

Магомедов А. М.

Учебное пособие по дискретной математике
(макетирование и просмотр)

Махачкала – 2014

Магомедов А. М.

Учебное пособие по дискретной математике (макетирование и просмотр). – Махачкала, 2014. – 32 с.

В учебном пособии рассмотрены особенности подготовки математических текстов монографического типа и вопросы электронного просмотра текста. Учебное пособие может быть полезно для подготовки дипломных работ, а также для оформления текстов диссертаций и монографий с использованием системы Latex.

Рецензент:

Лугуев Т.С. – доцент, канд. физ.-мат. наук

© Магомедов А. М. 2014

Оглавление

ВВЕДЕНИЕ	4
ЧАСТЬ I. ВЕРСТКА МАТЕМАТИЧЕСКОГО ТЕКСТА	8
Особенности подготовки больших текстов	8
Вставка с сохранением форматирования.....	8
Команды секционирования	11
Титульная страница. Нумерация страниц	11
Стиль страницы	12
Колонтитулы.....	13
Ссылки	15
Расстояние между строками	16
Ключевые термины для теорем, лемм и др.	17
Структура корневого файла и отдельных разделов монографии ...	18
ЧАСТЬ II. ОРГАНИЗАЦИЯ ЭЛЕКТРОННОГО ПРОСМОТРА.....	21
Формулировка задачи и предварительные разъяснения	21
Вопросы автоматизации решения подзадач	24
Параметры безопасности	29
Сложные элементы набора: формулы, рисунки и др.	29
ЗАКЛЮЧЕНИЕ.....	31
ЛИТЕРАТУРА	32

ВВЕДЕНИЕ

В соответствии с [1], *макетом* в полиграфии называют *предварительный образец книги* (журнала), изготавливаемый в издательстве для точного размещения текстового и иллюстративного материала. Макетом называется также *пробный экземпляр книги* или её переплёт, изготавливаемый в типографии до начала выпуска всего тиража. Для учебников и монографий по математике наиболее подходящей системой для изготовления макета считается Latex. Область применения настольной издательской системы Latex (в данном пособии использована ее распространенная разновидность MikTeX 2.9) простирается от подготовки математических записок объемом в полстранички (например, для набора текста экзаменационного билета, содержащего формулы) до создания математических монографий.

Невероятно, но факт: весьма серьезные специалисты относят систему обработки печатных документов TeX, первая версия которой создана в 1978 г. Д. Кнудом, к одному из важнейших достижений XX столетия [2]. Коль скоро существует обширная литература по компьютерной верстке математических текстов, актуальность написания пособия по верстке изданий монографического типа с применением данной системы требует разъяснений. В нашем случае разъяснения привязаны к конкретному контексту написания данного пособия (кафедра дискретной математики и информатики математического факультета ДГУ): качество набора и верстки дипломных работ выпускников математического факультета недостаточно высокое, то же можно сказать и о качестве набора некоторых монографий, изданных

специалистами факультета, а также о качестве текстов некоторых авторефератов диссертаций, размещенных на сайте ВАК учеными факультета. Однако пособие не призвано заменить существующие руководства по набору математических текстов; подобная задача для брошюры столь малого объема была бы заведомо непосильной. Акцент сделан на верстку больших документов типа монографий: разбивка текста на фрагменты с независимой отладкой, автособирание в целостную монографию, двусторонние поля, разбиение на главы и части, колонтитулы, преамбула, особенности создания оглавления, титульного листа, литературы и т.п. позволят специалисту, не обладающему предварительным опытом, сократить время на подготовку макета своей монографии или иного печатного документа.

Здесь же напомним, что печатным документом, по терминологии TeX'a, называется результат компиляции входного файла [2, с. 62] или, что почти то же, исходного текста. Исходный текст может содержаться в нескольких файлах, а корневой входной файл помимо исходного текста содержит обычно еще и преамбулу.

Согласно [1], монография (от *моно* и *графия*) – научный труд, в котором с наибольшей полнотой исследуется определенная тема. В ней обобщается и анализируется литература по данному вопросу, выдвигаются новые гипотезы и решения, способствующие развитию науки. Близкие толкования приняты и в других словарях. В [3] подчеркивается, что монография – научный труд в виде книги с углублённым изучением одной темы или нескольких тесно связанных между собой тем. Монография обычно сопровождается обширными

библиографическими списками, примечаниями и т. д. Создателями монографии может быть как один автор, так и целый коллектив, а сам термин отражает специфику рассматриваемой в ней проблематики, ее относительно узкую направленность, цельность.

Согласно Государственному стандарту по издательскому делу (ГОСТ 7.60-2003, п. 3.2.4.3.1.1) монографией является *«научное или научно-популярное издание, содержащее полное и всестороннее исследование одной проблемы или темы и принадлежащее одному или нескольким авторам»* [2]. Объем монографии не регулируется, но «классическим» считается объем текста свыше 120 страниц формата А4, написанных четырнадцатым размером шрифта Times New Roman через полуторный интервал, что примерно равно пяти печатным листам. Перед изданием в виде книги рукопись монографии должна пройти рецензирование специалистами по профилю монографии, имеющих ученые степени. Сведения о рецензентах указываются в выходных данных монографии. Считается, что рецензентов должно быть не менее двух. Если сведения о рецензентах не указаны в выходных данных книги, такое издание не считается научным [2].

Среди ученых принято завершать любую достаточно длительную работу по исследованию какой-то определенной темы публикацией соответствующей монографии, которая обычно содержит детальное описание методики исследования, изложение результатов проведённой работы, а также их интерпретацию.

Вопросы макетирования математических текстов рассмотрены в первой части данного пособия. Вторая часть посвящена вопросам ги-

пертекстового просмотра электронного издания в 3-фреймовом окне: в верхнем фрейме размещается набор ссылок на крупные единицы текста, например, главы; в левом-нижнем – список вложенных структурных единиц (параграфов) выбранной главы, в правом-нижнем фрейме отображается содержимое по выбранной в левом окне ссылке. Для одновременного изменения содержимого двух фреймов использованы скрипты. Однако для самостоятельной организации просмотра не требуется ни близкое знакомство с языками создания скриптов, ни даже детальное знание языка разметки гипертекста. Достаточно внимательно изучить простые примеры, приведенные в данном пособии.

ЧАСТЬ I. ВЕРСТКА МАТЕМАТИЧЕСКОГО ТЕКСТА

Особенности подготовки больших текстов

Для определенности отметим, что используется операционная система Windows 7.0, версия MiKTeX 2.9 и редактор TeXnicCenter.

Пусть составлен следующий набор команд, учитывающих тип документа «книга» (первая команда), буквальное воспроизведение символов текста (команда `\verb`), отображение русских букв (третья команда), а также задающих начало и конец документа.

Результат компиляции приведен ниже.

```
\documentclass{book}
\usepackage{verbatim}
\usepackage[russian]{babel}
\begin{document}
Для русскоязычного автора важно вывести текст,
набранный кириллицей. На данном этапе не обращаем
внимание на параметр \verb!{book}! Команды
\verb!\documentclass!.
\end{document}
```



1

Для русскоязычного автора важно вывести текст, набранный кириллицей. На данном этапе не обращаем внимание на параметр `{book}` команды `\documentclass`.

Вставка с сохранением форматирования

Если требуется вставить без форматирования не набор из нескольких символов, наподобие `{book}` или `\documentclass` из предыдущего упражнения, а фрагмент текста значительных размеров, со-

зданный в другой среде (например, в Delphi), то вместо команды `\verb` удобно применять команду

```
\begin{verbatim} ... \end{verbatim}.
```

К тому же, чтобы подготовить текст на русском языке, рекомендуется наряду с пакетом `babel` (с опцией `russian`) загрузить и пакет `indentfirst`. Пакеты также видят и используют «свои» опции (необязательные параметры), перечисленные непосредственно в `\documentclass`, поэтому можно записать:

```
\documentclass[russian,a4paper]{book}%{article}
\usepackage{babel, verbatim, indentfirst}
\usepackage[cp1251]{inputenc}%кодировка исходного текста

\begin{document}
\begin{verbatim}
uses crt;
var
  i, s: integer;
begin
  s:=0;
  for i:=1 to 100 do s:=s+i;
  WriteLN ('s= ', s);
  readkey
end.
\end{verbatim}
\end{document
```



```

uses crt;
var
  i, s: integer;
begin
  s:=0;
  for i:=1 to 100 do s:=s+i;
  WriteLN ('s= ', s);
  readkey
end.

```

Стандартными являются 6 классов: `article`, `book`, `report`, `letter`, `proc`, `slides`. Они могут иметь не только те две опции, которые указаны в нашем примере. Например, `10pt`, `11pt`, `12pt` задают базовый шрифт (т.е. `\normalsize`) равным 10, 11 или 12, по умолчанию используется `10pt`.

Опции `oneside` или `twoside` форматируют документы для печати на одной или обеих сторонах листа; во втором случае четные и нечетные страницы по-разному размещаются на листах для облегчения их брошюровки. По умолчанию используется `oneside` за исключением класса `book`, где по умолчанию предусмотрена двусторонняя печать.

Поскольку мы собираемся использовать класс `book`, то добавим также, что опции `openright` или `openany` указывают соответственно, что главы должны начинаться на правой странице или же на любой. Эти опции применяются только с классами `report` (где по умолчанию действует опция `openany`) и `book` (где действует `openright`), так как главы, начинающиеся с команды секционирования `\chapter`, имеются только в командах этих двух классов.

Команды секционирования

Отметим и следующие команды:

`\subsection{обычный текст}` %печатает заголовок раздела,
`\subsubsection{математический текст}`%заголовки подразделов.

В классе `book` доступны секции `\chapter`, `\section`, `\subsection`, `\subsubsection`, `\paragraph`, `\subparagraph` или те же с последующим символом «*» – тогда команды только печатают заголовок, но не записывают ничего в оглавление и в колонтитулы и не увеличивают счетчик нумерации разделов. Они могут иметь [необязательный параметр], который заносится в оглавление и колонтитулы, и {обязательный параметр} – текст заголовка.

Титульная страница. Нумерация страниц

Параметры `titlepage` или `notitlepage` изменяет действие команды `\maketitle`, печатающей заголовок, список авторов и дату, и процедуры `\abstract`, печатающей аннотацию к документу. Если действует опция `titlepage`, то заголовок и аннотация размещаются на отдельных страницах, в противном случае – непосредственно перед основным текстом. По умолчанию `titlepage` используется для всех классов, кроме `article`.

Чтобы получить полный контроль над форматом и содержанием титульной страницы, лучше всего использовать процедуру

$$\begin{titlepage} \dots \end{titlepage},$$

которая подготавливает чистую страницу и присваивает ей непечатаемый номер 1. Так сказано в [2], но в практике автора номер все-таки выводился и для его подавления пришлось использовать команду `\thispagestyle{empty}`. Более подробно:

```

\newpage
\thispagestyle{empty}
\centerline{ФГБОУ <<Дагестанский государственный университет>>}
\par\medskip
\vspace{6cm}
\centerline{\bf А.\,М.\,Магомедов}
\vspace{2cm}
\centerline{\bf ЖЕСТКАЯ ОПТИМИЗАЦИЯ РАСПИСАНИЯ}
\par\medskip \centerline{\it {Учебная монография}}
\vspace{10cm}
\centerline{-----}
\centerline{МАХАЧКАЛА -- 2014}

```

Как видим, сведения о стиле страницы требуют немало внимания.

Стиль страницы

Как сказано в [2, с. 399], стиль страницы – полосы набора, установленный классом документа, изменяют декларации

`\pagestyle{аргумент}` и `\thispagestyle{аргумент}`.

Область действия первой декларации начинается со следующей страницы, вторая воздействует только на текущую страницу. Аргумент может принимать четыре значения:

`plain` – используется по умолчанию стандартными классами кроме `book` и печатает номера страниц в нижнем колонтитуле, а верхний колонтитул остается пустым;

`empty` – оба колонтитулы пусты, поддерживается нумерация страниц, но номера не печатаются;

`headings` – используется классом `book` по умолчанию; печатает номера страниц и другую информацию, определяемую классом документа (обычно заголовки разделов), в верхнем колонтитуле, нижний остается пустым;

`myheadings` – содержание верхнего колонтитула должен определять сам автор, используя команды `\markboth` и `\markright` из следующего раздела, а нижний колонтитул пуст.

Примечание. Обычно декларацию `\pagestyle` вставляют после команд типа `\chapter`, которые начинают печатать текст с новой страницы (это потому, что содержание колонтитулов определяют в конце форматирования страницы).

Колонтитулы

При выборе стиля страницы `headings` или `myheadings` используются декларации `\markboth{left}{right}` или `\markright{right}` соответственно. Аргументы соответствуют информации на левую и правую страницы, причем при односторонней печати все страницы считаются правыми. Например, при классе документа `book`: при двустороннем стиле печати для команд секционирования `\chapter` и `\section` название раздела в колонтитулы заносят декларации `\markboth` (устанавливает пустой колонтитул для правой страницы) и `\markright` соответственно, а при односторонней печати – только команда секционирования `\chapter` с помощью `\markright`. Если у нас односторонняя печать, то, как было сказано, все страницы считаются правыми, следовательно, по умолчанию колонтитулы будут пустыми.

Правило 1. Чтобы переопределить эти установки, необходимо поместить `\markboth` сразу после команды секционирования, а `\markright` непосредственно перед и после команды секционирования, но первая команда должна быть опущена, если команда секционирования начинает новую страницу.

Правило 2. Если требуется, чтобы очередная глава не имела номера, ее следует начать с команды `\chapter*`. Но при этом и название главы не заносится в верхние колонтитулы. Чтобы глава не сопровождалась номером, но название главы печаталось в верхние колонтитулы, применяется команда `\markboth`. Приведем два варианта.

Первый правильный вариант: в `\documentclass[... , twoside]{...}`

```
\chapter{Это название главы }
\pagestyle{myheadings}
\markboth{Это название раздела }{Это название раздела}
\section{Это название параграфа}
```

Результат: название главы заносится в колонтитул четных страниц, колонтитулы нечетных пусты.

Второй правильный вариант: в `\documentclass[... , oneside]{...}`

```
\chapter{Это название главы }
\pagestyle{headings}
\section{Это название параграфа}
\markright{Это название раздела}
```

Результат: название главы заносится в колонтитул четных страниц, колонтитулы нечетных пусты.

Рекомендуем обратить внимание и на следующий вариант. В преамбуле написано:

```
\documentclass[14pt,reqno,intlimits, twoside] {rusthesis}
\chapter{Название главы}
\pagestyle{headings}
\newcommand{\eqdef}{\stackrel{\mathrm{def}}{=}}
\markboth{Текст 1}{Текст 2}
% Первый текст выводится на четные страницы, в конец
\section {Название параграфа}
% выводится на нечетные страницы, с начала
```

Результат: текст 1 выводится на четные страницы с выравниваем вправо, текст 2 игнорируется, название параграфа выводится на нечетные страницы с выравниванием влево.

На каждой странице главы, независимо от других разделов, выводится колонтитул, указанный после `\markright`. Если при этом вместо `oneside` указать `twoside`, то на четных страницах указывается название главы, на нечетных – названия разделов, т.е. `\markright` игнорируется.

Если сохранить `twoside` в преамбуле, а в каждой главе сохранить первые две строки, как они написаны выше, но при этом `\markright {}` заменить на `\markboth{левый текст}{правый текст}`, то правый текст игнорируется, на четных страницах печатается название главы, для нечетных же выполняется правило: «если `\markboth` имеется, то печатается его левый текст, если отсутствует, то – название раздела (`section`)».

Интересно, что если непосредственно перед командой `\section{}` набрать:

```
\markboth{Краткий левый колонтитул}{Правый колонтитул}
\section{Название секции},
```

то всюду в пределах секции нечетным колонтитулом становится левый колонтитул, а четным – не название главы, а правый колонтитул. Отметим, что при этом в нашем примере в классе документа было указано `twoside`, но вместо `book` указано `rusthesis`.

Ссылки

Зачем нумеруют разделы? Прежде всего – для удобства ссылок: с помощью команды `\label{текст}` помечают раздел, команда `\ref{текст}`

метки} печатает номер помеченного раздела, а команда `\pageref{текст метки}` печатает ссылку на номер страницы, на которую попадает метка. Пакет `varioref` усиливает возможности ссылок; например, команда `\vref{текст метки}` печатает ссылку, как и `\ref{текст метки}`, добавляя при этом «на с. 25», «на предыдущей странице», «на следующей странице».

Расстояние между строками

Десятичное число, равное величине межстрочного интервала (интерлиньяжа) задается с помощью `\baselinestretch` и по умолчанию равно 1 ([2, с. 404]). Это значение изменяется декларацией `\renewcommand`. Например, размещение в преамбуле входного файла команды

```
\renewcommand{\baselinestretch}{1.25}
```

приведет к тому, что весь печатный документ будет напечатан через 1,5 интервала, двум интервалам соответствует значение 1.67, половине интервала – 0.75.

Если же нужно изменить интерлиньяж в каком-либо фрагменте, то надо написать:

```
{%открывающая фигурная скобка необходима
```

```
\renewcommand{\baselinestretch}{1.25}
```

```
\selectfont % в виде эксперимента эта команда была пропущена нами, но нужная цель при этом достигается
```

```
.... Текст, в котором надо изменить интерлиньяж ...
```


Высота и ширина текста можно задавать следующим образом:

```
\textheight=число и \textwidth=число. Команды
```

`\topmargin=число`, `\oddsidemargin=число`, `\evensidemargin=число`, задающие соответственно верхнее поле и левые поля для нечетных и четных страниц, нужно интерпретировать следующим образом. Фактические значения полей получаются прибавлением одного дюйма к указанным числам: поле отсчитывается не от края листа, а от линии, параллельной краю листа и отстоящей от него на 1 дюйм. Таким образом, если в преамбуле документа задать команду `\oddsidemargin=0pt`, то отступ от края листа будет составлять 2,54 см. Для того, чтобы задавать отступы от края листа менее 2,54 см, необходимо использовать отрицательные значения.

О ширине отступа в начале абзаца и других элементах оформления страницы см. [2, с. 402-405]. Подчеркнем, что правое поле задается только косвенно – заданием ширины текста и ширины левого поля. Например, для задания одинаковых левых полей при двусторонней печати (опция `twoside` в `\documentclass`) в конце преамбулы можно написать:

```
\textwidth=16.5cm
\oddsidemargin=-0.2cm
\evensidemargin=-0.2cm
```

Ключевые термины для теорем, лемм и др.

Некоторые редакции математических журналов требуют применения своих стилевых пакетов. Например, журнал «Прикладная дискретная математика» предлагает пакет `adm.sty`, которая поддерживает в частности следующие декларации для утверждений, следствий и т.п.

```
\begin{State}\label{label1} \end{State}
```

```

\begin{Proof}                                \end{Proof}
\begin{Corollary} \label{Cor1}              \end{Corollary}
\begin{Theorem}                               \end{Theorem}
\footnotetext{Работа поддержана грантом ... }
\begin{abstract} ... \keywords{... }        \end{abstract}

```

Представление о других поддерживаемых командах дает следующее перечисление, которое мы скопировали из `adm.sty`: `section`, `equation`, `table`, `figure`, `algorithm`, `footnote`, `Theorem`, `Lemma`, `Corollary`, `Def`, `Hypothesis`, `Problem`, `Example`, `Remark`, `State-rm`, `State`, `Definition`.

Структура корневого файла и отдельных разделов монографии

Печатный текст значительного размера (например, монографии) не только обладает титульным листом, оглавлением, основным текстом, заключением и списком литературы, но при этом требуется, чтобы основной текст был секционирован, обладал нумерацией страниц и колонтитулами. Сказанное относится к печатному документу. Но существуют и вопросы, относящиеся к облегчению процесса преобразования исходного документа к печатному.

Один из вариантов таков: вначале командой `\input` вставляется файл преамбулы, затем внутри команд

```
\begin{document} ... \end{document}
```

выполняется создание титульной страницы (команда `\maketitle`) и оглавления (команда `\tableofcontents`), затем командами `include{название}` последовательно вставляются введение, главы, заключение и литература.

Для выполнения отладки текста по частям рекомендуется после вставки преамбулы использовать команду `\includeonly` с указанием в параметрах тех частей текста, которые включаются в компоновку текста, при этом сохраняется глобальная нумерация страниц. Вместо этого можно ограничиться использованием только команд `\input`, комментируя те из них, которые не нужны на данном этапе. Надо иметь в виду, что команда `\include`, в отличие от `\input`, вызывает переход на новую страницу; поэтому рекомендуется использовать прежде всего для включения глав. Таким образом, один из вариантов структуры корневого файла следующий.

```
\input{Mypreambula}
\includeonly{chapter0, chapter1, chapter2, conclusion, chapterlit}
\begin{document}
```

```
\begin{titlepage}
формирование титульной страницы
\end{titlepage}
```

```
\tableofcontents
```

```
\include{chapter0} \include{chapter1} \include{chapter2}
\include{conclusion} \include{chapterlit}
\end{document}
```

Глава может иметь, например, структуру:

```
\chapter{Построение и жесткое уплотнение расписания}
\newcommand{\eqdef}{\stackrel{\mathrm{def}}{=}}
Для расписаний различных типов рассмотрены ...
\endinput
```

Расположенная в корневом файле (см. выше) команда `\tableofcontents` включит параметр команды `\chapter` в оглавление после слова «ГЛАВА», за которым следует ее номер. Если включить в оглавление и «Введение» (или «Заключение») тем же способом – располагая в его

начале команду `\chapter`, то в оглавлении и в тексте введение будет помечено неуместным термином «ГЛАВА», поэтому можно поступить следующим образом, подавляя данный термин, но включая пункт «Введение» в оглавление отдельной командой `\addcontentsline`.

Введение может иметь, например, структуру:

```
\chapter*{Введение}
\addcontentsline{toc}{chapter}{Введение}
\vspace{0.5cm}
\centerline{ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ}
\par\medskip
{\bf Актуальность темы}.
ТЕКСТ ВВЕДЕНИЯ
\par\medskip
Автор выражает признательность ...
\endinput
```

ЧАСТЬ II. ОРГАНИЗАЦИЯ ЭЛЕКТРОННОГО ПРОСМОТРА

Формулировка задачи и предварительные разъяснения

Рассмотрим специфические особенности организации гипертекстового просмотра глав и параграфов монографии, учебника и т.п. в окне из трех фреймов. Заметим, что организация просмотра структуры вида «Книги – названия произведений – тексты произведений» и иных близких структур полностью аналогична организации просмотра структуры «Главы учебника – названия параграфов – тексты параграфов».

Верхний фрейм (обозначим его через А) отведем под наименования глав, всюду в дальнейшем количество глав обозначено через n . При выборе главы требуется отобразить в левом фрейме L список ссылок – названий параграфов выбранной главы, а при выборе ссылки из этого списка – вывести в фрейм R текст соответствующего параграфа (см. рис. 1).

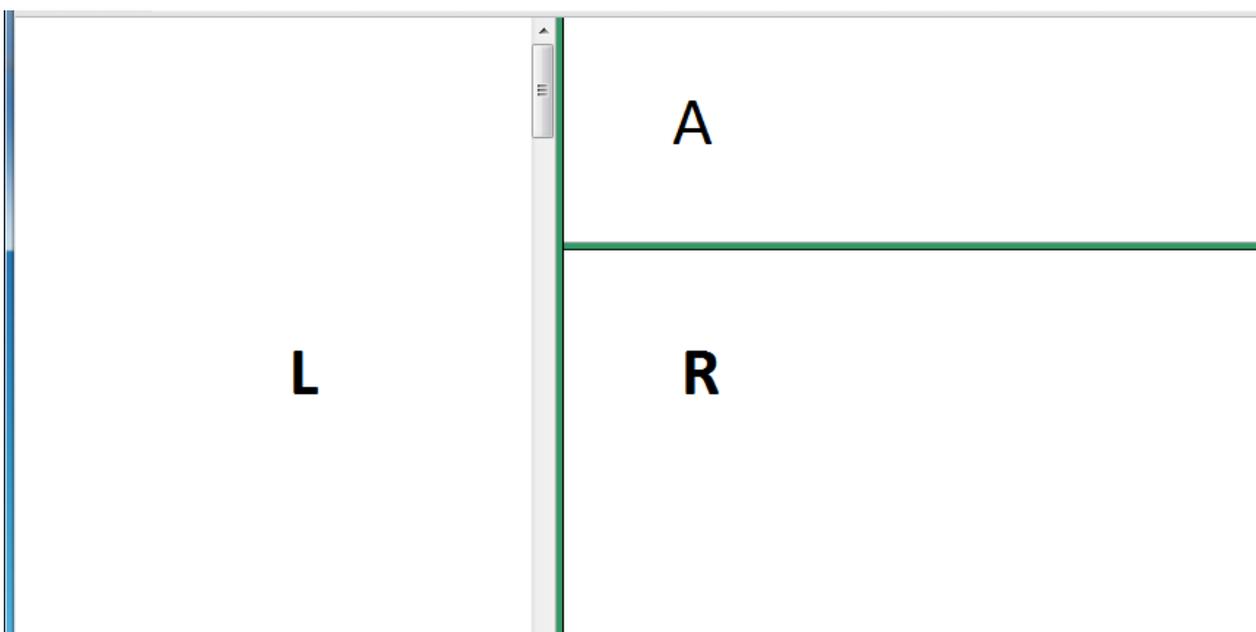


Рис. 1. Расположение фреймов.

Пусть имеются три гипертекстовых файла: A.htm, L1.htm и R1.htm. Если подготовить (например, с помощью текстового редактора Блокнот) файл main.htm с приведенным ниже текстом, то просмотр файла main.htm с помощью браузера приведет к отображению содержимого этих трех файлов в «соответствующих» фреймах окна браузера.

```
<html>
<frameset cols="1, 3">
  <frame name="L" src="L1.htm">
  <frameset rows="24%, 1*">
    <frame name="A" src="A.htm">
    <frame name="R" src="R1.htm">
  </frameset>
</frameset>
</html>
```

Текст файла main.htm.

Мы ограничились краткими именами файлов в предположении, что эти файлы расположены в той же папке, что и main.htm. О создании фреймов с помощью тегов <Frameset> и <Frame> см., например, [5].

Приведем пример текста файла A.htm.

```
<html>
<body>
<a href="#" name="Comb"> Комбинаторика </a>;
<a href="#" name="Graph"> Графы </a>
</body>
</html>
```

Текст файла A.htm.

Текст файла R1.htm должен быть структурирован таким образом, чтобы при выборе *i*-й ссылки в фрейме L в фрейме R отображался

ся текст i -го раздела (параграфа) файла R1.htm. Стандартным решением является теговое оформление наименования каждого параграфа с тем, чтобы, во-первых, каждый параграф был доступен по тому или иному имени (в нашем подходе в качестве имени взято число), во-вторых, наименование параграфа при просмотре выделялось нужным стилем:

```
<h2><a name=i> Наименование раздела </a> </h2>.
```

Здесь i выступает в качестве символа десятичной записи номера раздела, фактически же подразумевается константа: 1, 2, ... Приведем пример.

```
<html>
<body>
<h2><a name=1> 1. Определения </a> </h2>
.....
<h2><a name=2> 2. Две задачи комбинаторики</a> </h2>
.....
<h2><a name=3> 3. Размещения</a> </h2>
.....
</body>
</html>
```

Текст файла R1.htm.

В качестве текста файла L1.htm можно выбрать набор тегов вида

```
<a href="R1.htm#i" target=R1><span style='color:blue'> Наименование
раздела </a><br>.
```

Здесь остаются в силе приведенные выше разъяснения относительно символического использования i . При выборе соответствующей ссылки текст файла R1.htm, начиная с раздела, номер которого приведен после знака '#', отобразится в фрейме R.

Приведем пример.

```
<html>
<body>
<a href="R1.htm#1" target=R><span style='color:blue'>
1. Определения </a><br>
<a href="R1.htm#2" target=R><span style='color:blue'>
2. Формулировка двух задач комбинаторики </a><br>
<a href="R1.htm#3" target=R><span style='color:blue'>
3. Размещения без повторений </a><br>
</body>
</html>
```

Текст файла L1.htm.

При наших схематических текстах результат имеет вид, изображенный на рис. 2.

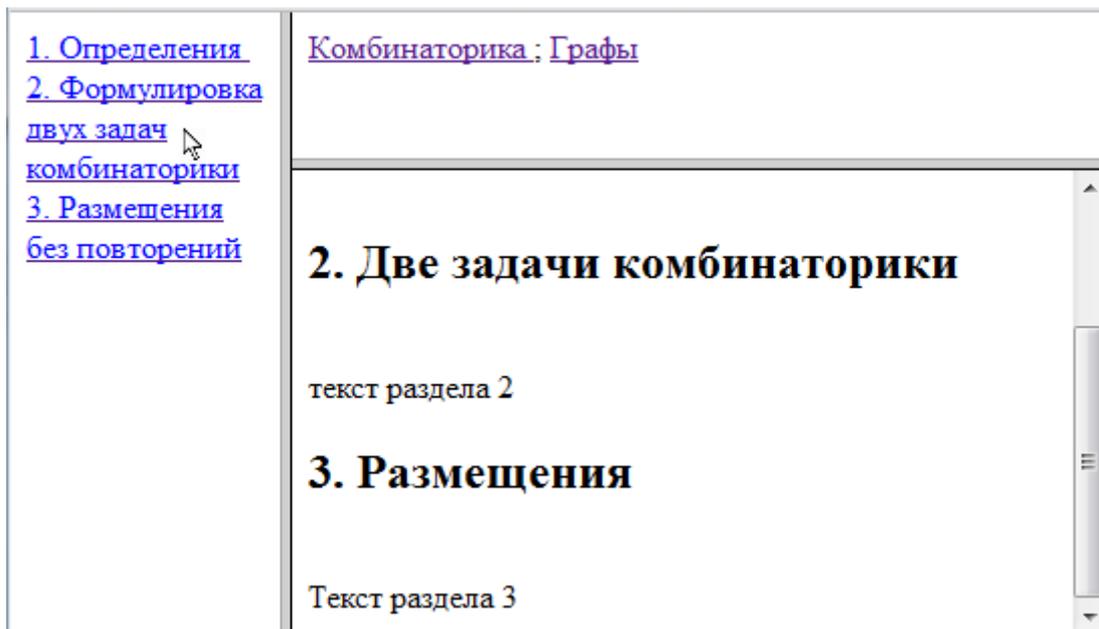


Рис. 2. Результат просмотра файла main.htm. В фрейме L выбрана вторая ссылка.

Вопросы автоматизации решения подзадач

До сих пор рассматривался лишь вопрос просмотра материала первой из глав, перечисленных в фрейме A: хотя мы и не подчеркивали это обстоятельство явно, использовались лишь файлы L1.htm и

R1.htm, относящиеся к первой главе. Понятно, что для просмотра материала второй, третьей и т.д. глав необходимо подготовить для них аналогичные файлы, назовем их L2.htm и R2.htm, L3.htm и R3.htm и т.д. Необходимо, но недостаточно. Ведь пока непонятен алгоритм определения нужного содержимого фреймов L и R при выборе произвольной главы в фрейме A. Более того, алгоритм не был сформулирован даже применительно к выбору первой главы в фрейме A, а предшествующий «успех» был предопределен исходным текстом файла main.htm, который носит статический характер и определяемые им действия – одни и те же при выборе любой ссылки в списке фрейма A. Понятно, что эти действия должны быть привязаны к выбору той или иной ссылки: «Комбинаторика», «Графы» и т.д. Соответствующая привязка осуществляется ниже с помощью обработчика событий – щелчков мышью по той или иной ссылке. Для краткости приведем решение для случая двух глав, обработка выбора третьей, четвертой и других глав выполняется идентично.

```
<html>
<head>
<script language="VBScript">
sub Comb_onclick()
  parent.frames("L").location.href="L1.htm"
  parent.frames("R").location.href="R1.htm"
end sub

sub Graph_onclick()
  parent.frames("L").location.href="L2.htm"
  parent.frames("R").location.href="R2.htm"
end sub
</script>
</head>
```

```

<body>
<a href="#" name="Comb"> Комбинаторика </a>;
<a href="#" name="Graph"> Графы </a>
</body>
</html>

```

Усовершенствованный текст файла A.htm для случая двух глав.

Задача организации просмотра естественным образом разбивается на подзадачу подготовки гипертекстов и подзадачу организации просмотра с применением скриптов для автоматического изменения содержимого фреймов в L при выборе главы в фрейме A с отображением в фрейме R содержимого параграфа, ссылка на который выбрана в списке из фрейма L. Решение второй подзадачи – одновременного изменения содержимого нескольких фреймов при выборе гиперссылки в другом фрейме (в нашем случае – при выборе гиперссылки "Комбинаторики" и др. в фрейме A) приведено выше. Оно достигается (см. [6]) использованием скрипта – небольшой программы, которую можно, например, расположить в заголовке html-документа (т.е. внутри тегов <head> и </head>).

Скрипт может использовать нумерацию фреймов: Frames (0) – в нашем случае левый фрейм (он был создан раньше других, см. main.htm), Frames (1) и Frames (2) – правый верхний и правый нижний фреймы соответственно. Более точно, Frames – это коллекция рамок родительского для нашего окна объекта Parent. Для обращения к фреймам можно использовать не только индексы, но и имена: Parent.Frames ("L"), Parent.Frames ("A") и Parent.Frames ("R").

Осталось рассмотреть подзадачу подготовки гипертекстов L_i , R_i , $i=1, 2, \dots, n$.

При небольших размерах текста и малого количества ссылок подготовка гипертекста «вручную» не представляет трудности. Но автору приходилось встречаться с ситуацией, когда количество ссылок измеряется сотнями (например, при подготовке программы гипертекстового просмотра полного собрания сочинений из 10 томов [7]). В таких ситуациях вполне надежное выполнение действий по вставке в текст соответствующих закладок в ручном режиме представляется маловероятным, и вопросы определенной автоматизации составления гипертекстов приобретают определенную остроту.

Из соображений краткости изложения особенности использования специализированных редакторов создания гипертекстов здесь не рассматриваются.

С целью сосредоточиться на элементах автоматизации подготовки материала, до конца данного раздела будем исходить из предположения о простом формате текста (без формул, рисунков и т.п.), к тому же будем полагать, что текст учебника подготовлен в формате документа Word. Сначала для первой главы составим список ссылок – названий параграфов: выделим все названия стилем "Заголовок", применим команду меню "Ссылки-оглавление" MS Word 2013 для обособления списка названий, после чего вырежем данный список в отдельный файл "Names1" в виде обычного текста, а текст самой главы скопируем в файл "Text1" с сохранением принятого в оригинале разбиения на строки.

В программе на языке Delphi с объектами S, T и V типа TStringList выполним следующие шесть действий.

1. Загрузим в S список названий параграфов из файла "Names1". Далее через NS обозначено количество строк в списке S .

2. Для $i=0, 1, \dots, NS-1$ выполним запись в $T[i]$ гиперссылки на i -ю закладку документа $R1.htm$ с целью отображения ее в фрейме R , другими словами, $T[i]$ присвоим строковое выражение:

```
'<a href="R1.htm#'+ IntToStr (i)+ "' '+target=R>'+ IntToStr (i)+'.' '+ S[i]+'</a><br>'
```

3. Вставим в начало и конец списка T фрагменты `<html><body>` и `</body></html>` соответственно и сохраним список T в файле "L1.htm".

Создание файла $L1.htm$ завершено. Создадим файл $R1.htm$.

4. Загрузим в объект V список из файла "Text1".

5. Для каждого $i=0, 1, \dots, NS-1$ найдем в списке V строку, содержащую $S[i]$ (совпадение текстов рассматривается с точностью до регистра букв), номер найденной строки обозначим через k и выполним замену $V[k]$ на строковое выражение:

```
'<h2><a name='+ IntToStr(i) +'> '+ IntToStr(i) + '.' '+S[i]+' </a> </h2>'
```

Здесь первое из двух представлений числа i является именем закладки, второе служит отображаемым номером параграфа с названием $S[i]$.

6. К каждой строке из V добавим тег `
` (переход на новую строку) для сохранения принятого ранее способа разбиения на строки. Затем вставим `<html> <body>` и `</body> </html>` соответственно в начало и конец списка V и сохраним список строк V в файле $R1.htm$.

Аналогично создаются файлы $L2.htm$, $R2.htm$ и др.

Параметры безопасности

В зависимости от настроек браузера на компьютере, двойной щелчок по пиктограмме файла main.htm вместо ожидаемого открытия окна с фреймами для просмотра пособия может инспирировать сообщение о запрете использования скриптов. В этом случае для изменения настроек параметров безопасности следует открыть диалоговое окно «Свойства браузера», перейти на вкладку «Безопасность», где нажмем кнопку «Другой...» и в открывшемся окне «Параметры безопасности» выберем соответствующие разрешительные опции.

Предложим иной способ обойти запреты без изменения настроек на том или ином компьютере. Построим проект Delphi: разместим на форме компоненту WebBrowser1 из VCL (страница Internet), объявим переменную path строкового типа и определим обработчик события OnCreate для формы определим следующим образом.

```
procedure TForm1.FormCreate(Sender: TObject);
begin
  path:= GetCurrentDir+'\main.htm';
  if Fileexists (path) then
    webBrowser1.navigate (path)
  else
    showMessage('Не найден main.htm');
end;
```

Остается разместить исполняемый файл в той же папке, что и main.htm, затем запустить.

Сложные элементы набора: формулы, рисунки и др.

С некоторой долей условности можно принять бытующее представление, что набор текста учебника по дискретной математике в целом проще нежели набор текста учебника, например, по математи-

ческому анализу. Электронное издание обычно рассматривается как совокупность графической, текстовой, цифровой, формульной, табличной документации (иногда сюда добавляется речевая-, видео- и фото- информация). Как известно, в MS Word имеются развитые средства создания соответствующих фрагментов, в версии MS WORD 2010 и 2013 включены дополнительные средства набора математических формул. Сохранение соответствующего документа в формате mht – «Веб-страница в одном файле» и использование результата вместо примененных выше htm-файлов выглядит предпочтительнее. Другой подход заключается в сохранении фрагментов, предназначенных для отображения в фрейме R, в pdf-формате вне зависимости от среды их создания (как известно, MS Word также допускает сохранение документа в формате pdf). В этом случае выигрывается качество текста, но приходится разрезать R.pdf на множество мелких фрагментов.

ЗАКЛЮЧЕНИЕ

Если материал Части I в одинаковой мере может быть использован специалистами разных направлений математики, то рекомендации по электронному просмотру, приведенные в Части II, в большей мере рассчитаны на математические (и иные) издания, где текстовая составляющая является доминирующей.

В Части I предполагается, что читатель владеет основами системы компьютерного набора небольших математических текстов (типа заметок, статей). Однако уверенность, что набор и верстка текста учебника или монографии объемом в 200-300 страниц отличаются лишь бóльшим количеством набираемых символов, улетучивается при первом же столкновении с реальностью. И поиски решений возникающих проблем макетирования и верстки занимают время, вполне соотносимое со временем, которое автор расходует на набор собственно текста. На сокращение этих поисков и направлены изложенные в Части I сведения и рекомендации.

Из Части II видно, что организация электронного просмотра включает две основные подзадачи, одна из которых заключается в частичной автоматизации процесса подготовки гипертекстового материала, другая – в удобной визуализации материала с использованием элементов скриптов. При этом предварительные знания о языках программирования желательны, но не относятся к разряду необходимых.

Автор благодарит рецензента Т.С. Лугуева за ценные замечания и рекомендации.

ЛИТЕРАТУРА

1. Большая советская энциклопедия: В 30 т. – М.: "Советская энциклопедия", 1969-1978.
2. Котельников И.А., Чеботаев П.З. LATEX по-русски. – 3-е издание, перераб. и доп. – Новосибирск: Сибирский хронограф, 2004. – 496 с.
3. Свободная википедия (статья «Монография») // способ доступа: <http://ru.wikipedia.org/wiki>.
4. Антонова С.Г., Васильев В.И., Жарков И.А., Коланькова О.В., Ленский Б.В., Рябина Н.З., Соловьев В.И. Редакторская подготовка изданий: Учебник // Под общ. ред. Антоновой С.Г. – М.: Издательство МГУП, 2002. – 468 с.
5. Якушина Е. Изучаем Интернет, создаем Web-страничку. 2-е издание – СПб: Питер, 2003. – 256 с.
6. Армстронг Т. ActiveX: Создание Web-приложений. Пер. с англ. – К.: Издательская группа BHV, 1999. – 512 с.
7. Магомедов А.М., Магомедов Т.А., Магомедов М.А. Мультимедийное прочтение А.С. Пушкина (программа) // Свидетельство № 2010611946 о гос. регистрации программы для ЭВМ от 15.03.2010 в Реестре программ для ЭВМ.