МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ВОЛЖСКИЙ ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ (ФИЛИАЛ)
ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО ОБРАЗОВАТЕЛЬНОГО
УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОЛГОГРАДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

М. А. Маслова

Учебно-методическое пособие по лабораторным работам по дисциплине «Технология подготовки ВКР»

Электронное учебное пособие

Волжский

2024

ББК 74.026.65я73
УДК 004(07)
М 317

Рецензенты:
заведующий кафедрой методики преподавания математики и физики, ИКТ ФГБОУ ВО ВГСПУ, профессор, д.п.н.
Смыковская Т.К.,
ведущий программист ООО «Инженеры информации»
Ридель А.В.

Издается по решению редакционно-издательского совета
Волгоградского государственного технического университета

Маслова, М. А.
Учебно-методическое пособие по лабораторным работам по дисциплине «Технология подготовки ВКР» [Электронный ресурс] : учебное пособие / М. А. Маслова ; Министерство науки и высшего образования Российской Федерации, ВПИ (филиал) ФГБОУ ВО ВолгГТУ. – Электрон. текстовые дан. (1 файл: 2,7 МБ). – Волжский, 2024. – Режим доступа: http://lib.volpi.ru. – Загл. с титул.экрана.
ISBN 978-5-9948-4851-7

Учебное пособие содержит изложение работе с системой Latex в рамках курса «Технология подготовки выпускной квалификационной работы». Предназначено для студентов высших учебных заведений, обучающихся по направлению (специальности) 09.03.01 «Информатика и вычислительная техника» и 09.03.04 «Программная инженерия», а также других технических специальностей.

Ил. 33, табл. 4, библиограф.: 5 назв.

ISBN 978-5-9948-4851-7

Содержание

Введение

Предлагаемый лабораторный практикум предназначен для студентов направления 09.03.01 «Информатика и вычислительная техника» и 09.03.04 «Программная инженерия». Цель данного руководства – закрепление теоретических знаний по курсу «Технология подготовки выпускной квалификационной работы».

В научной IT-сфере при написании различных работ зачастую необходимо прибегать к математическому аппарату. Математический аппарат удобнее оформлять при помощи системы Latex. К тому же многие крупные научные издательства используют Latex, предоставляя авторам файл со стилями.

Настоящее электронное учебное пособие имеет целью формирование у бакалавра необходимых знаний, касающихся общих представлений и принципов работы с издательской системой Latex, с последующим применением полученных знаний на практике. Основными задачами учебного пособия являются: помощь студентам в изучении и применении основ работы издательской системой Latex.

В состав лабораторных работ входит теоретическая часть, порядок выполнения работы, варианты заданий и контрольные вопросы.

Лабораторная работа №1. Введение в LATEX

1. Основные понятия и синтаксис LATEX

TEX (произносится «тех», пишется также «TeX») – это созданная замечательным американским математиком и программистом Дональдом Кнутом (Donald E.Knuth) система для верстки текстов с формулами. Сам по себе TEX представляет собой специализированный язык программирования (Кнут не только придумал язык, но и написал для него транслятор, причем таким образом, что он работает совершенно одинаково на самых разных компьютерах), на котором пишутся издательские системы, используемые на практике. Точнее говоря, каждая издательская система на базе TEXa представляет собой пакет макроопределений (макропакет) этого языка. LATEX (произносится «латех» или «лэйтех», пишется также «LaTeX») – это созданная Лесли Лампортом (LeslieLamport) издательская система на базе TEXa.

Прежде, чем углубиться в изучение собственно LATEXa, скажем несколько слов о других издательских системах на базе TEXa. Наряду с LATEXом, распространены также макропакеты PlainTEX и AMS-TEX. Макропакет Plain TEX был разработан самим Дональдом Кнутом, рассматривавшим его в качестве платформы для построения более сложных систем; на практике он используется и как средство для обмена текстами (текст, подготовленный для PlainTEXa, сравнительно несложно переделать в исходный текст для того же LATEXa). Что касается AMS-TEXa, то эта издательская система сориентирована на важный, но узкий круг приложений: верстку статей для математических журналов, издаваемых Американским Математическим Обществом. Соответственно, в AMS-TEXe предусмотрено большое количество весьма изощренных возможностей для создания сложных математических формул, но при этом нет многих вещей, которые естественно было бы ожидать в издательских системах общего назначения (например, автоматической нумерации частей документа).

LATEX в этом отношении более гармоничен. Наконец, недавно появился AMS-LATEX, призванный сочетать мощь LATEXа как издательской системы и изощренные возможности набора формул, предоставляемые AMS-TEXом. Эта система находится в процессе становления, и писать о ней книгу пока преждевременно.

Все издательские системы на базе TEXа обладают достоинствами, заложенными в самом TEXе. Для новичка их можно описать одной фразой: напечатанный текст выглядит «совсемкак в книге». LATEX, как издательская система, предоставляет удобные и гибкие средствадостичь этого книжного качества. В частности, указав с помощью простых средств логическую структуру текста, автор может не вникать в детали оформления, причем эти детали при необходимости нетрудно изменить (чтобы, например, сменить шрифт, которым печатаются заголовки, не надо искать по всему тексту, меняя все заголовки, а достаточно заменить одну строчку в «стилевом файле»). Такие вещи, как нумерация разделов, ссылки, оглавление и т.п., получаются почти что сами собой.

Огромным достоинством систем на базе TEXа является высокое качество и гибкость верстки абзацев и математических формул (в этом последнем отношении TEX до сих пор не превзойден).

TEX (и все издательские системы на его базе) неприхотлив к используемой технике: им вполне можно пользоваться, например, на компьютерах на базе 80286-процессора, 1, а исходные тексты можно готовить и на совсем уж примитивных машинах. С другой стороны, TEX-овские файлы обладают высокой степенью переносимости: Вы можете подготовить LATEXовский исходный текст на своем IBM PC, переслать его в издательство и быть уверенными, что там Ваш текст будет правильно обработан и на печати получится в точности то же, что получилось у Вас при пробной печати на Вашем любимом матричном принтере (с той единственной разницей, что фотонаборный автомат даст текст более высокого качества).

Благодаря этому обстоятельству TEX стал очень популярен как язык международного обмена статьями по математике и физике.

Есть у TEXa и недостатки. Первый из них (разделяемый TEXом со всеми другими издательскими системами) таков: он работает относительно медленно, занимает много памяти, а полученный результат нельзя напечатать на дешевом принтере. Вторая особенность TEXa, которая может не понравиться тем, кто привык к редакторам наподобие Chiwriter, – это то, что он не является системой типа WYSIWYG: работа с исходным текстом и просмотр того, как текст будет выглядеть на печати, – разные операции. Впрочем, благодаря этой особенности, время на подготовку текста существенно сокращается.

Далее, хотя параметры оформления менять легко, создать принципиально новое оформление (новый «стиль») – непростое дело. Впрочем, LATEX предоставляет довольно широкиевозможности для модификации стандартных стилей.

Наконец, переносимость TEXовских текстов снижается, если в них предусмотрен импортграфических файлов (эта возможность в TEXe зависит от его реализации).

Преимущества

1. Четкая система структуризации разделов документа.

2. Автоматическая нумерация разделов, изображений, таблиц, расстановка ссылок на литературу и перекрестных ссылок, переносов.

3. Развитый механизм генерации библиографии, автоматическое создание оглавления одной командой.

4. LaTeX автоматизирует подготовку алфавитного указателя.

5. Высочайшее качество подготовки математических, физических, химических и биологических текстов с учетом принятых в полиграфии стандартов, а также любого документа научного характера (в т.ч. гуманитарного направления).

6. Автоматическая система «умной» компоновки и балансировки элементов документа в зависимости от пространствалиста и заданного типа документа.

7. Позволяет определять на странице несколько министраниц, причем на каждой из них допустимы сноски.

8. Большой набор инструментов для вставки диаграмм, графов, схем, алгоритмов в векторной форме.

9. Широкий спектр возможных документов: от простых деловых писем до статей, докладов и книг высочайшего полиграфического качества.

10. Владение LaTeX является показателем профессиональной компетентности научного работника любой отрасли (темболее работника сферы образования).

11. Является свободно распространяемым ПО.

Недостатки

1. В процессе компиляции файла возможно потребление значительных машинных ресурсов, больших, чем у примитивных текстовых процессоров.

2. Высокое качество типографии требует и хорошего качества печатного оборудования.

3. LaTeX великолепен и прост для стандартных типов документов, однако разметка нестандартного документа требует профессиональных навыков.

Необходимо отметить, что перечисленные недостатки являются условными:

1. современные компьютеры и прикладное ПО существенно ускоряют процесс компиляции файлов;

2. современное печатное оборудование дает высокое качество печати;

3. нестандартные форматы документов крайне редко требуются в науке и образовании.

Программа LATEX состоит из текста, команд и комментариев. Текст разбивается на слова, разделяемые пробелами, и абзацы, разделяемые пустыми строками. Между словами можно поставить любое количество пробелов, а между абзацами любое количество пустых строк, при компиляции они игнорируются. Символы перевода строки считаются обычными пробелами. Компилятор сам разбивает текст на строки и устанавливает промежутки между словами и строками, исходя из заданного стиля верстки. Дополнительные пробелы следует использовать для наглядного представления структуры программы и конструкций LATEX.

Комментарий открывается символом процента % и ограничивается концом строки. В комментариях, состоящих из нескольких строк, каждая из них должна начинаться символом процента. Текст и команды, находящиеся в комментарии, при компиляции полностью игнорируются. Комментарии могут находиться в любом месте программы, в том числе в аргументах и параметрах команд. Комментарий не является пробелом и потому не разрывает текст, но попытка разбить комментарием имя команды вызовет ошибку. Проиллюстрируем сказанное следующим примером (рис. 1.1):

```
Текст % А
будет не% комментарий
прерывен!% пропадет!
```

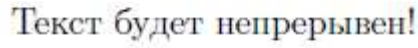


Рисунок 1.1 – Результат выполнения команды

Признаком команды является обратный слэш, за которым следует ее имя, параметры и аргументы:

```
\samplecommand[параметр]{аргумент}.
```

Имя команды состоит из латинских букв и звездочки *, которая может стоять в конце него. Аргументы команд заключаются вфигурные скобки, а

параметры в квадратные. Прописные и строчные буквы в именах различаются.

LATEX не имеет какого-либо специального признака окончанияимени команды. Им служит любой символ, отличный от латинских букв и звездочки. После имени с равным успехом могут стоять как пробел, 6, так и скобки, знаки препинания, цифры, русские буквы и т.д., в любом случае компилятор распознает команду.

Наличие и количество аргументов и параметров вводится определением команды. Ее строение не ограничено каким-либо шаблоном. LATEX не регламентирует порядок следования аргументов и параметров, однако есть общие правила, которым они подчиняются. Во-первых, аргументы и параметры всегда следуют за именем команды. Во-вторых, они могут содержать другие команды, пробелы, комментарии и т.д. И наконец, между именем, аргументами и параметрами допускаются пробелы и комментарии, но не должно быть пустых строк.

Аргумент содержит информацию, без которой команда неможет быть выполнена, или объекты, с которыми она должна совершить какое-то действие. Если аргумент не выделен явно фигурными скобками, вместо него используется символ (или команда), следующий непосредственно за командой, например. Если полученный таким образом аргумент неудовлетворяет требованиям команды, компилятор выдаст сообщение об ошибке. В частности, команда \hspace 1cm некорректна. Ее аргументом служит длина, характеризующаяся величинойи размерностью, а число 1, получаемое в качестве аргумента, размерности не имеет и потому длиной не является.

Описанный алгоритм действий компилятора называется правилом наследования аргумента.

Параметр служит для настройки каких-либо аспектов выполнения команды. Если параметр не используется, его можно опустить вместе с квад-

ратными скобками. В таком случае используется значение параметра, заданное по умолчанию при определении команды.

Для часто выполняемых действий используются односимвольные команды, или служебные символы, упрощенный синтаксискоторых является исключением из правил. Они собраны в первой колонке таблицы 1.1, помеченной символом •, во второй описано их назначение, а в последней приведены команды, используемые, чтобы напечатать сами символы. Часть из них также имеют нестандартный синтаксис.

Таблица 1.1 – Служебные символы

•	Назначение
\	признак команды
#	параметр команды
{	открывающая скобка
}	закрывающая скобка
_	нижний индекс
^	верхний индекс
$	математическая мода
&	разделение ячеек
%	комментарий
~	неразрывный пробел

Отметим, что команды нижнего и верхнего индекса имеют аргумент, подчиняющийся приведенному выше правилу наследования.

Одной из основных конструкций LATEX является окружение:

```
\begin{имя окружения}[параметры]{аргументы}
область действия окружения
\end{имя окружения}
```

Оно формируется командами \begin и \end, охватывающими часть программы, являющейся областью действия окружения. Их аргумент – имя

окружения, определяющее операцию, производимую окружением. Для имен справедливы требования, предъявляемые к именам команд.

Открывающая команда \begin может иметь дополнительные аргументы и параметры, а у закрывающей команды \end они отсутствуют, так как ее задача лишь ограничить область действия производимой операции. Окружения позволяют более наглядно представить структуру текста программы и в этом их основное назначение.

Области действия команд тесно связаны с группами – одним из базовых понятий LATEX. Прежде чем приступить к верстке, компилятор считывает множество определений команд и значений переменных, задающих их параметры. Лишь малая часть переменных является глобальными и еще меньшая – статическими, т.е. сохраняющими неизменными свои значения во время компиляции. Подавляющая же часть представляет собой локальные динамические переменные, значение которых определено только внутри группы.

Группой является аргумент и параметр команды, окружение, математическая формула, ячейка таблицы или просто часть текста, заключенная в фигурные скобки. С точки зрения компилятора, сам текст программы также представляет собой группу, так как заключен в окружение document. Группы вкладываются друг в друга: текст программы содержит окружения и команды, в аргументах и параметрах которых находятся другие окружения и команды со своими аргументами и параметрами, и т.д. Уровень вложенности может быть очень велик. При выходе из вложенной группы измененные локальные переменные восстанавливают свои значения, а вновь введенные переменные и команды становятся неопределенными.

Исходя из сказанного, можно условно выделить два типа команд. Область действия одних ограничена их собственным аргументом. Они создают группу (аргумент) и внутри нее меняют динамическую переменную,

например, параметр шрифта. К данному типу команд относятся и окружения, область действия которых также является группой.

Команды другого типа, называемые декларациями, действуют в пределах группы, в которой находятся, начиная с позиции, в которой они стоят. Декларации не действуют на предшествующий текст, но изменяя значение какой-либо динамической переменной, меняют режим верстки следующего за ними текста, вплоть до выхода из группы.

2. Входные файлы LATEX

Исходными данными для LATEX является обычный текстовый файл с расширением .tex. Его можно создать в любом текстовом редакторе (блокнот, MicrosoftWord, встроенный редактор Far и пр.). Он содержит текст документа вместе с командами, указывающими LATEX, каким образом верстать текст.

3. Структура входного файла.

Каждый документ LATEX должен следовать определенной структуре. Так, каждый входной файл должен начинаться с команды:

```
\documentclass[...]{...}
```

Она указывает, документ какого типа вы собираетесь писать. В квадратных скобках указываются параметры команды, в фигурных скобках указывается тип документа. После этого вы можете включать команды, влияющие на стиль документа в целом, или загружать пакеты, добавляющие новые возможности в систему LATEX. Для загрузки такого пакета используется команда:

```
\usepackage{...}
```

Когда настройка закончена, начинается тело документа командой:

```
\begin{document}
```

Далее вводится текст документа с командами TEX. В конце документа добавляется команда:

```
\end{document}
```

Любой текст, который следует после неё, LATEX игнорирует. На рисунке 1.2 представлено содержимое минимального файла LATEX.

```
\documentclass{article}
\begin{document}
Main text
\end{document}
```

Рисунок 1.2 – Минимальный входной файл LATEX

Область между командами `\documentclass{}` и `\begin{document}` называется преамбулой.

Область между командами `\begin{document}` и `\end{document}` называется телом документа.

4. Компиляция pdf-документа из входного файла

Создание pdf-документа по входному файлу выполняется в два шага.

Шаг 1. В командной строке выполните команду:

`latex<имя входного файла без расширения>`

Команда преобразует входной файл в файл формата dvi (DeviceIndependent), пригодный к распечатке.

В настоящее время файлы формата dvi используются для предпросмотра итогового документа.

Файл dvi можно просмотреть при помощи утилиты Yap, распространяемой вместе с дистрибутивом MikTeX.

Шаг 2. В командной строке выполните команду:

`dvipdfm<имя бинарного файла документа>`

Команда создает итоговый pdf-документ.

5. Задания

4.1. Создайте TEX-документ в любом текстовом редакторе (например, простой текстовый редактор с подсветкой синтаксиса - notepad++).

Поместите в тело документа следующий текст:

TeX – это компьютерная программа, созданная Дональдом Кнутом (Donald E.Knuth). Она предназначена для вёрстки текста и математических формул. Кнут начал писать TeX в 1977 году из-за расстройства от того, что Американское Математическое Сообщество делало с его статьями в процессе их публикации. Где-то в 1974 году он даже прекратил посылать статьи: «Просто мне было слишком больно смотреть на конечный результат». TeX в том виде, в котором мы его используем, был выпущен в 1982 году и слегка улучшен с годами. Последние несколько лет TeX стал чрезвычайно стабилен. Кнут утверждает, что в нем практически нет ошибок. Номер версии TeX сходится к Pi и сейчас равен 3.14159. TeX произносится как «TeX».

Выполните компиляцию документа, создайте pdf-файл.

Откройте программу предпросмотра pdf-файлов Previewer и просмотрите результат.

Для отображения русского текста необходимо подключить пакет `babel` с параметром `russian` и пакет `inputenc` с параметром `cp1251`. Изучите справку по команде `\usepackage[]{}` и подключите пакеты `babel` и `inputenc` в вашем исходном файле. Выполните компиляцию. Проверьте полученный результат.

4.2. Выполните оформление документа:

Замените везде в тексте слово TeX на официальный логотип TEX. Для этого воспользуйтесь командой `\TeX`.

Вместо текстовых кавычек (" ") воспользуйтесь принятыми в России французскими кавычками («»). Для этого используйте команды "< и ">.

Замените там, где это необходимо, дефис (-) на длинное тире (–). Для этого воспользуйтесь командой `\---`.

Замените в тексте слово «Pi» на математический символ π – `\pi`.

Проверьте полученный результат.

4.3. Поместите в конец тела исходного файла формулу:

```
\begin{equation}
\int \limits_S \left( \frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y} \right)\, dx \, dy =\oint \limits_C P\,dx + Q \, dy
\end{equation}
```

Изучите справку по параметрам команды `\documentclass`. Посмотрите, какое влияние на внешний вид документа оказывают необязательные параметры `twocolumn`, `leqno`, `fleqn`. Можно ли добиться таких же эффектов в текстовом редакторе MS Word?

4.4. Используя любое из предложенных учебных пособий по TEX, измените стиль шрифта для фамилии автора TeX на курсивный, стиль шрифта текста цитаты на полужирный, а текст, описывающий назначение TeX,– наклонным стилем.

4.5. Запрограммируйте в отдельном документе формулы интегрирования суммы и разности двух функций, а также правило интегрирования функции, умноженной на постоянную (формулы должны располагаться в отдельных строках и пронумерованы автонумерацией).

5. Контрольные вопросы

1. Что нужно изменить в тексте документа, если вы планируете использовать кодировку DOS (CP-866)?
2. Что такое символы группирования "{" и "}"? Для чего они используются в TEX?
3. Что такое окружения, для чего они используются в TEX?
4. Что такое параметры команды TEX?
5. Что означает команда `\documentclass`? Какие существуют параметры команды `\documentclass`. Как при помощи данной команды выставить основной шрифт документа размером 14 пт.

Лабораторная работа №2. Структура документа LATEX

1. Введение

Как правило, большинство документов, таких как книги, технические отчёты, научные статьи, имеют следующий стандартный набор компонентов:

- Заголовок;
- Список авторов;
- Аннотация;
- Основной текст, состоящий из разделов, каждый из которых начинается с заголовка;
- Список использованной литературы.

На рисунке 2.1 представлен пример простого документа с описанной выше структурой.

```
\documentclass[11pt]{article}
\usepackage[english, russian]{babel}
\usepackage[cp1251]{inputenc}
\begin{document}
\title{Название документа}
\author{А.В. Иванов}
\maketitle
\begin{abstract}
Аннотация. Аннотация. Аннотация. Аннотация. Аннотация. Аннотация.
\end{abstract}
\section{Заголовок раздела первого уровня}
\subsection{Заголовок раздела второго уровня}
Текст раздела второго уровня
\subsubsection{Заголовок раздела третьего уровня}
Текст раздела третьего уровня
\begin{thebibliography}{Литература}
\bibitem{1} Сюткин Вл. Набор математических формул в LaTeX2.
\end{thebibliography}
\end{document}
```

Рисунок 2.1 – Структура

Параметры страницы

Страница печатного документа состоит из верхнего и нижнего колонтитулов и области, в которой размещается содержание документа: текст и подстрочные примечания. Кроме того, на боковых полях страницы могут размещаться заметки на полях, которые печатает команда `\marginpar`. Размер и расположение колонтитулов, области с содержанием документа и заметок на полях задаются нерастяжимыми командными длинами, приведенными на рисунке 2.2. Их значения, установленные по умолчанию, можно изменить в преамбуле документа декларациями `\setlength` и `\addtolength`.

Текущие значения параметров компоновки страницы можно узнать с помощью пакета layout из коллекции tools. Команда `\layout` из этого пакета печатает макет страницы, на которой она находится, с указанием значений всех параметров. Команда различает правые и левые страницы, одно- и двухколоночный режимы печати.

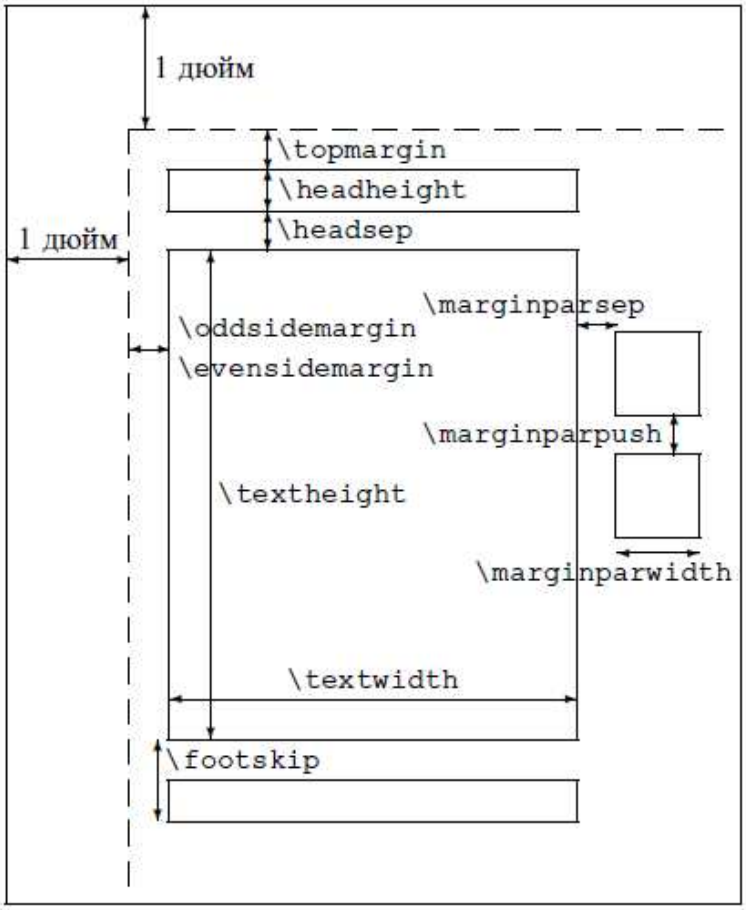


Рисунок 2.2 – Макет страницы с командами, задающими размер и расположение верхнего и нижнего колонтитулов, области с содержанием документа и заметок на полях

Команды `\oddsidemargin` и `\evensidemargin` задают левое поле для нечетные (odd) и чётных (even) страниц, соответственно.

Титульная страница и аннотация

Стандартный заголовок, состоящий из названия, имён авторов и даты создания документа, печатает команда

```
\maketitle
```

Ей должны предшествовать две команды

```
\title{title}
\author{author(s)}
```

содержащие название документа title и имена авторов author(s). Для разбиения длинного названия или списка авторов на строки используется команда \\. Аргументы обеих команд могут быть пустыми.

Перед `\maketitle` можно с помощью команды

`\date{date}`

указать дату создания документа date. Если команда `\date` отсутствует, то печатается текущая дата.

Если дата не нужна, то надо использовать команду `\date` с пустым аргументом {}.

Аргументы команд `\title`, `\author`и `\date` могут содержать команду

`\thanks{text}`

которая печатает text как подстрочное примечание.

В стандартных классах команда `\maketitle` печатает заголовок на отдельной странице, если действует опция titlepage. Страница, следующая за титульной, нумеруется как 1-ая. Если действует опция notitlepage, то заголовок печатается с новой страницы прямо перед содержанием документа. В классе article по умолчанию используется notitlepage, а в классах book, report и slides–titlepage.

В аргументе команды `\author` можно использовать команду `\and` для разделения author(s) на боксы. LATEX, формируя из этих боксов строку, отделяет их друг от друга большими пробелами. Каждый бокс может сам состоять из нескольких строк.

Если формат стандартного заголовка не соответствует требуемому, то надо использовать командные скобки

`\begin{titlepage} . . . \end{titlepage}`

для создания титульной страницы. На этой странице печатается содержание окружения titlepage.

Страница, следующая за титульной, нумеруется как 1-ая.

В классах article и report определены командные скобки

`\begin{abstract} . . . \end{abstract}`

для печати аннотации к статье. Она печатается на отдельной странице, если действует опция titlepage. Перед аннотацией LATEX печатает заголовок Abstract. Он хранится в команде

`\abstractname`

которую можно переопределить с помощью `\renewcommand`. Пакет babel с опцией russian переопределяет её как Аннотация.

При наличии опции twocolumn аннотация, как и сам документ, печатается в двухколоночном режиме. Пакет abstract позволяет напечатать её в одноколоночном режиме (см. документацию кпакету).

Секционирование документа

В стандартных классах, за исключением letter, определены команды для разделения документа насекции. Все команды имеют по два аргумента, один из которых является обязательным. Обязательный аргумент печатается как название секции и, если отсутствует необязательный аргумент, включается в оглавление и используется при оформлении колонтитулов. Для оглавления и колонтитулов можно задать более компактное название секции в виде необязательного аргумента. Неустойчивые команды в подвижных аргументах должны быть защищены командой `\protect`. Для разбиения длинного названия на строки внутри обязательного аргумента можно использовать команду \\.

LATEX автоматически нумерует секции, уровень которых не превышает число, которое хранится в счётчике

`secnumdepth`

Его значение можно изменить декларацией `\setcounter`. Секциям, созданным командой `\section`, присвоен уровень 1. Счётчики секций более младшего уровня определены как внутренние по отношению к счётчикусекций старшего уровня, поэтому в каждой секции подсекции нумеруются независимо друг от друга.

Все команды секционирования имеют *-форму. Она имеет только обязательный аргумент, который печатается как название секции. Такие секции не нумеруются и не заносятся в оглавление и колонтитулы.

Части

Команды

```
\part[toc]{head}   \part*{head}
```

позволяют разделить документ на части. В классах book и report команда `\part` печатает заголовок в центре отдельной страницы: сначала слово Part и порядковый номер части прописными римскими цифрами, а затем с новой строки название части head. В классе article заголовок печатается прямо перед содержанием части. При наличии опции, для оглавления и колонтитулов используется не head, а toc. Команда `\part*` печатает только название части.

Команды `\part` и `\part*` являются необязательными, поэтому они не влияют на порядок нумерации более младших секций.

Слово Part хранится в команде

```
\partname
```

Её можно переопределить с помощью `\renewcommand`. Пакет babel с опцией russian переопределяет её как Часть.

Главы

В классах book и report определены команды для разделения документа на главы:

```
\chapter[toc]{head}   \chapter*{head}
```

Каждая глава печатается с новой страницы: правой, если действует опция openright, или любой, если действует опция openany. В классе book по умолчанию используется openright, а в классе report–openany.

Команда `\chapter` печатает сначала слово Chapter и порядковый номер главы арабскими цифрами. Название главы head печатается с новой стро-

ки. При наличии опции, для оглавления и колонтитулов используется не head, а toc. Команда `\chapter*` печатает только название главы.

Слово Chapter хранится в команде

`\chaptername`

Её можно переопределить с помощью `\renewcommand`. Пакет babel с опцией russian переопределяет её как Глава.

Разделы

Для деления документа на разделы, подразделы и подподразделы определены команды:

```
\section[toc]{head}          \section*{head}
\subsection[toc]{head}       \subsection*{head}
\subsubsection[toc]{head}    \subsubsection*{head}
```

Они печатают перед названием секции head её порядковый номер. При наличии опции, для оглавления и колонтитулов используется не head, а toc. *-форма команд печатает только название секции.

Параграфы

Команды

```
\paragraph[toc]{head}        \paragraph*{head}
\subparagraph[toc]{head}     \subparagraph*{head}
```

печатают в начале первого абзаца секции шрифтом полужирной насыщенности её название head.

При этом команда `\paragraph` подавляет абзацный отступ, а `\subparagraph` – нет. Между секциями вставляется дополнительный вертикальный промежуток. При наличии опции, для оглавления и колонтитулов используется не head, а toc. *-форма команд печатает только название секции.

Разделение книги на части

В классе book определены декларации

`\frontmatter\mainmatter\backmatter`

которые открывают вводную, основную и заключительную части книги, соответственно. В вводной части страницы нумеруются римскими цифрами, а в остальных – арабскими (в России принятасквозная нумерация арабскими цифрами). Кроме того, в вводной и заключительной частях команда `\chapter` не печатает номер главы, но заносит её заголовок в оглавление.

Приложения

Если в документе имеются приложения, то перед ними надо поставить декларацию

`\appendix`

После неё новая глава начинается не со слова Chapter, а со слова Appendix. Кроме того, меняется формат нумерации глав: вместо арабских цифр используются заглавные латинские буквы A, B, C и т.д. В классе article, где нет глав, буквами нумеруются разделы, созданные командой `\section`.

Слово Appendix хранится в команде

`\appendixname`

Её можно переопределить с помощью `\renewcommand`. Пакет babel с опцией russian переопределяет её как Приложение.

3. Задание

Оформите в среде LATEX документ, используя текстовую информацию из pdf согласно варианту, содержащий следующие элементы:

- Заголовок – название, список авторов, сноска к названию статьи (воспользуйтесь командой `\maketitle`), в сноске указать: Лабораторная работа №2. Вариант №_*ваш вариант*_. ФИО студента
- Аннотация (воспользуйтесь командными скобками `\begin{abstract} \end{abstract}`)
- Основной текст, состоящий из разделов, каждый из которых начинается с заголовка (для вставки заголовка раздела используйте команду `\section{}`)

- Список использованной литературы. (Воспользуйтесь командными скобками `\begin{thebibliography}{<Отступ>} \end{thebibliography}`).

При необходимости, если LATEX не может грамотно выполнить перенос в документе, вставляйте перенос вручную (команда "\-" указывает LATEX те места в слове, где можно делать перенос). Заметьте, что данная команда является только рекомендацией и LATEX может её игнорировать. Команда "\\" обрывает вёрстку текущей строки и начинает новую строку. Применяется для разбиения длинных названий на строки.

В начале каждого абзаца необходимо вставлять команду "`\par`". Данная команда предписывает LATEX выполнить перенос строки и сделать абзацный отступ.

3. Контрольные вопросы

1. Введение и список литературы обычно не нумеруются. Каким образом убрать из заголовка раздела автоматически вставляемый командой `\section{}` номер?
2. Что означает параметр команды `\bibitem`, расположенный в квадратных скобках.
3. Как убрать дату из заголовка статьи?

Лабораторная работа №3. Верстка мультифайлового документа в LATEX

1. Несколько файлов в LATEX

При использовании LaTeX можно столкнуться с такой проблемой, что документ становится слишком большим, чтобы им манипулировать. Следовательно, нам следует поделить файл так, чтобы с его содержимым можно было легче справиться.

Давайте посмотрим на примере:

```
% main.tex
\documentclass[a4paper]{article}
\begin{document}
Привет, Latex, это моя первая часть.
Привет, Latex, это моя вторая часть.
\end{document}
```

Это просто обычный файл LaTeX. Теперь давайте разделим его на две части с помощью ключевого слова \input:

```
%основной_файл.tex
\documentclass[a4paper]{article}
\begin{document}
Привет, Latex, это моя первая часть.
\input{второй_файл}
\end{document}

% второй_файл.tex
Привет, Latex, это моя вторая часть.
```

2. Задание

Разбейте документ, созданный в предыдущей лабораторной работе, на следующие файлы.

1. Main: содержит все стили применяемые ко всем файлам и подключение всех файлов.
2. HeaderRu: содержит заголовок, авторов, ключевые слова и аннотацию на русском языке.
3. HeaderEng: содержит заголовок, авторов, ключевые слова и аннотацию на английском языке.
4. IntroducePart: содержит введение.
5. MainPart: содержит основную часть статьи.
6. ConclusionPart: содержит заключение.
7. Bibliography: содержит список литературы.

3. Контрольные вопросы

1. Что такое ключевое слово `\input` в контексте LaTeX?
2. Как использовать ключевое слово `\input` для объединения нескольких файлов LaTeX в один документ?
3. Какой синтаксис используется для разделения файла LaTeX на несколько частей?
4. Что должно быть содержимым файлов, на которые ссылается ключевое слово `\input`?
5. Что нужно учитывать при разделении файла LaTeX на части?

Лабораторная работа №4. Форматирование текста. Позиционирование текста и списки

1. Позиционирование текста

Абзац начинается командой `\par`, которая осуществляет перевод строки, выполняет вертикальный отступ (величина которого задается переменной `\parskip`) и делает для первой строки абзаца отступ от левого края абзаца (величина которого задается переменной `\parindent`).

Все строки в командных скобках `\begin{center} \end{center}` или в области действия декларации `\centering` центрируются.

Строки в командных скобках `\begin{flushleft} \end{flushleft}` или в области действия декларации `\raggedright` прижимаются к левому краю страницы.

Строки в командных скобках `\begin{flushright} \end{flushright}` или в области действия декларации `\raggedleft` прижимаются к правому краю страницы.

Для ручной вставки горизонтального или вертикального промежутка необходимо использовать команду `\hspace{<размер>}` или `\vspace{<размер>}`.

2. Списки

Маркированный список создаётся при помощи окружения `\begin{itemize} \end{itemize}`

Каждый элемент списка начинается с команды `\item`. Например:

```
\begin{itemize}
\itemПервый элемент
\itemВторой элемент
\itemТретий элемент
\end{itemize}
```

Допускаются вложенные списки четырех уровней. Перед каждым из элементов печатается установленный по умолчанию маркёр.

По умолчанию для маркированного списка первого уровня устанавливается маркер "•".

Для второго уровня - "-"

Для третьего уровня - "*"

Для четвертого уровня - "."

Вид маркёра задаётся переменными `\labelitemi`, `\labelitemii`, `\labelitemiii`, `\labelitemiv` для списков первого, второго, третьего и четвертого уровня, соответственно.

Нумерованный список создаётся при помощи окружения `\begin{enumerate} \end{enumerate}`

3. Задание

Отформатируйте документ, созданный в предыдущей работе, в соответствии со следующими правилами:

Оформление основного текста:

- абзацный отступ – 1,25 см;
- первый абзац раздела не имеет абзацного отступа;
- маркированные списки помечаются маркёром "-" (необходимо переопределить команду `\labelitemi`);
- интервал между абзацами – 6pt;
- выравнивание абзаца – по левому краю.

Оформление заголовка раздела:

- выравнивание заголовка раздела по левому краю;
- отступ сверху и снизу заголовка раздела – 6pt;
- размер шрифта заголовка – 14pt.

4. Контрольные вопросы

1. Что происходит со строками в командах `\begin{center}` и `\end{center}`?
2. Что происходит со строками в области декларации `\centering`?
3. Что происходит со строками в командах `\begin{flushleft}` и `\end{flushleft}`?
4. Что происходит со строками в области декларации `\raggedright`?
5. Что происходит со строками в командах `\begin{flushright}` и `\end{flushright}`?
6. Что происходит со строками в области декларация `\raggedleft`?
7. Как вставить горизонтальный или вертикальный промежуток в LaTeX?

Лабораторная работа №5. Формулы в LATEX

1. Общие сведения о формулах

Пожалуй, наиболее существенное преимущество LaTeX – высочайшее качество оформления математических символов. В первой главе мы уже отмечали, что LaTeX отлично подходит для верстки технической документации и документов, содержащих математическую символику.

Виды формул

Формулы LaTeX можно разделить на две категории:

- внутренние – пишутся в строке абзаца;
- выключенные – указываются отдельной строкой и выравниваются по центру.

Для набора внутренней формулы ее текст помещают между символами

```
\( выражение \)
```

Например (рис. 5.1):

```
Квадратное уравнение имеет вид \(ax^2+bx+c=0\), где
\(a\), \(b\) и \(c\) -- действительные коэффициенты.
```

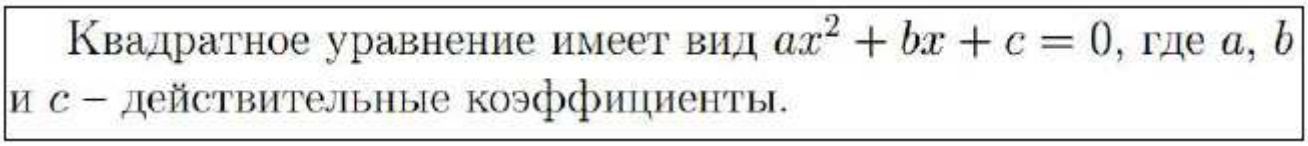


Рисунок 5.1 – Результат примера

Выключенная формула набирается внутри следующих ограничителей:

```
\[ формула \]
```

Например (рис. 5.2):

```
Согласно тождеству Эйлера справедливо:
\[
e^{i\pi} + 1 = 0.
\]
где \(i\) -- комплексная единица.
```

Рисунок 5.2 – Результат примера

Формулы, в отличие от текста, имеют несколько иной алгоритм обработки. Укажем некоторые особенности.

1. Любые пробелы игнорируются и настраиваются только специальными командами (горизонтальными пробелами).
2. В формулах запрещены пустые строки.
3. Шрифт формулы устанавливается курсивом.

Старый стандарт оформления формул

LaTeX поддерживает и другой, более старый стандарт оформления формул. Так, внутренняя формула помещается между ограничителями

$формула$

а выключенная формула –

```
$$
    формула
$$
```

На самом деле сейчас нет жесткого требования к выбору нового или старого оформления. Используйте то, которое вам удобнее.

2. Оформление формул.

Таблица 5.1 – Верхние и нижние индексы

Команда	Описание
_	Нижний индекс
^	Верхний индекс
{ }	Скобки для ограничения выражения

Пример:

```
Для прямоугольного треугольника с катетами $a$, $b$ и ги-
потенузой $c$ справедливо
$$
c^2 = a^2 + b^2.
$$
Формула серной кислоты --- $H_2SO_4$.
Двойной индекс: $X_{2017}^{2018}$.
Если забыть про скобки: $X_2017^2018$.
```

Результат (рис.5.3):

Для прямоугольного треугольника с катетами a, b и гипотенузой c справедливо

$$c^2 = a^2 + b^2.$$

Формула серной кислоты — H_2SO_4.

Двойной индекс: X_{2017}^{2018}.

Если забыть про скобки: X_2017^2018.

Рисунок 5.3 – Результат выполнения примера

Дроби

```
\frac{числитель}{знаменатель}
```

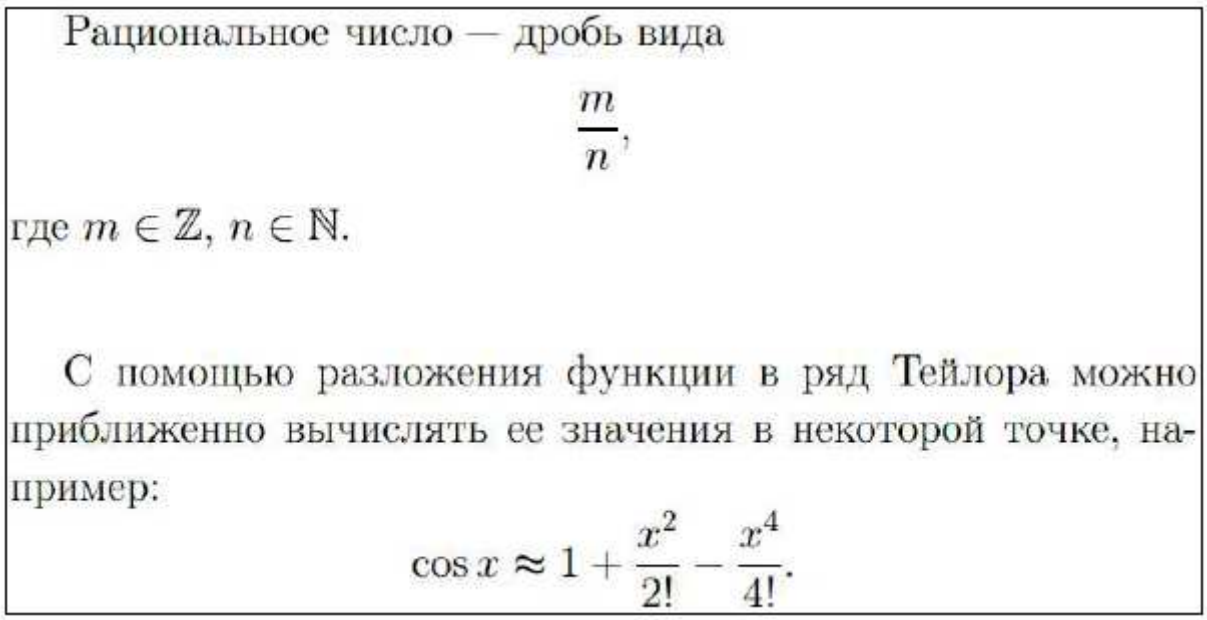
Печатает дробь с указанным числителем и знаменателем.

Пример:

```
Рациональное число --- дробь вида
$$
\frac{m}{n},
$$
где $m\in \mathbb{N}$, $n\in \mathbb{Z}$.
```

```
С помощью разложения функции в ряд Тейлора можно прибли-
женно вычислять ее значения в некоторой точке, например:
$$
\cos x \thickapprox
1+ \frac{x^2}{2!} - \frac{x^4}{4!}.
$$
```

Результат (рис. 5.4):


Рациональное число — дробь вида

$$\frac{m}{n},$$

где $m \in \mathbb{Z}$, $n \in \mathbb{N}$.

С помощью разложения функции в ряд Тейлора можно приближенно вычислять ее значения в некоторой точке, например:

$$\cos x \approx 1 + \frac{x^2}{2!} - \frac{x^4}{4!}.$$

Рисунок 5.4 – Результат выполнения примера

При вложении дробей в дроби (т.н. «многоэтажные дроби») LaTeX будет их масштабировать. Если требуется этого избежать, укажите команду \displaystyle. Примечательно, что она действует только в пределах уровня дроби, на котором указана.

Пример:

```
Дробь автоматически масштабируется:
$$
\frac{1}{1+\frac{1}{x}}
$$
Дробь не масштабируется:
$$
```

```
\frac{1}{\displaystyle 1+\frac{1}{x}}
$$
```

Результат (рис. 5.5):

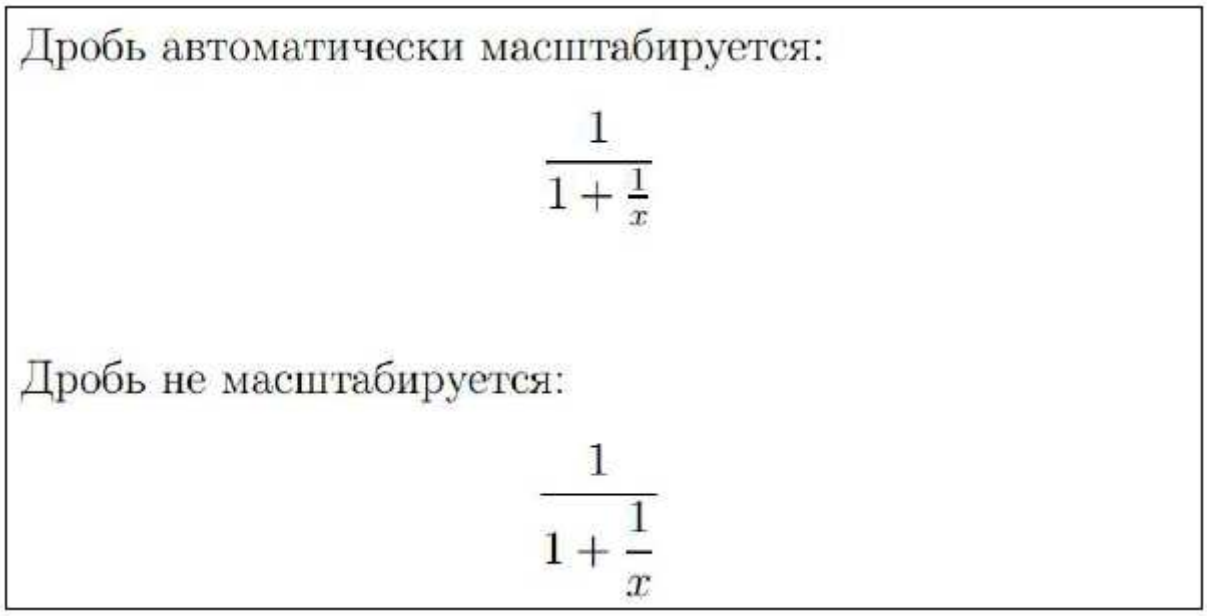


Рисунок 5.5 – Результат выполнения примера

Корни

```
\sqrt{выражение}
\sqrt[n]{выражение}
```

Первая команда берет квадратный корень от выражения.

Вторая команда берет корень n-й степени от выражения.

Пример:

```
$$
f(x) = \sqrt{\frac{x^4+1}{x^2+1}}
$$
Забавный факт из математики:
$$
\sqrt{2 + \sqrt{2 + \sqrt{2 + \ldots}}} = 2.
$$
$$
\sqrt[3]{8} = 2
```

```
$$
```

Результат (рис. 5.6):

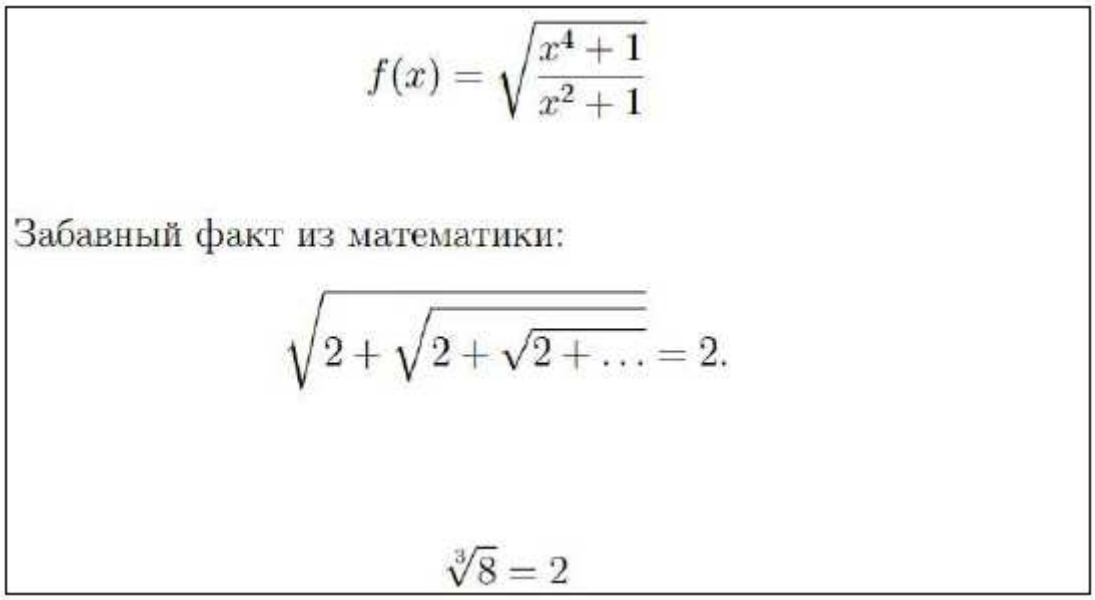


Рисунок 5.6 – Результат выполнения примера

Скобки

Скобки выступают в роли ограничителей операций формулы. По умолчанию LaTeX не масштабирует скобки согласно высоте вложенного в них выражения, поэтому такой вариант оформления подходит только для формул не выше одной строки.

Для масштабирования скобок к открывающей скобке приписывают команду `\left`, а к закрывающей – `\right`:

```
\left( выражение \right)
```

Если скобку требуется установить только с одной стороны, то для левого ограничителя используйте команду вида

```
\left( выражение \right.
```

а для правого

```
\left. выражение \right)
```

В таблице (рис. 5.7) приведены поддерживаемые в LaTeX скобки:

Команда	Вид	Команда	Вид
(	(	)	)
[	[	]	]
\{	{	\}	}
\lfloor	⌊	\rfloor	⌋
\lceil	⌈	\rceil	⌉
\langle	⟨	\rangle	⟩
|	‖	\\|	‖
/	/	\backslash	\

Рисунок 5.7 – Таблица поддерживаемых в LaTeX скобок

Пример:

Обычные скобки не масштабируются:

```
$$
(1+\frac{1}{2})
$$
А вот так лучше:
$$
\left( 1+\frac{1}{2} \right)
$$
Непарный ограничитель:
$$
|x| =
\left\{
\begin{array}{l}
x, \: \mbox{если} \: x \geqslant 0, \\
-x, \: \mbox{если} \: x <0.
\end{array}
\right.
$$
```

Здесь отметим ряд важных замечаний. Во-первых, для вывода текста в формуле используется команда `\mbox`. Во-вторых, окружение `array` позволяет оформлять матрицы (таблично подобные структуры).

Результат (рис. 5.8):

Рисунок 5.8 – Результат выполнения примера

Символ суммирования

`\sum`

Оператор суммирования.

С помощью верхних и нижних индексов можно проставлять пределы суммирования.

Пример:

```
Среднее арифметическое нескольких чисел:
$$
\overline{X} = \frac{a_1 + a_2 + \ldots + a_n}{n}
$$
или кратко
$$
```

```
\overline{X} = \frac{1}{n} \sum_{k=1}^{n} a_k
$$
228
```

Результат (рис. 5.9):

Среднее арифметическое нескольких чисел:

$$\overline{X} = \frac{a_1 + a_2 + \ldots + a_n}{n}$$

или кратко

$$\overline{X} = \frac{1}{n} \sum_{k=1}^{n} a_k$$

Рисунок 5.9 – Результат выполнения примера

Символы интегрирования

```
\int, \iint, \iiint, \idotsint, \oint
```

Символы интегралов. Первые три – обычный, двойной, тройной и кратный интеграл, последний – по замкнутому контуру.

Для указания пределов интегрирования используются команды форматирования индексов. Также после команды интеграла может следовать команда \limits, которая размечает пределы интегрирования строго сверху и снизу символа интеграла. (В обычном режиме пределы печатаются справа и сбоку, поэтому по вертикали такая формула будет занимать меньше места).

Пример:

```
Неопределенный интеграл:
$$
\int (ax + b) \: dx = \frac{ax^2}{2} + bx + c.
$$
Определенный интеграл (форма 1):
$$
```

```
\int_{a}^{b} f(x) \: dx.
$$
Определенный интеграл (форма 2):
$$
\int\limits_{a}^{b} f(x) \: dx.
$$
Двойной и тройной интеграл:
$$
\iint\limits_{\Omega} f(x,y) \: dxdy, \quad
\iiint\limits_{\Omega} f(x,y,z) \: dx dy dz.
$$
Кратный интеграл в общем виде:
$$
\idotsint\limits_{\Omega} f(x) \: dx.
$$
Криволинейныйинтеграл:
$$
\oint\limits_{C} F(x,y) \: dx + G(x,y) \: dy.
$$
```

Результат (рис. 5.10):

Неопределенный интеграл:

$$\int (ax+b)\,dx = \frac{ax^2}{2} + bx + c.$$

Определенный интеграл (форма 1):

$$\int_a^b f(x)\,dx.$$

Определенный интеграл (форма 2):

$$\int\limits_a^b f(x)\,dx.$$

Двойной и тройной интеграл:

$$\iint\limits_\Omega f(x,y)\,dxdy, \quad \iiint\limits_\Omega f(x,y,z)\,dxdydz.$$

Кратный интеграл в общем виде:

$$\int\limits_\Omega \cdots \int f(x)\,dx.$$

Криволинейный интеграл:

$$\oint\limits_C F(x,y)\,dx + G(x,y)\,dy.$$

Рисунок 5.10 – Результат выполнения примера

Пределы

```
\lim
```

Печатает символ предела.

Пример:

```
$$
\lim_{n\to\infty} 1 + \frac{1}{n} = 1.
```

```
$$
```

Результат (рис. 5.11):

$$\lim_{n\to\infty} 1+\frac{1}{n} = 1.$$

Рисунок 5.11 – Результат выполнения примера

Математические функции и греческие буквы

В таблице (рис. 5.12) приведены математические функции. По правилам типографии их принято оформлять прямым начертанием.

Команда	Вид	Команда	Вид	Команда	Вид
\sin	sin	\cos	cos	\tan	tan
\ctg	ctg				
\arcsin	arcsin	\arccos	arccos	\arctan	arctan
\sec	sec	\csc	csc		
\log	log	\ln	ln	\lg	lg
\exp	exp				
\sinh	sinh	\cosh	cosh	\tanh	tanh
\min	min	\max	max	\dim	dim

Рисунок 5.12 – Таблица математических функций

Греческие символы также набираются с помощью специальных команд (рис. 5.13):

α	\alpha	λ	\lambda	ϕ	\phi
β	\beta	μ	\mu	χ	\chi
γ	\gamma	ν	\nu	ψ	\psi
δ	\delta	ξ	\xi	ω	\omega
ϵ	\epsilon	o	o	ε	\varepsilon
ζ	\zeta	π	\pi	ϑ	\vartheta
η	\eta	ρ	\rho	ϖ	\varpi
θ	\theta	σ	\sigma	ϱ	\varrho
ι	\iota	τ	\tau	ς	\varsigma
κ	\kappa	υ	\upsilon	φ	\varphi

Γ	\Gamma	Ξ	\Xi	Φ	\Phi
Δ	\Delta	Π	\Pi	Ψ	\Psi
Θ	\Theta	Σ	\Sigma	Ω	\Omega
Λ	\Lambda	Υ	\Upsilon		

Рисунок 5.13 – Таблица греческих символов

Пример:

```
Косинус двойного угла:
$$
\cos 2x = \cos^2 x - \sin^2 x.
$$
Сложная функция:
$$
y(x) =
\cos \left(
\frac{1 + x^2}{\ln|x+1|}
\right).
$$
Греческие символы:
$$
F(x,\xi,\psi) = \alpha \xi^2(x) + \beta \psi^2(x).
$$
```

Результат (рис. 5.14):

Косинус двойного угла:

$$\cos 2x = \cos^2 x - \sin^2 x.$$

Сложная функция:

$$y(x) = \cos\left(\frac{1+x^2}{\ln|x+1|}\right).$$

Греческие символы:

$$F(x,\xi,\psi) = \alpha\xi^2(x) + \beta\psi^2(x).$$

Рисунок 5.14 – Результат выполнения примера

3. Окружение equation

Автоматическая нумерация

Окружение equation

Оформляет выключенные формулы и автоматически нумерует их в порядке следования.

- Может быть помечена командой \label, что позволяет ссылаться на формулу в тексте.
- Ссылаться на формулу можно с помощью команды \ref.
- Более удобной является ссылка \eqref из пакета AMS: она автоматически пишет номер в скобках.

Пример:

```
Формула Эйлера утверждает, что для любого комплексного(действительного) числа $x$ выполнено следующее равенство:
  \begin{equation} \label{eq:euler_formula}
  e^{ix} = \cos x + i \sin x
  \end{equation}
```

Результат (рис. 5.15):

Формула Эйлера утверждает, что для любого комплексного (действительного) числа x выполнено следующее равенство:

$$e^{ix} = \cos x + i \sin x \qquad (1)$$

В частности, при $x = \pi$ в (1), получим тождество Эйлера:

$$e^{i\pi} + 1 = 0. \qquad (2)$$

Рисунок 5.15 – Результат выполнения примера

Произвольное обозначение формул

Формула может быть обозначена любым символом или текстом. Для этого в конце формулы используется команда `\eqno`. Эта команда работает только для выключенных формул внутри парных символов

`$$`.

Пример:

Для комплексной единицы справедливо

```
$$
i^2 + 1 = 0, \quad\mbox{где}\quadx\in\mathbb{C}.
\eqno(**)
$$
```

Результат (рис. 5.16):

Для комплексной единицы справедливо

$$i^2 + 1 = 0, \quad \text{где} \quad x \in \mathbb{C}. \qquad (**)$$

Рисунок 5.16 – Результат выполнения примера

Старайтесь избегать такого подхода в обозначении формул: он не позволяет организовать автоматически обновляемую ссылку.

Кроме того, старайтесь работать именно с окружение mequation. Оно обезопасит вас от неверной нумерации формул вручную.

4. Разные символы

Приведем еще ряд примеров часто используемых символов (рис. 5.17 – 5. 21).

†	\dag	§	\S	£	\pounds
‡	\ddag	¶	\P	№	\No
Ø	\O	ø	\o	©	\copyright
Œ	\OE	œ	\oe	ı	\i
Æ	\AE	æ	\ae	ȷ	\j
Å	\AA	å	\aa	ß	\ss
Ł	\L	ł	\l		

Рисунок 5.17 – Таблица символов

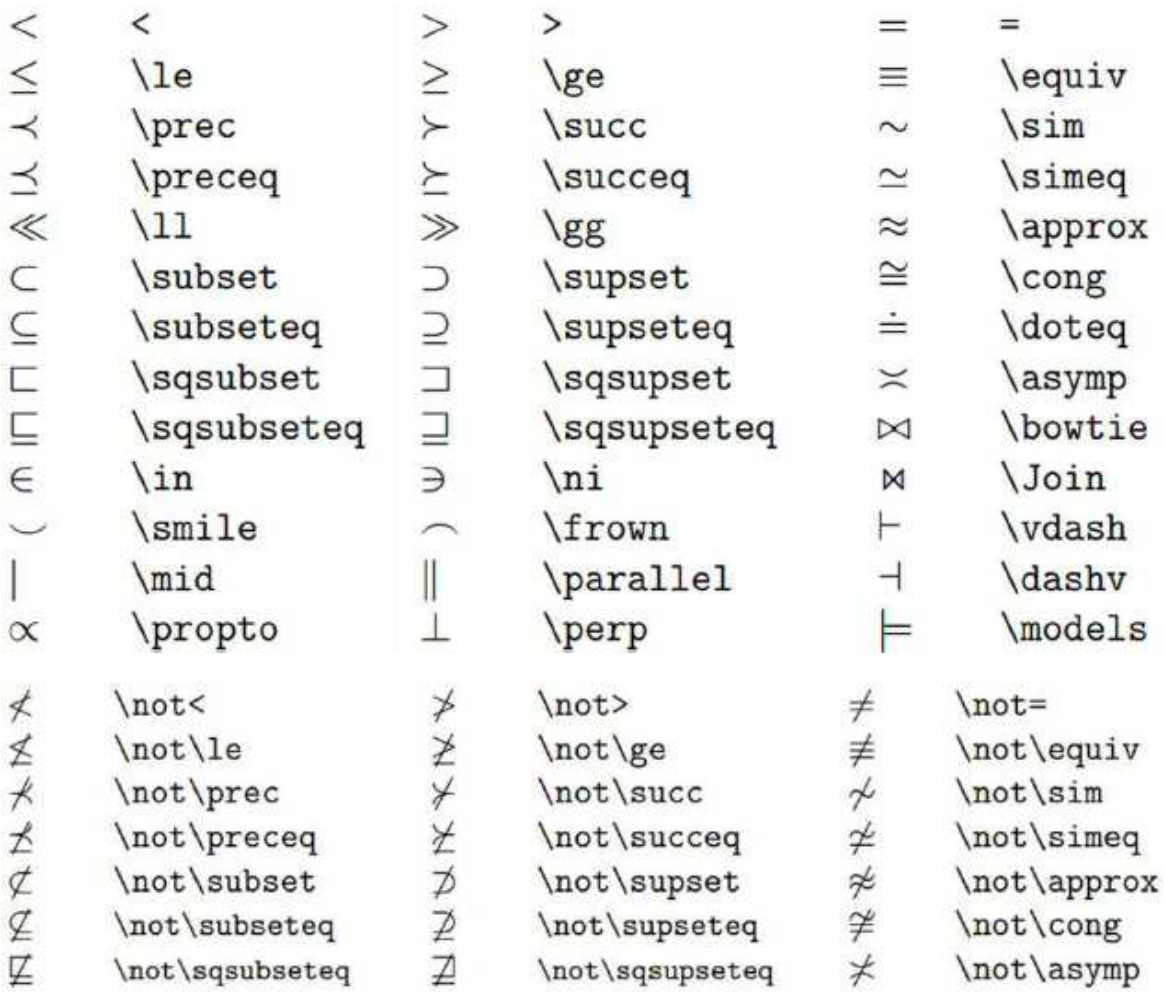
$\aleph$	\aleph	$\prime$	\prime	$\forall$	\forall
$\hbar$	\hbar	$\emptyset$	\emptyset	$\exists$	\exists
$\imath$	\imath	∇	\nabla	$\neg$	\neg
$\jmath$	\jmath	$\surd$	\surd	$\flat$	\flat
ℓ	\ell	$\top$	\top	$\natural$	\natural
$\wp$	\wp	$\bot$	\bot	$\sharp$	\sharp
$\Re$	\Re	$\Vert$	\Vert	$\clubsuit$	\clubsuit
$\Im$	\Im	$\angle$	\angle	$\diamondsuit$	\diamondsuit
∂	\partial	$\triangle$	\triangle	$\heartsuit$	\heartsuit
∞	\infty	$\backslash$	\backslash	$\spadesuit$	\spadesuit
$\mho$	\mho	$\Box$	\Box	$\Diamond$	\Diamond
$\cdots$	\cdots	$\vdots$	\vdots	$\ddots$	\ddots

$\angle$	\angle	$\beth$	\beth	$\square$	\square
$\measuredangle$	\measuredangle	$\gimel$	\gimel	$\blacksquare$	\blacksquare
$\sphericalangle$	\sphericalangle	$\daleth$	\daleth	$\lozenge$	\lozenge
$\Finv$	\Finv	$\digamma$	\digamma	$\blacklozenge$	\blacklozenge
$\Game$	\Game	$\eth$	\eth	$\triangledown$	\triangledown
$\mho$	\mho	$\Bbbk$	\Bbbk	$\blacktriangledown$	\blacktriangledown
$\nexists$	\nexists	$\hslash$	\hslash	$\blacktriangle$	\blacktriangle
$\varnothing$	\varnothing	$\hbar$	\hbar	$\bigstar$	\bigstar
$\circledS$	\circledS	$\backprime$	\backprime	$\complement$	\complement

Рисунок 5.18 – Таблица символов

$+$	+	$\oplus$	\oplus	$\setminus$	\setminus
$-$	-	$\ominus$	\ominus	$/$	/
$\pm$	\pm	$\otimes$	\otimes	$\vee$	\vee
$\mp$	\mp	$\oslash$	\oslash	$\wedge$	\wedge
$\times$	\times	$\odot$	\odot	$\cap$	\cap
$\div$	\div	$\bigcirc$	\bigcirc	$\cup$	\cup
$*$	\ast	$\dagger$	\dagger	$\uplus$	\uplus
$\star$	\star	$\ddagger$	\ddagger	$\sqcap$	\sqcap
$\diamond$	\diamond	$\wr$	\wr	$\sqcup$	\sqcup
$\circ$	\circ	$\bullet$	\bullet	$\amalg$	\amalg
$\triangleleft$	\triangleleft	$\cdot$	\cdot	$\bigtriangleup$	\bigtriangleup
$\triangleright$	\triangleright	$:$	:	$\bigtriangledown$	\bigtriangledown

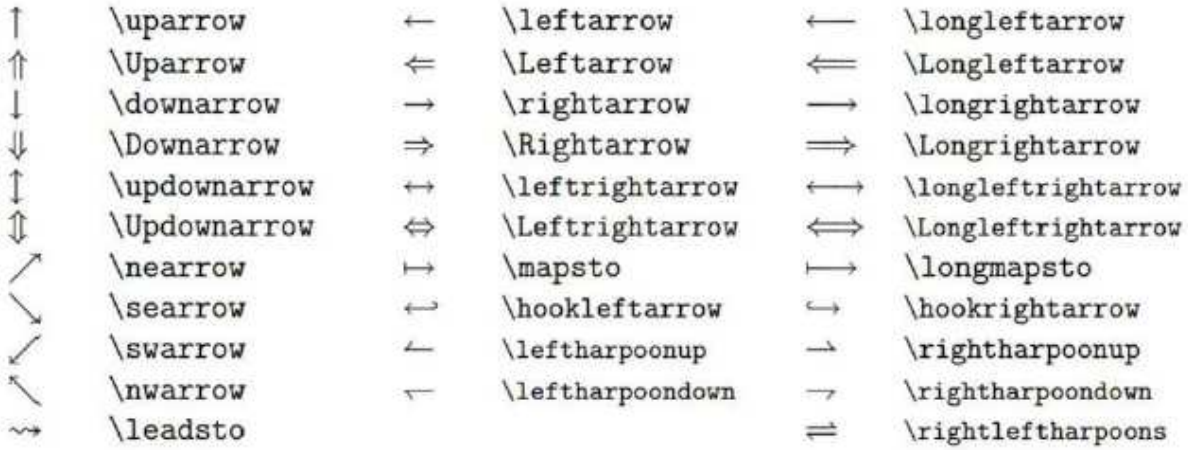
Рисунок 5.19 – Таблица символов



$<$	<	$>$	>	$=$	=
$\leq$	\le	$\geq$	\ge	$\equiv$	\equiv
$\prec$	\prec	$\succ$	\succ	$\sim$	\sim
$\preceq$	\preceq	$\succeq$	\succeq	$\simeq$	\simeq
$\ll$	\ll	$\gg$	\gg	$\approx$	\approx
$\subset$	\subset	$\supset$	\supset	$\cong$	\cong
$\subseteq$	\subseteq	$\supseteq$	\supseteq	$\doteq$	\doteq
$\sqsubset$	\sqsubset	$\sqsupset$	\sqsupset	$\asymp$	\asymp
$\sqsubseteq$	\sqsubseteq	$\sqsupseteq$	\sqsupseteq	$\bowtie$	\bowtie
$\in$	\in	$\ni$	\ni	$\Join$	\Join
$\smile$	\smile	$\frown$	\frown	$\vdash$	\vdash
$\mid$	\mid	$\parallel$	\parallel	$\dashv$	\dashv
$\propto$	\propto	$\perp$	\perp	$\models$	\models

$\not<$	\not<	$\not>$	\not>	$\neq$	\not=
$\not\leq$	\not\le	$\not\geq$	\not\ge	$\not\equiv$	\not\equiv
$\not\prec$	\not\prec	$\not\succ$	\not\succ	$\not\sim$	\not\sim
$\not\preceq$	\not\preceq	$\not\succeq$	\not\succeq	$\not\simeq$	\not\simeq
$\not\subset$	\not\subset	$\not\supset$	\not\supset	$\not\approx$	\not\approx
$\not\subseteq$	\not\subseteq	$\not\supseteq$	\not\supseteq	$\not\cong$	\not\cong
$\not\sqsubseteq$	\not\sqsubseteq	$\not\sqsupseteq$	\not\sqsupseteq	$\not\asymp$	\not\asymp

Рисунок 5.20 – Таблица символов



$\uparrow$	\uparrow	$\leftarrow$	\leftarrow	$\longleftarrow$	\longleftarrow
$\Uparrow$	\Uparrow	$\Leftarrow$	\Leftarrow	$\Longleftarrow$	\Longleftarrow
$\downarrow$	\downarrow	$\rightarrow$	\rightarrow	$\longrightarrow$	\longrightarrow
$\Downarrow$	\Downarrow	$\Rightarrow$	\Rightarrow	$\Longrightarrow$	\Longrightarrow
$\updownarrow$	\updownarrow	$\leftrightarrow$	\leftrightarrow	$\longleftrightarrow$	\longleftrightarrow
$\Updownarrow$	\Updownarrow	$\Leftrightarrow$	\Leftrightarrow	$\Longleftrightarrow$	\Longleftrightarrow
$\nearrow$	\nearrow	$\mapsto$	\mapsto	$\longmapsto$	\longmapsto
$\searrow$	\searrow	$\hookleftarrow$	\hookleftarrow	$\hookrightarrow$	\hookrightarrow
$\swarrow$	\swarrow	$\leftharpoonup$	\leftharpoonup	$\rightharpoonup$	\rightharpoonup
$\nwarrow$	\nwarrow	$\leftharpoondown$	\leftharpoondown	$\rightharpoondown$	\rightharpoondown
$\leadsto$	\leadsto			$\rightleftharpoons$	\rightleftharpoons

Рисунок 5.21 – Таблица символов

5. Задания

Используя документ из предыдущей работы, оформить все формулы из статьи.

6. Контрольные вопросы

1. Чем отличается внутренняя формула от выключенной?
2. Перечислите команды для оформления индексов, дробей, корней.
3. Как правильно оформить скобки?
4. Какие интегралы можно корректно отобразить в LaTeX?
5. Укажите преимущества использования окружения equation.

Лабораторная работа №6. Рисунки в LATEX

1. Основные сведения

Рисунки вставляются в документ при помощи окружения figure:

```
\begin{figure}[!ht]
\begin{center}
\includegraphics[scale=0.5]{images/figure.eps}\caption{Пример рисунка}\label{figure1}
\end{center}
\end{figure}
```

Здесь необязательный параметр окружения figure указывает TEX, как расположить рисунок. Может принимать следующие значения:

h - "печатать прямо здесь";

b - "печатать по нижнему краю страницы";

t - "печатать вверху страницы";

p - "разместить на отдельной странице, целиком отведённой под иллюстрации".

Параметры можно комбинировать, как показано в примере выше. Следует помнить, что указание, данное в квадратных скобках, носит рекомендательный характер. Чтобы повысить вероятность того, что TEX воспользуется данной рекомендацией, в список параметров нужно вставить символ "!" так, как показано в примере выше.

Рисунок еще можно вставлять в документ при помощи команды `\includegraphics` из пакета graphicx. Необязательный параметр scale указывает масштабирование рисунка. Например, значение 0.5 заставляет TEX уменьшить рисунок в два раза. При необходимости можно явно задавать высоту и ширину рисунка при помощи необязательных параметров height и width.

Команда `\caption` задает подпись под рисунком. Если необходимо поместить метку рисунка, то используется команда `\label` после команды `\caption`.

Окружение center предписывает LATEX центрировать рисунок.

Основным форматом рисунков в TEX является формат EPS. В LATEX существуют инструменты, реализующие возможность использования иных форматов (JPG, TIFF и пр.). Например, инструмент pdflatex позволяет использовать рисунки в формате PDF. Мы будем использовать только формат EPS.

EPS – Стандартный формат для импорта-экспорта PostScript файлов между различными приложениями. Целью EPS файла является его включение, «инкапсуляция» вовнутрь другого PostScript файла. EPS-файл может содержать произвольную комбинацию текста, графики, растровых изображений, как произвольный PostScript файл, но только с некоторыми ограничениями.

Для конвертирования рисунков в формат EPS используются следующие инструменты:

- Gimp
- AdobePhotoshop
- sam2p (используется в режиме командной строки)
- bmeps (входит в состав MikTeX 2.5, доступна из командной строки).

FAQ по использованию утилиты sam2p можно найти здесь.

Для конвертирования рисунков при помощи утилиты sam2p можно использовать следующую команду:

sam2p.exe -c:none -t:a85 -s:gray4 <входной файл>EPS2: <выходной файл>

2. Задание

Используя документ из предыдущей работы, оформить все рисунки (изображения) из статьи.

3. Контрольные вопросы

1. Что такое окружение figure в LaTeX?
2. Что делает необязательный параметр у окружения figure и какие значения он может принимать?
3. Как комбинировать параметры окружения figure?
4. Какие аргументы передаются в команду `\includegraphics` для вставки изображения?

Лабораторная работа №7. Таблицы LATEX

1. Основные сведения

Таблица – это одна из наиболее простых конструкций для оформления структурированных данных. Текст, оформленный с помощью таблицы, существенно проще воспринимать и анализировать. И зачастую содержимое таблиц описано более лаконично, избегаются повторы текстовой информации.

За оформление таблиц и похожих структур в LaTeX отвечают два стандартных окружения: tabbing и tabular.

1.1. Окружение tabbing

Указанное окружение предназначено для табулирования текстовой информации. Этот прием хорошо известен (рис.7.1), например, активным пользователям MS Word: с помощью специальных символов табуляции можно устанавливать соответствующее положение курсора в строке или выравнивать содержимое нескольких строк по указанному отступу.

№	ФИО	Возраст	Стаж (лет)
1	Иванов П.М.	45	17
2	Петров Д.А.	56	28
3	Егоров	48	21

Рисунок 7.1 – Табулирования текстовой информации в MS Word

Именно поэтому табулятор нельзя назвать инструментом для полноценной верстки таблицы: он лишь позволяет имитировать табличное оформление.

Окружение tabbing поддерживает следующие команды (табл. 7.1):

Таблица 7.1 – Команды, поддерживаемые окружением tabbing

Команда	**Описание**
\=	Задает позицию табулятора (курсора). Дополнительно можно указать пробел: \hspace{x}\=
\kill	Убирает строку. Как правило, это первая (форматирующая) строка, которая указывает шаблон отступов
\>	Сдвигает курсор к следующей позиции
\\	Начало новой строки
*	Запрет разрыва строки
\’	Выравнивание вправо (относительно табулятора)
\+	Сдвигает все позиции вправо
\-	Сдвигает все позиции влево

Пример:

```
\documentclass[12pt]{article}
\usepackage[T2A]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage[english,russian]{babel}
\linespread{1.3}
\begin{document}
\begin{tabbing}
MMM \= MMMMMMMMMMM \= MMMMM \= MMMMMMM \kill
№ \> ФИО \> Возраст \> Стаж (лет)\\
1 \> Иванов П.М. \> 45 \> 17\\
2 \> Петров Д.А. \> 56 \> 28\\
3 \> Егоров К.А. \> 48 \> 21
\end{tabbing}
\begin{tabbing}
\hspace{3em}\= \hspace{12em}\= \hspace{6em}\=
\hspace{6em}\kill
№ \> ФИО \> Возраст \> Стаж (лет)\\
1 \> Иванов П.М. \> 45 \> 17\\
```

```
2 \> Петров Д.А. \> 56 \> 28\\
3 \> Егоров К.А. \> 48 \> 21
\end{tabbing}
\end{document}
```

Результат (рис. 7.2):

№	ФИО	Возраст	Стаж (лет)
1	Иванов П.М.	45	17
2	Петров Д.А.	56	28
3	Егоров К.А.	48	21

№	ФИО	Возраст	Стаж (лет)
1	Иванов П.М.	45	17
2	Петров Д.А.	56	28
3	Егоров К.А.	48	21

Рисунок 7.2 – Результат выполнения примера

В первом примере шаблон задается с помощью устраняемой строки, где буква M резервирует позицию одного символа (можно было указать любой другой символ). Во втором примере расстояние между табуляторами устанавливается горизонтальным пробелом.

Перечислим особенности и ограничения окружения tabbing:

1. Позиции всех табуляторов и отступов должны быть заданы пользователем явно, иначе текст соседних табуляторов может перекрывать его.

2. Табуляторы работают только для текстовой разметки.

3. Разделительных линий (границ) tabbing не поддерживает.

4. Вложение также не поддерживается.

5. При необходимости содержимое «колонок» переносится по строкам на следующую станицу.

1.2. Окружение tabular

Для отрисовки таблицы с границами используется окружение tabular. Основным аргументом окружения является команда указания колонки, которая также определяет и способ выравнивания: l (по левому краю), c (по центру), r (по правому краю). Вертикальную линию позволяет рисовать команда |. Например,

```
\begin{tabular}{l|r|cc}
```

отрисовывает таблицу в четыре колонки, первая колонка выравнивается по левому краю, вторая – по правому, третья и четвертая – по центру; вертикальные границы есть только у второй колонки.

Содержимое таблицы формируется построчно. Содержимое ячеек в строке разделяется символом &, а каждая строка завершается командой разрыва строки \\. Горизонтальную линию позволяет сделать команда `\hline`.

Автоматически масштабируемые таблицы

По умолчанию LaTeX подгоняет ширину колонок по содержимому ячеек.

Пример:

```
\documentclass[12pt]{article}
\usepackage[T2A]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage[english,russian]{babel}
\linespread{1.3}
\begin{document}
\begin{tabular}{l||l||cc}
\hline
№ & ФИО & Возраст & Стаж (лет)\\
\hline\hline
1 & Иванов П.М. & 45 & 17 \\
2 & Петров Д.А. & 56 & 28 \\
```

```
3 & Егоров К.А. & 48 & 21
\end{tabular}
\end{document}
```

Результат (рис. 7.3):

№	ФИО	Возраст	Стаж (лет)
1	Иванов П.М.	45	17
2	Петров Д.А.	56	28
3	Егоров К.А.	48	21

Рисунок 7.3 – Результат выполнения примера

Ячейки можно оставлять пустыми.

Пример:

```
\begin{tabular}{l||l||cc}
\hline
& ФИО & Возраст & Стаж (лет)\\
\hline\hline
1 & Иванов П.М. && 17 \\
2 & Петров Д.А. & 56 & 28 \\
3 & Егоров К.А. &&
\end{tabular}
```

Результат (рис.7.4):

	ФИО	Возраст	Стаж (лет)
1	Иванов П.М.		17
2	Петров Д.А.	56	28
3	Егоров К.А.		

Рисунок 7.4 – Результат выполнения примера

Таблицы с фиксированной шириной колонок

Если ширина колонок должна быть задана точно, используется команда

```
\begin{tabular}{p ширина}
```

которую требуется указать для каждой колонки. Команды выравнивания при этом не работают: оно осуществляется по ширине.

Пример:

```
\begin{tabular}{p{1cm}||p{5cm}||p{2.5cm}p{2.5cm}}
\hline
№ & ФИО & Возраст & Стаж (лет)\\
\hline\hline
1 & Иванов П.М. & 45 & 17 \\
2 & Петров Д.А. & 56 & 28 \\
3 & Егоров К.А. & 48 & 21
\end{tabular}
```

Результат (рис. 7.5):

№	ФИО	Возраст	Стаж (лет)
1	Иванов П.М.	45	17
2	Петров Д.А.	56	28
3	Егоров К.А.	48	21

Рисунок 7.5 – Результат выполнения примера

Особенности окружения tabular

1. Колонки растягиваются пропорционально содержимому.

2. Допускается горизонтальное выравнивание для каждой колонки.

3. Таблицы могут быть вложены.

4. Таблицы не переносятся частями на другую страницу: для этого потребуется подключение дополнительных пакетов.

Разумеется, это далеко не все возможности LaTeX в вопросе верстки таблиц. Допускается объединение ячеек, настройка их фона и других свойств. Делается это, в том числе с помощью дополнительных пакетов.

2. Задание

Используя документ из предыдущей работы, оформить простые таблицы из статьи. Если в статье отсутствуют простые таблицы, то создайте в основной части статьи из любого абзаца таблицу, согласно примеру (табл. 7.2).

Таблица 7.2 – Пример таблицы

№	Предложение
1	/первое предложение из абзаца/
2	/второе предложение из абзаца/
…	…

Названием таблицы указать название раздела статьи, из которой взят абзац.

3. Контрольные вопросы

1. Какие стандартные окружения в LaTeX используются для оформления таблиц?
2. Что такое окружение tabbing в LaTeX и для чего оно используется?
3. Какие команды поддерживает окружение tabbing?
4. В чем разница между окружением tabbing и окружением tabular в LaTeX?
5. Что такое окружение tabular в LaTeX и для чего оно используется?
6. Как команда `\hline` создает горизонтальные линии в LaTeX?
7. Можно ли оставлять ячейки таблицы пустыми в LaTeX?

Лабораторная работа №8. Сложные таблицы в LATEX

1. Основные сведения

Для того чтобы объединять ячейки из соседних столбцов, нужно использовать команду, объединяющую, начиная с данного, N столбцов:

```
\multicolumn{число N столбцов}{опции}{содержимое}
```

Для объединения ячеек из соседних строк, можно использовать пакет multirow. Это позволит создать ячейки, занимающие несколько строк, применяя команду:

```
\multirow{число N строк}{ширина}{содержимое}
```

Создадим простую таблицу из 3-х строк и 3-х столбцов (рис. 8.1), обозначая для определенности ячейки буквами от «A» до «M».

```
\begin{tabular}{|c|c|c|}
\hline
A & B & C \\ \hline
D & E & F \\ \hline
K&L&M \\ \hline
\end{tabular}
```

A	B	C
D	E	F
K	L	M

Рисунок 8.1 – Результат выполнения примера

Выполним объединение ячеек из соседних строк и столбцов этой таблицы (рис. 8.2).

```
\begin{tabular}{|c|c|c|} \hline
\multirow{2}{*}{ЯчейкиАи D}
& \multicolumn{2}{c|}{Ячейки B и C} \\ \cline{2-3}
& E & \multirow{2}{*}{Ячейки F и M} \\ \cline{1-2}
```

```
K& L & \\ \hline
\end{tabular}
```

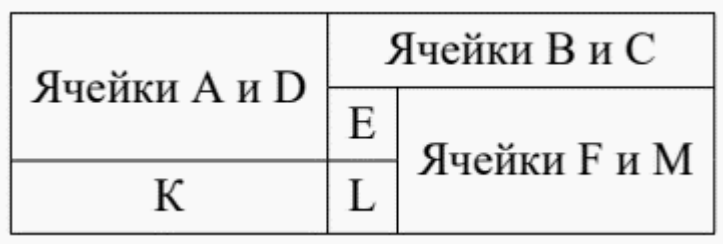


Рисунок 8.2 – Результат выполнения примера

Для одновременного объединения ячеек из соседних строк и столбцов в этой таблице требуется последовательно применить команды \multicolumn и \multirow , вложив одну в другую (рис. 8.3).

```
\begin{tabular}{|c|c|c|} \hline
A& \multicolumn{2}{c|}{ЯчейкиВиС} \\ \cline{2-3}
D& \multicolumn{2}{c|}{\multirow{2}{*}{ЯчейкиE, F,
LиM}} \\ \cline{1-1}
K& \multicolumn{2}{c|}{} \\ \hline
\end{tabular}
```

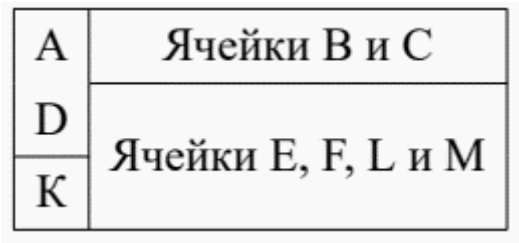


Рисунок 8.3 – Результат выполнения примера

Команда `\multicolumn{2}{c|}{}` в последней ячейке таблицы нужна для проведения правой вертикальной линии на месте исходной ячейки «M».

2. Задание

Используя документ из предыдущей работы, оформить сложные таблицы из статьи. Если в статье отсутствуют простые таблицы, то создайте в основной части статьи из любого абзаца таблицу, согласно примеру (табл. 8.1).

Таблица 8.1 – Пример таблицы

№	Предложение
1	/первое предложение из абзаца/
	/второе предложение из абзаца/
2	/третье предложение из абзаца/
	/четвертое предложение из абзаца/
…	…

Названием таблицы указать название раздела статьи, из которой взят абзац.

3. Контрольные вопросы

1. Как в LaTeX объединять ячейки из соседних столбцов?
2. Для чего используется пакет multirow в LaTeX?
3. Какую команду нужно использовать для объединения ячеек из соседних строк в LaTeX?
4. Как создать простую таблицу из трех строк и трех столбцов в LaTeX?
5. Каким образом выполнить объединение ячеек из соседних строк и столбцов таблицы в LaTeX?

Литература

1. Кузнецов А.В. Основы LATEX. Учебное пособие. М.: НИЯУМИФИ, 2021. 364 с.
2. Якубович, Д. А. Издательская система LaTeX [Электронный ресурс] : учеб.пособие / Д. А. Якубович, Е. С. Еропова ; Владим. гос. ун-т им. А. Г. и Н. Г. Столетовых. – Владимир : Изд-во ВлГУ, 2019. –327 с.
3. Насыров, В. В. Пакеты прикладных программ для физиков: LATEX : учебное пособие / В. В. Насыров ; [научный редактор А. И. Мазур]. — Хабаровск : Изд-во Тихоокеан. гос. ун-та, 2019. — 84 с.
4. Львовский С.М. LATEX: подробное описание. 2005. – 304 с.
5. Львовский С.М. Набор и вёрстка в системе LATEX. 2003. – 448 с.

Электронное учебное издание

Мария Александровна **Маслова**

УЧЕБНО-МЕТОДИЧЕСКОЕ ПОСОБИЕ ПО ЛАБОРАТОРНЫМ РАБОТАМ ПО ДИСЦИПЛИНЕ «ТЕХНОЛОГИЯ ПОДГОТОВКИ ВКР»

Учебное пособие

Электронное издание сетевого распространения

Редактор Матвеева Н.И.

Темплан 2024 г. Поз. № 7.
Подписано к использованию 13.05.2024. Формат 60х84 1/16.
Гарнитура Times. Усл. печ. л. 4,0.

Волгоградский государственный технический университет.
400005, г. Волгоград, пр. Ленина, 28, корп. 1.

ВПИ (филиал) ВолгГТУ.
404121, г. Волжский, ул. Энгельса, 42а.