;
 10: **иначе если** $a = 0$ **то**
 11: **цикл** // бесконечный цикл
 12: при определённых условиях **выход**;
 13: **иначе** // при $a < 0$
 14: $a := 1$;

```
\begin{Algorithm}[t]
\caption{Показаны все команды управления.}
\label{AlgExample}
\REQUIRE $x, y$;
\ENSURE $z = F(x,y)$;
\STATE инициализация: $b := a$;
\FOR{$i=1,\dots,n$}
  \FORALL{$w \in W$ таких, что $w \geq 0$}
    \REPEAT
      \STATE\label{algCalcU}
      важный шаг: вычисление вектора~$u_i$;
      \UNTIL{$\|u_i-u_{i-1}\|>\epsilon$};
    \ENDFOR
  \ENDFOR
\IF{$a>0$}
  \WHILE{$W \neq \varnothing$}
    \STATE $W := W - \{a\}$;
  \ENDWHILE
\ELSIF{$a=0$}
  \LOOP[бесконечный цикл]
    \STATE при определённых условиях \EXIT;
  \ENDLOOP
\ELSE[при $a<0$]
  \STATE $a:=1$;
\ENDIF
\end{Algorithm}
```

4. Рекомендации по оформлению

Придерживаясь следующих правил, авторы существенно облегчают подготовку сборника.

Некоторые правила типографики. Скобки всех видов набираются вплотную к тексту, который они окружают. Знаки препинания набираются слитно с предшествующим текстом и отдельно от последующего.

Кавычки делаются в русском тексте так: `<<текст>>`, в английском так: `‘text’`. Использовать символ " нельзя!

Многоточия в тексте и формулах делаются командой `\dots`.

Тире задается командой `"---"` (двойная кавычка и три минуса, окруженные пробелами): `Знание---сила`.

В длинных словах с дефисом, таких, как «счётно-аддитивно», дефис делается командой `"=`, иначе слово не будет переноситься: `счётно=аддитивно`. Команда `"~` запрещает перенос по дефису: `F-преобразование, F~пре\-образование`.

Неразрывный пробел `~` ставится между коротким предлогом и последующим словом, а также между очень короткой формулой и связанным с ней по смыслу словом: `число~N в~k~раз больше, чем~n`.

Между идущими подряд формулами иногда нужен дополнительный пробел:

<code>\$a=1,b=2\$</code>	<code>a = 1, b = 2</code>	— плохо
<code>\$a=1\$, \$b=2\$</code>	<code>a = 1, b = 2</code>	— получше
<code>\$a=1\$, \: \$b=2\$</code>	<code>a = 1, b = 2</code>	— хорошо
<code>\$a=1\$, \; \$b=2\$</code>	<code>a = 1, b = 2</code>	— хорошо

Иногда в формуле надо убрать пробелы вокруг знака операции. Например, если знак \times используется не как произведение, а для указания размеров матрицы или растрового изображения, то его лучше не окружать пробелами:

<code>\$640\times 480\$</code>	<code>640 \times 480</code>	— плохо
<code>\$640{\times}480\$</code>	<code>640\times 480</code>	— хорошо

Дополнительный пробел `\quad` рекомендуется вставлять между длинными выражениями, идущими через запятую в выключной формуле.

Короткий пробел `\,`, ставится после знака номера: `\No\,6`; в инициалах: `И.\,В.\,Анов`; в сокращениях: `т.\,к.; т.\,е.; и~т.\,д.`

Не следует использовать жирный шрифт для выделения *важных слов* или *терминов*. Это делается командой `\emph{текст}`.

Правила форматирования исходного кода облегчают его чтение и работу над корректурой:

- начинайте каждое предложение с новой строки;
- набирайте отдельной строкой команды `\begin`, `\end`, `$$`, `\[`, `\]`, `\section`, `\subsection`, `\paragraph`, `\item`, `\bibitem`, `\par`;
- внутритекстовые формулы, за исключением совсем коротких, набирайте отдельной строкой;
- длинные описания формул разбивайте на строки; используйте таблицы для выделения вложенных скобок и логически обособленных частей формул, как показано в Примере 1.

Пример 1. Форматирование сложной формулы:

$$R'_N(F) = \frac{1}{N} \sum_{i=1}^N \left(P(+1 | x_i) C(+1, F(x_i)) + P(-1 | x_i) C(-1, F(x_i)) \right).$$

```
\begin{align*}
R'_N(F)
= \frac{1}{N} \sum_{i=1}^N
\Bigl(
& P(+1 | x_i) C(+1, F(x_i)) \Bigr)
+ \Bigl(
& P(-1 | x_i) C(-1, F(x_i)) \Bigr).
\end{align*}
```

Ссылка на грант(ы), если она есть, задаётся в заголовке статьи командой `\thanks`. В конце статьи ссылаться на грант уже не нужно.

Список литературы

- [1] *Воронцов К. В.* $\text{\LaTeX} 2_{\epsilon}$ в примерах. — 2006. — <http://www.ccas.ru/voron/latex.html>.
- [2] *Гуссенс М., Миттельбах Ф., Самарин А.* Путеводитель по пакету \LaTeX и его расширению $\text{\LaTeX} 2_{\epsilon}$. — Москва: Мир, 1999. — 606 с.
- [3] *Котельников И. А., Чеботаев П. З.* $\text{\LaTeX} 2_{\epsilon}$ по-русски. — Новосибирск: Сибирский хронограф, 2004. — 489 с.
- [4] *Львовский С. М.* Набор и вёрстка в пакете \LaTeX . — 3-е издание. — Москва: МЦНМО, 2003. — 448 с.