

Н.Т. Суханова

**ПОДГОТОВКА ТЕКСТОВЫХ ДОКУМЕНТОВ С  
ПОМОЩЬЮ ИЗДАТЕЛЬСКОЙ СИСТЕМЫ LaTeX**

*Учебное пособие*

Нижний Новгород  
2023

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение высшего образования  
«Нижегородский государственный архитектурно-строительный университет»

---

Н.Т. Суханова

ПОДГОТОВКА ТЕКСТОВЫХ ДОКУМЕНТОВ С  
ПОМОЩЬЮ ИЗДАТЕЛЬСКОЙ СИСТЕМЫ LaTeX

Утверждено редакционно-издательским советом университета  
в качестве учебно-методического пособия

Нижегород  
ННГАСУ  
2023

ББК 32.973.2р3  
С 91  
УДК 681.3.06 (07)

Рецензенты:

*С.В. Болдин* – к.т.н., доцент кафедры ПИИТО (ФГБОУ ВО «Нижегородский государственный педагогический университет им. Козьмы Минина»)  
*А.В. Поначугин* – к.э.н., доцент кафедры ПИИТО (ФГБОУ ВО «Нижегородский государственный педагогический университет им. Козьмы Минина»)

Суханова Н.Т. Подготовка текстовых документов с помощью издательской системы LaTeX [Текст]: учеб. пособие / Н.Т. Суханова; Нижегород. гос. архитектур.- строит. ун-т. – Н. Новгород: ННГАСУ, 2023. – 124 с. ISBN 978-5-528-00517-1

Учебное пособие содержит краткие теоретические сведения о технологии подготовки текстовых документов LaTeX естественно-научного характера в интегрированной среде TeXstudio. Приводятся практические задания, примеры выполнения практических заданий и задания для самостоятельной работы.

Пособие рекомендовано для студентов, обучающихся в Нижегородском государственном архитектурно-строительном университете при изучении дисциплины Технологии обработки информации, аспирантов и преподавателей. Может представлять интерес для лиц, желающих освоить технологии подготовки документов LaTeX.

ББК 32.973.2р3

ISBN 978-5-528-00517-1

© Н.Т. Суханова, 2023  
©ННГАСУ, 2023

## Предисловие

Во многих технических сообществах: компьютерных, математических, физических, химических и пр., общепризнанно, что LaTeX является наиболее качественной системой подготовки научных печатных публикаций и является своеобразным стандартом в этой области. Огромное количество книг, научных и научно-технических журналов издаются с помощью LaTeX.

LaTeX представляет собой издательскую систему, включающую набор программных пакетов. С позиций конечного пользователя LaTeX представляет собой систему, состоящую из свода правил и команд, с помощью которых исходный текст документа преобразуется в окончательную печатную версию.

С помощью LaTeX возможно автоматизировать многие аспекты верстки документов, такие как нумерация страниц, таблицы, иллюстрации, формулы, колонтитулы, сноски, ссылки, предметный указатель, оглавление и многое другое. Помимо этого имеется большой набор макросов, облегчающих создание сложных документов.

Однако LaTeX не очень прост в освоении, но обладает массой преимуществ перед популярными текстовыми процессорами и редакторами презентаций.

LaTeX-документ – это обычный текстовый файл, в котором так же содержится и некоторый объем команд для LaTeX процессора. Созданный с её помощью документ в определенном смысле является программой, в результате выполнения которой получается качественно оформленный документ.

## Введение

Представленное вниманию читателей учебное пособие «Подготовка текстовых документов с помощью издательской системы LaTeX» ориентировано на студентов, обучающихся в Нижегородском государственном архитектурно-строительном университете при изучении дисциплин Технологии обработки информации.

Основная цель учебного пособия – формирование практических навыков создания документов естественно-научного характера, в том числе курсовых и выпускных квалификационных работ, научных статей и разработки презентаций. При этом в качестве среды разработки используется TeXstudio.

В пособии представлены восемь лабораторных работ, позволяющих приобрести базовые навыки создания документов LaTeX. Тематика лабораторных работ следующая: «Основные приемы набора текста LaTeX», «Создание и редактирование списков и таблиц», «Форматирование текстового документа LaTeX», «Создание математических формул», «Титульный лист, подчеркивание, рамки, переносы», «Рисунки и графика», «Сервисы для документов LaTeX», «Презентации LaTeX».

Структурно каждая лабораторная работа имеет цель, сведения из теории, необходимые для выполнения заданий, примеры и варианты практических заданий.

Целью данного пособия является приобщение студентов к процессу разработки текстового документа LaTeX, наполненного формулами, таблицами, графиками и прочими элементами, а также разработки презентаций на качественно новом уровне.

Учебно-методическое пособие ориентировано главным образом на использование студентами, обучающимися по направлению подготовки 09.03.02 Информационные системы и технологии в рамках изучения курса Технологии обработки информации. Однако может быть использовано студентами других специальностей, магистрантами, аспирантами, преподавателями в процессе подготовки LaTeX документов.

В основу учебного пособия положены материалы лабораторных работ по дисциплине Технологии обработки информации, читаемой автором в течение нескольких лет в Нижегородском государственном архитектурно-строительном университете.

## Лабораторная работа 1

### Основные приемы набора текста LaTeX

**Цель:** создание файла-заготовки документа, использование спецсимволов, структура документа LaTeX, преамбула и её расширенные возможности, шрифтовое оформление текста, нумерованные списки, оглавление

#### Теоретические сведения

**LaTeX** – издательская система, предназначенная для набора математических текстов с помощью технологии **TeX**. LaTeX предоставляет удобные и гибкие средства, позволяющие достичь полиграфического качества документа. Например, можно указать логическую структуру текстового документа с помощью простых средств, не вникая в детали его создания. Программа дает возможность создавать математические формулы высокого качества.

#### Подготовка документа

Создадим **файл-заготовку** LaTeX или, как его еще называют, **входной файл**. Входной файл и есть ваш документ, написанный по правилам LaTeX. Текст для LaTeX можно набирать в любом текстовом редакторе. Текст состоит из обычного текста, спецсимволов и команд LaTeX, по которым происходит размещение текста и формул на странице рукописи. Текст не должен содержать шрифтовых выделений, разбивки на страницы и т.д.

#### Команды и их задание в тексте

С точки зрения записи в исходном тексте команды делятся на два типа:

1) команды, состоящие из знака \ и одного символа после него, не являющегося буквой

2) команды, состоящие из \ и последовательности букв, называемой *именем команды* (имя может состоять и из одной буквы). Например, команды

`\TeX` и `\LaTeX` генерируют эмблемы систем TeX и LaTeX соответственно. В имени команды, а также между `\` и именем не должно быть пробелов; имя команды нельзя разрывать при переносе на другую строку.

**В именах команд прописные и строчные буквы различаются.** Например, `\large`, `\Large` и `\LARGE` – это три разные команды.

После команды первого типа пробел в исходном тексте ставится или не ставится в зависимости от того, что вы хотите получить на печати.

**В чем разница между `$1` и `$ 1`?      В чем разница между `\$1 b` и `\$ 1`?**

После команды второго типа обязательно должен стоять либо пробел, либо символ, не являющийся буквой (это необходимо, чтобы TeX смог определить, где кончается имя команды и начинается дальнейший текст).

Пример с командой `\s1` (она переключает шрифт на наклонный):

*38 попугаев*

`\s138` попугаев

*Подарок мартышке*

`\s1` Подарок мартышке

**Группы** – важнейшее понятие TeXa – это часть текста, к которому применено определенное форматирование. Синтаксически задается фигурными скобками, *внутри* которых дана определенная команда. После закрывающейся фигурной скобки TeX «забудет» о всех настройках внутри них и восстановит те параметры, которые были до скобок.

**Например**, нужно напечатать часть текста полужирным шрифтом (команда `\bf`):

<p>Полужирным шрифтом набрано только <b>это</b> слово; после скобок все идет как прежде.</p>	<p>Полужирным шрифтом набрано только <code>{\bf это}</code> слово; после скобок все идет как прежде</p>
--	---

Сами по себе фигурные скобки не генерируют никакого текста и не влияют на шрифт; единственное, что они делают – это ограничивают *группу* внутри файла; команда, а не скобка переключает шрифт. Группы могут быть вложены друг в друга.

## Спецсимволы

Большинство символов в исходном тексте прямо обозначают то, что будет напечатано (если в исходном тексте стоит запятая, то и на печати выйдет запятая). Следующие 10 символов: `{ } $ & # % _ ^ ~ \` имеют особый статус; если вы употребите их в тексте «просто так», то получите сообщение об ошибке. Печатное изображение знаков, соответствующих первым семи из них, можно получить, если в исходном тексте поставить перед соответствующим символом без пробела знак `\` (**backslash**).

### Например:

Курс тугрика повысился на 7%, и теперь за него дают \$200.	Курс тугрика повысился на 7\%, и теперь за него дают \\$200.
---	---

**Фигурные скобки** ограничивают *группы* в исходном файле.

**Знак доллара** ограничивает математические формулы. При наборе математических же формул используются знаки `_ ^ < > |`.

**Знак ~** означает «неразрывный пробел» между словами.

Если символ `%` употреблен в тексте не в составе комбинации `\%`, то он является символом комментария: все символы, расположенные на строке после него, LaTeX игнорирует (в том числе и сам `%`). С помощью символа `%` в исходный текст можно вносить пометки «для себя».

Каждый *входной файл* LaTeX имеет следующую *структуру*:

```
\documentclass[list]{style}
```

    преамбула

```
\begin{document}
```

    текст документа

```
\end{document}
```

`\documentclass` – команда, обязательная в каждом входном файле LaTeX.

В этой команде:

1) **list** – список из одной или нескольких опций стиля документа, разделяемых запятыми без пробелов. Стандартные опции LaTeX:

- **11pt, 12pt** – устанавливается высота шрифта в пунктах
- размер листа бумаги рукописи, например, **a4paper**
- **twoside** – различает страницы с нечетными и четными номерами. Например, различаются величины левых и правых полей.
- **twocolumn** – текст выводится в двух колонках
- **fleqn** – используется в математических формулах
- **leqno** - номера формул по умолчанию ставятся справа от формулы. При установке этой опции номера будут ставиться слева от формулы.

2) опция **style** задает основной стиль документа: **article, book, report**.

**Book** означает, что документ будет оформлен, как книга: все главы будут начинаться с нечетных страниц, текст будет снабжен колонтитулами некоторого определенного вида и т.п.

**Article** – служит для оформления статей в журналах, создания коротких отчетов.

**Report** – среднее между **article** и **book**. Служит для создания более длинных отчетов, содержащих несколько глав.

Реже используются **proc** (оформление «труды конференции»), **letter** (оформление деловых писем).

В преамбуле приводятся команды, относящиеся к разметке и оформлению всего документа, например, устанавливающие параметры оформления текста.

На месте «*текст документа*» пишется сам текст рукописи. Команды **\begin{document}** и **\end{document}** обязательны, они служат отметками начала и конца рукописи. Если даже после **\end{document}** в файле написано еще что-то, LaTeX это проигнорирует.

### Окружения

*Окружение* – это фрагмент файла, имеющий следующую конструкцию:

**\begin{имя\_окружения}**

текст

`\end{имя_окружения}`

В зависимости от окружения, часть текста наделяется определенными свойствами.

**Пример** минимального набора команд преамбулы, позволяющий набирать текст, большое количество формул и выводить графику:

`\documentclass[12pt]{article}`

`\usepackage{amsmath}`

`\usepackage{amssymb}`

`\usepackage{russian}`

`\usepackage{graphicx}`

`\textheight=240mm`

`\textwidth=170mm`

### Расширения LaTeX

Существует много расширений основных возможностей LaTeX-а. Такие расширения сосредоточены в текстовых стилевых файлах, которые обычно поставляются вместе с программой LaTeX. Дополнительные возможности LaTeX содержатся в файлах расширения, названных пакетами или **packages**. Эти файлы имеют расширение **.sty** или **.cls**. Для загрузки пакета применяется команда `\usepackage{имя пакета, которую необходимо разместить в преамбуле}`.

**Например:**

`\usepackage{amsmath}` – загрузка пакета математики

`\usepackage{amssymb}` – загрузка пакета символов (при этом станут доступны все символы закладок **Arrows, AmsSumb, AmsRets**)

`\usepackage{russian}` – для оформления документа по-русски

Лучше соблюдать указанный порядок подключения пакетов.

**Шрифт.** В процессе создания документа в среде LaTeX текст набирается, как правило, текущим шрифтом (обычно это шрифт **roman**). Для переключения текущего шрифта на *наклонный* нужно использовать команду `\slshape`, а

`\upshape` выполняет обратное переключение. Аналогично *жирность шрифта* можно изменить командами `\bfseries` и `\mdseries`.

Но можно обойтись команд `\upshape` и `\mdseries`, которые отменяют действия предыдущих команд. Для этого часть текста, которую нужно оформить полужирным или наклонным шрифтом, можно заключить в фигурные скобки и дать команду `\bfseries` или `\slshape` *внутри* скобок. При этом сразу после закрывающей фигурной скобки текст будет продолжать набираться тем шрифтом, который был до скобок.

Команд `\itshape` переключает шрифт на *курсивный*. Чтобы **подчеркнуть текст**, используется команда `\underline`. Её обязательный аргумент – подчеркиваемый текст. Альтернативный способ задания команд, переключающих начертание текста представлен в таблице 1.

Таблица 1

### Команды для изменения начертания текста

Команда	Название начертания
<code>\bf</code>	Полужирный шрифт (boldface)
<code>\it</code>	Курсив (italic)
<code>\sl</code>	Наклонный шрифт (slanted)
<code>\sf</code>	Рубленый шрифт (sansserif)
<code>\sc</code>	Капитель (SmallCaps)
<code>\tt</code>	Имитация пишущей машинки (typewriter)
<code>\rm</code>	Прямой светлый шрифт (roman)

Команды, меняющие размер шрифта (Таблица 2), одновременно устанавливают начертание roman. Поэтому полужирный большой шрифт требует команды `\large\bf`.

Таблица 2

### Команды для изменения размера шрифта

Команда	Название размера
<code>\tiny</code>	Малюсенький
<code>\scriptsize</code>	Очень малюсенький (как индексы)
<code>\footnotesize</code>	Маленький (как сноски)
<code>\small</code>	Мелкий
<code>\normalsize</code>	Нормальный
<code>\large</code>	Большой

<b>Команда</b>	<b>Название размера</b>
<code>\Large</code>	Очень большой
<code>\LARGE</code>	Совсем большой
<code>\huge</code>	Громадный
<code>\Huge</code>	грандиозный

### **Счетчики. Рубрикация документа**

*Счетчики* – это переменные с целыми значениями, контролирующие количество страниц, глав, частей, параграфов и прочее документа. Команда `\setcounter{счетчик}{n}` изменяет содержание счетчика, присваивает счетчику значение n.

Система LaTeX позволяет автоматически объявлять и нумеровать части, главы, параграфы, пункты, подпункты в документе. Для этого используются команды рубрикации, которые размещают в начале соответствующего раздела документа. Параметром каждой команды является название соответствующего раздела документа. Такими командами являются команды объявления:

части документа – `\part{название части}`

главы – `\chapter{название главы}` (эта команда доступна только в стилях

**book** и **report**, в стиле **article** ее нет

параграфа – `\section{название секции}`

подпараграфа – `\subsection{название подсекции}`

подподпараграфа – `\subsubsection{название подподсекции}`

Нумерация каждого раздела начинается с 1. Можно самостоятельно устанавливать значения номеров. Например, `\setcounter{section}{15}` присваивает номеру текущего счетчика параграфа 15. Следующий параграф будет иметь номер 16.

### **Практические задания**

#### **Задание 1. Познакомиться со справочной информацией LaTeX.**

1. Открыть справочную информацию по работе с системой LaTeX. Для этого выполнить команду **Справка – Справка LaTeX**. Познакомиться с главой 2 и 3. Изучить основы обработки текста из главы 4.

2. Открыть справку по работе с программой **LaTeX**, выполнив команду меню **Справка – Руководство пользователя**. Познакомьтесь с информацией из главы **2. Редактирование документа TeX**.

**Задание 2. Набрать статью в LaTeX стилем article и сохранить в .pdf формате.**

Стиль **article** характеризуется тем, что все страницы являются правосторонними и нечетными, а номер страницы находится в центре основания, если не загружен пакет, устанавливающий другое расположение нумерации страниц.

1. Создайте преамбулу входного документа стиля **article** (LaTeX это может сделать автоматически): **Файл (File) – Открыть/Новый (Open/New)**, ввести имя нового файла, кнопка **«Открыть»**, подтвердить создание нового файла (Рисунок 1).

После этого откроется новое рабочее окно редактора, в котором будет присутствовать заготовка документа. Видно, что в статье будут применяться пакеты **indentfirst** – автоматический отступ у первого абзаца параграфа. Будет загружен пакет **graphicx** – для ввода графики в текст.

Команда **\maketitle** выводит заглавие в стилях документа **article, book**.

Команда **\thanks{text}** может стоять внутри скобок **title** или скобок **author**.

Команда **\tableofcontents** автоматически генерирует оглавление, которое будет содержать названия всех частей документа.

2. Добавьте пакет **\usepackage{russian}** для оформления статьи по-русски

3. Добавьте пакет **\usepackage{color}** для выделения цветом части текста (порядок подключения пакетов лучше соблюсти как на рисунке 1)

4. Введите **название статьи, имя автора, название первого параграфа**

5. Добавьте благодарности или комментарии от имени автора командой **\thanks**

6. Добавьте аннотацию статьи командой **\abstract**

```

\documentclass[a4paper]{article}
\usepackage{russian}
\usepackage[indentfirst]{indentfirst} % Indent First Paragraph
\usepackage{color}
\usepackage{graphicx}

\title{Классическая механика}
\author{В.А.Андреев, Д.Ю.Поленов\thanks{Институт общей физики им. А.М.Прохорова РАН}}
\date{\today}
\begin{document}
\maketitle
\tableofcontents
\abstract{В статье рассматривается один из основных разделов теоретической физики}
\section{Классификация физических теорий}
Все физические теории можно разделить на два больших класса
\section{Предмет и методы классической механики}
Предметом классической механики являются медленные движения макроскопических тел.
\section{Ньютоновская механика}
В основу Ньютоновского построения классической механики положены три постулата
\end{document}

```

Рисунок 1. – Оформление текста документа в Задание 1

Создайте три параграфа командами `\section`. Результат можно посмотреть на рисунке 2. Команда `\section` означает «начать новый раздел документа», а команда `\section*` означает «начать новый раздел документа, не нумеруя его».

7. Наберите текст статьи (для получения абзаца, например, при создании нумерованного списка, надо предварить его пустой строкой или перед первой строкой абзаца поставить команду `\indent`. Расположение текста в файле-заготовке достаточно произвольное. Можно не следить за длиной строки. При написании входного файла не делайте переносов):

**Добавить информацию для документа задания 1:**

**`\section{Классификация физических теорий}`**

Все физические теории можно разделить на два больших класса:

1) Теории, рассматривающие изменение состояния объектов без определения природы этих изменений. Эти теории охватываются понятием механика.

2) Теории, изучающие природу воздействия на объекты со стороны других объектов. Эти теории называют теориями поля.

### **\section{Предмет и методы классической механики}**

Предметом классической механики являются медленные движения макроскопических тел.

Основными методами классической механики являются:

1. Ньютоновская механика
2. Лагранжев формализм
3. Гамильтонов формализм
4. Метод Гамильтона-Якоби

### **\section{Ньютоновская механика}**

В основу ньютоновского построения классической механики заложены три постулата:

- Первый постулат: существуют локально-инерциальные системы отсчета.
- Второй постулат определяет вид уравнений движения.
- Согласно третьему постулату сумма всех сил взаимодействия в замкнутой механической системе равна нулю.

5. Сохраните документ в формате tex.

6. Выполнить компиляцию документа – команда **Инструменты** –

**Компиляция & Просмотр (F5)** или кнопка на панели инструментов .

7. Вернитесь в файл-заготовку, установите величину абзацного отступа командой **\parindent=1in** (установку величины отступа можно сделать в любом месте файла или в преамбуле, если отступ в документе предполагается единым Рисунок 2).

## Классическая механика

В.А. Андреев

15 ноября 2020 г.

## Содержание

1	Классификация физических теорий	1
2	Предмет и методы классической механики	1
3	Ньютоновская механика	1

## Аннотация

В статье рассматриваются один из основных разделов теоретической физики

## 1 Классификация физических теорий

Все физические теории можно разделить на два больших класса

## 2 Предмет и методы классической механики

Предметом классической механики являются медленные движения макроскопических тел.

## 3 Ньютоновская механика

В основу Ньютоновского построения классической механики положены три постулата

Рисунок 2. – Результат компиляции документа, представленного в первом задании

## Задания для самостоятельной работы

## Вариант 1

Лекция  
Введение

В данной лекции вводится понятие операционной системы; рассматривается эволюция развития операционных систем; описываются функции операционных систем и подходы к построению операционных систем.

**1. Что такое операционная система****1.1. Структура вычислительной системы****1.2. Что такое ОС**

*1.2.1. Операционная система как виртуальная машина*

*1.2.2. Операционная система как менеджер ресурсов*

*1.2.3. Операционная система как защитник пользователей и программ*

*1.2.4. Операционная система как постоянно функционирующее ядро*

**1.3. Краткая история эволюции вычислительных систем**

- 1.3.1. *Первый период (1945–1955 гг.). Ламповые машины. Операционных систем нет*
- 1.3.2. *Второй период (1955 г.–начало 60-х). Компьютеры на основе транзисторов. Пакетные операционные системы*
- 1.3.3. *Третий период (начало 60-х – 1980 г.). Компьютеры на основе интегральных микросхем. Первые многозадачные ОС*
- 1.3.4. *Четвертый период (с 1980 г. по настоящее время). Персональные компьютеры. Классические, сетевые и распределенные системы*

#### **1.4. Основные понятия, концепции ОС**

- 1.4.1. *Системные вызовы*
- 1.4.2. *Прерывания*
- 1.4.3. *Исключительные ситуации*
- 1.4.4. *Файлы*
- 1.4.5. *Процессы, нити*

#### **2. Архитектурные особенности ОС**

- 2.1. Монолитное ядро**
- 2.2. Многоуровневые системы (Layered systems)**
- 2.3. Виртуальные машины**
- 2.4. Микроядерная архитектура**
- 2.5. Смешанные системы Классификация ОС**

### **Вариант 2**

#### **Лекция Процессы**

В лекции описывается основополагающее понятие процесса, рассматриваются его состояния, модель представления процесса в операционной системе и операции, которые могут выполняться над процессами операционной системой.

##### **1. Понятие процесса**

###### **1.1. Состояния процесса**

###### **1.2. Операции над процессами и связанные с ними понятия**

- 1.2.1. *Набор операций*
- 1.2.2. *Process Control Block и контекст процесса*
- 1.2.3. *Одноразовые операции*
- 1.2.4. *Многоразовые операции*

*1.2.5. Переключение контекста***Заключение****Вариант 3****Лекция***Планирование процессов*

В этой лекции рассматриваются вопросы, связанные с различными уровнями планирования процессов в операционных системах. Описываются основные цели и критерии планирования, а также параметры, на которых оно основывается. Приведены различные алгоритмы планирования.

**1. Уровни планирования****1.1. Критерии планирования и требования к алгоритмам****1.2. Параметры планирования****1.3. Вытесняющее и невытесняющее планирование****1.4. Алгоритмы планирования***1.4.1. First-Come, First-Served (FCFS)**1.4.2. Round Robin (RR)**1.4.3. Shortest-Job-First (SJF)**1.4.4. Гарантированное планирование**1.4.5. Приоритетное планирование**1.4.6. Многоуровневые очереди (Multilevel Queue)**1.4.7. Многоуровневые очереди с обратной связью (Multilevel Feedback Queue)***Заключение****Вариант 4****Лекция***Кооперация процессов и основные аспекты ее логической организации*

Одной из функций операционной системы является обеспечение санкционированного взаимодействия процессов. Лекция посвящена основам логической организации такого взаимодействия. Рассматривается расширение понятия процесс – нить исполнения (thread).

**1. Взаимодействующие процессы****2. Категории средств обмена информацией**

3. *Логическая организация механизма передачи информации*
  - 3.1. Как устанавливается связь?
  - 3.2. Информационная валентность процессов и средств связи
  - 3.3. Особенности передачи информации с помощью линий связи
    - 3.3.1. Буферизация
    - 3.3.2. Поток ввода/вывода и сообщения
  - 3.4. Надежность средств связи
  - 3.5. Как завершается связь?
4. *Нити исполнения*

## Вариант 5

### Лекция

#### *Алгоритмы синхронизации*

Для корректного взаимодействия процессов недостаточно одних организационных усилий операционной системы. Необходимы определенные внутренние изменения в поведении процессов. В настоящей лекции рассматриваются вопросы, связанные с такими изменениями, приводятся программные алгоритмы корректной организации взаимодействия процессов.

1. *Interleaving, race condition и взаимоисключения*
2. *Критическая секция*
3. *Программные алгоритмы организации взаимодействия процессов*
  - 3.1. Требования, предъявляемые к алгоритмам
  - 3.2. Запрет прерываний
  - 3.3. Переменная-замок
  - 3.4. Строгое чередование
  - 3.5. Флаги готовности
  - 3.6. Алгоритм Петерсона
  - 3.7. Алгоритм булочной (Bakery algorithm)
  - 3.8. Аппаратная поддержка взаимоисключений
4. *Команда Test-and-Set (проверить и присвоить 1) Команда Swap (обменять значения)*

## Вариант 6

### Лекция

#### *Механизмы синхронизации*

Для повышения производительности вычислительных систем и облегчения задачи программистов существуют специальные механизмы синхронизации. Описание некоторых из них – семафоров Дейкстры, мониторов Хора, очередей сообщений – приводится в этой лекции.

### **1. Семафоры**

#### **1.1. Концепция семафоров**

#### **1.2. Решение проблемы producer-consumer с помощью семафоров**

### **2. Мониторы**

### **3. Сообщения**

### **4. Эквивалентность семафоров, мониторов и сообщений**

#### **4.1. Реализация мониторов и передачи сообщений с помощью семафоров**

#### **4.2. Реализация семафоров и передачи сообщений с помощью мониторов**

### **5. Реализация семафоров и мониторов с помощью очередей сообщений**

## **Вариант 7**

### **Лекция**

### **Тупики**

В лекции рассматриваются вопросы взаимоблокировок, тупиковых ситуаций и "зависаний" системы.

### **Введение**

#### **1. Условия возникновения тупиков**

#### **2. Основные направления борьбы с тупиками**

#### **3. Игнорирование проблемы тупиков**

#### **4. Способы предотвращения тупиков**

##### **4.1. Способы предотвращения тупиков путем тщательного распределения ресурсов. Алгоритм банкира**

##### **4.2. Предотвращение тупиков за счет нарушения условий возникновения тупиков**

###### *4.2.1. Нарушение условия взаимоисключения*

###### *4.2.2. Нарушение условия ожидания дополнительных ресурсов*

###### *4.2.3. Нарушение принципа отсутствия перераспределения*

###### *4.2.4. Нарушение условия кругового ожидания*

##### **4.3. Обнаружение тупиков**

#### **5. Восстановление после тупиков**

**Вариант 8****Лекция***Организация памяти компьютера*

Простейшие схемы управления памятью В настоящей лекции рассматриваются простейшие способы управления памятью в ОС. Физическая память компьютера имеет иерархическую структуру. Программа представляет собой набор сегментов в логическом адресном пространстве. ОС осуществляет связывание логических и физических адресных пространств.

**1. Физическая организация памяти компьютера****1.1. Локальность****2. Логическая память****3. Связывание адресов****4. Функции системы управления памятью****5. Простейшие схемы управления памятью****5.1. Схема с фиксированными разделами****5.2. Один процесс в памяти****5.3. Оверлейная структура****5.4. Динамическое распределение. Свопинг****5.5. Схема с переменными разделами****6. Страничная память****7. Сегментная и сегментно-страничная организация памяти****Вариант 9****Лекция***Виртуальная память**Архитектурные средства поддержки виртуальной памяти*

Рассмотрены аппаратные особенности поддержки виртуальной памяти. Разбиение адресного пространства процесса на части и динамическая трансляция адреса позволили выполнять процесс даже в отсутствие некоторых его компонентов в оперативной памяти. Следствием такой стратегии является возможность выполнения больших программ, размер которых может превышать размер оперативной памяти.

**1. Понятие виртуальной памяти****2. Архитектурные средства поддержки виртуальной памяти****2.1. Страничная виртуальная память****2.2. Сегментно-страничная организации виртуальной памяти****2.3. Структура таблицы страниц****2.4. Ассоциативная память**

## 2.5. Инвертированная таблица страниц

### 3. Размер страницы

#### Вариант 10

##### Лекция

#### *Аппаратно-независимый уровень управления виртуальной памятью*

Большинство ОС используют сегментно-страничную виртуальную память. Для обеспечения нужной производительности менеджер памяти ОС старается поддерживать в оперативной памяти актуальную информацию, пытаясь угадать, к каким логическим адресам последует обращение в недалеком будущем.

1. *Исключительные ситуации при работе с памятью*
2. *Стратегии управления страничной памятью*
3. *Алгоритмы замещения страниц*
  - 3.1. **Алгоритм FIFO. Выталкивание первой пришедшей страницы**
    - 3.1.1. **Аномалия Белэди (Belady)**
  - 3.2. **Оптимальный алгоритм (OPT)**
  - 3.3. **Выталкивание дольше всего не использовавшейся страницы. Алгоритм LRU**
    - 3.3.1. *Выталкивание редко используемой страницы. Алгоритм NFU*
    - 3.3.2. *Другие алгоритмы*
4. *Управление количеством страниц, выделенных процессу. Модель рабочего множества*
  - 4.1. **Трешинг (Thrashing)**
  - 4.2. **Модель рабочего множества**
5. *Страничные демоны*
6. *Программная поддержка сегментной модели памяти процесса*  
*Отдельные аспекты функционирования менеджера памяти*

#### Вариант 11

##### Лекция

#### *Файлы с точки зрения пользователя*

В настоящей лекции вводится понятие и рассматриваются основные функции и интерфейс файловой системы.

1. *Общие сведения о файлах*

- 1.1. Имена файлов
- 1.2. Типы файлов
- 1.3. Атрибуты файлов
2. *Организация файлов и доступ к ним*
  - 2.1. Последовательный файл
  - 2.2. Файл прямого доступа
  - 2.3. Другие формы организации файлов
3. *Операции над файлами*
4. *Директории. Логическая структура файлового архива*
  - 4.1. Разделы диска. Организация доступа к архиву файлов.
5. *Операции над директориями*
6. *Защита файлов*
  - 6.1. Контроль доступа к файлам Списки прав доступа

## Вариант 12

### Лекция

#### *Реализация файловой системы*

Реализация файловой системы связана с такими вопросами, как поддержка понятия логического блока диска, связывания имени файла и блоков его данных, проблемами разделения файлов и управления дисковым пространством.

1. *Общая структура файловой системы*
2. *Управление внешней памятью*
  - 2.1. **Методы выделения дискового пространства**
    - 2.1.1. *Выделение непрерывной последовательностью блоков*
    - 2.1.2. *Связный список*
    - 2.1.3. *Таблица отображения файлов*
    - 2.1.4. *Индексные узлы*
  - 2.2. **Управление свободным и занятым дисковым пространством**
    - 2.2.1. *Учет при помощи организации битового вектора*
    - 2.2.2. *Учет при помощи организации связного списка*
  - 2.3. **Размер блока**
  - 2.4. **Структура файловой системы на диске**
3. *Реализация директорий*
  - 3.1. **Примеры реализации директорий в некоторых ОС**
    - 3.1.1. *Директории в ОС MS-DOS*
    - 3.1.2. *Директории в ОС Unix*

**3.2. Поиск в директории***3.2.1. Линейный поиск**3.2.2. Хеш-таблица**3.2.3. Другие методы поиска***3.3. Монтирование файловых систем****3.4. Связывание файлов****3.5. Кооперация процессов при работе с файлами***3.5.1. Примеры разрешения коллизий и тупиковых ситуаций***3.6. Надежность файловой системы***3.6.1. Целостность файловой системы***3.7. Порядок выполнения операций****3.8. Журнализация****3.9. Проверка целостности файловой системы при помощи утилит***3.9.1. Управление "плохими" блоками Производительность файловой системы***Вариант 13****Лекция***Система управления вводом-выводом*

В лекции рассматриваются основные физические и логические принципы организации ввода-вывода в вычислительных системах. Физические принципы организации ввода-вывода

**1. Общие сведения об архитектуре компьютера****1.1. Структура контроллера устройства****1.2. Опрос устройств и прерывания. Исключительные ситуации и системные вызовы****1.3. Прямой доступ к памяти (Direct Memory Access – DMA)****1.4. Логические принципы организации ввода-вывода***1.4.1. Структура системы ввода-вывода***1.5. Систематизация внешних устройств и интерфейс между базовой подсистемой ввода-вывода и драйверами****1.6. Функции базовой подсистемы ввода-вывода***1.6.1. Блокирующиеся, неблокирующиеся и асинхронные системные вызовы**1.6.2. Буферизация и кэширование**1.6.3. Spooling и захват устройств**1.6.4. Обработка прерываний и ошибок*

*1.6.5. Планирование запросов***1.7. Алгоритмы планирования запросов к жесткому диску***1.7.1. Структура жесткого диска и параметры планирования**1.7.2. Алгоритм First Come First Served (FCFS)**1.7.3. Алгоритм Short Seek Time First (SSTF) Алгоритмы сканирования (SCAN, C-SCAN, LOOK, C-LOOK)***Вариант 14****Лекция***Сети и сетевые операционные системы*

В лекции рассматриваются особенности взаимодействия процессов, выполняющихся на разных операционных системах, и вытекающие из этих особенностей функции сетевых частей операционных систем.

- 1. Для чего компьютеры объединяют в сети*
- 2. Сетевые и распределенные операционные системы*
- 3. Взаимодействие удаленных процессов как основа работы вычислительных сетей*
- 4. Основные вопросы логической организации передачи информации между удаленными процессами*
- 5. Понятие протокола*
- 6. Многоуровневая модель построения сетевых вычислительных систем*
- 7. Проблемы адресации в сети*
  - 7.1. Одноуровневые адреса*
  - 7.2. Двухуровневые адреса*
  - 7.3. Удаленная адресация и разрешение адресов*
  - 7.4. Локальная адресация. Понятие порта*
  - 7.5. Полные адреса. Понятие сокета (socket)*
- 8. Проблемы маршрутизации в сетях*
- 9. Связь с установлением логического соединения и передача данных с помощью сообщений Синхронизация удаленных процессов*

**Вариант 15****Лекция***Основные понятия информационной безопасности*

Рассмотрены подходы к обеспечению безопасности информационных систем. Ключевые понятия информационной безопасности:

конфиденциальность, целостность и доступность информации, а любое действие, направленное на их нарушение, называется угрозой. Основные понятия информационной безопасности регламентированы в основополагающих документах. Существует несколько базовых технологий безопасности, среди которых можно выделить криптографию.

1. *Угрозы безопасности*
2. *Формализация подхода к обеспечению информационной безопасности*
3. *Криптография как одна из базовых технологий безопасности ОС*
  - 3.1. **Шифрование с использованием алгоритма RSA Теорема Эйлера**

*Заключение*

**Вариант 16**

## **Лекция**

### *Защитные механизмы операционных систем*

Решение вопросов безопасности операционных систем обусловлено их архитектурными особенностями и связано с правильной организацией идентификации и аутентификации, авторизации и аудита.

1. *Идентификация и аутентификация*
  - 1.1. **Пароли, уязвимость паролей**
    - 1.1.1. *Шифрование пароля*
    - 1.1.2. *Авторизация. Разграничение доступа к объектам ОС*
  - 1.2. **Домены безопасности**
  - 1.3. **Матрица доступа**
    - 1.3.1. *Список прав доступа. Access control list*
    - 1.3.2. *Мандаты возможностей. Capability list*
    - 1.3.3. *Другие способы контроля доступа*
    - 1.3.4. *Смена домена*
  - 1.4. **Недопустимость повторного использования объектов**
2. *Выявление вторжений. Аудит системы защиты*
3. *Анализ некоторых популярных ОС с точки зрения их защищенности*
  - 3.1. **MS-DOS**
  - 3.2. **NetWare, IntranetWare**
  - 3.3. **OS/2**
  - 3.4. **Unix Windows NT/2000/XP**

## Лабораторная работа 2

### Создание и редактирование списков и таблиц

**Цель:** научиться создавать и редактировать списки (нумерованные, маркированные, многоуровневые), таблицы. Создание делового письма.

#### Теоретические сведения

##### Стиль **article**

Стиль **article** характеризуется тем, что все страницы являются правосторонними и нечетными, а номер страницы находится в центре основания, если не загружен пакет, устанавливающий другое расположение нумерации страниц.

##### Стиль **book, report**

Стиль **book** и стиль **report** похожи. В команде **\documentclass** по умолчанию принята опция **twoside**. Заголовки нечетных страниц содержат название главы и номер страницы, а четных – название секции и номер страницы. Если присутствует команда **\maketitle** – обычно сразу после начала командных скобок **document** – то создается отдельный титульный лист. Как и в статьях, перед этим должны стоять обе команды **\author** и **\tete** и может стоять команда **\date**; в аргументах этих команд могут встретиться команды **\thanks**. В стиле **book** нельзя использовать командные скобки **abstract**.

Стиль документа **report** (отчет) похож на **article**. Одно из основных отличий – в стиле **report** есть команда рубрикации **\chapter**. По умолчанию принят стиль страниц **empty**, то есть, заголовок страницы пустой, а ее номер находится в центре основания. Если используется стиль страниц **headings**, то заголовок страницы состоит из номера главы и номера страницы – если только не выбрана опция **twoside**: в этом случае заголовки четных страниц состоят из номера главы и номера страницы, а нечетных страниц – из номера секции и номера страницы. Это можно изменить с помощью опции **myheadings** и

связанных с ней команд. Команда `\maketitle`, размещенная в преамбуле, создает отдельный титульный лист. Так же, как и в статьях, перед командой `\maketitle` должны стоять команды `\title` и `\author` и может появиться команда `\date`. Если нужна аннотация, то после команды `\maketitle` должны располагаться командные скобки `abstract`. Аннотация набирается на отдельной странице.

**Списки.** Для создания списков в документе используются окружения `itemize` (для простейших списков), `enumerate` (для нумерованных) и `description` (для списков, в которых каждый пункт имеет заголовок). В любом из этих случаев элементы списка вводятся командой `\item`.

**Простейшие списки (`itemize`).** Каждый элемент перечня вводится командой `\item` без аргумента. При этом:

- каждый элемент списка снабжается черным кружком;
- перечни могут быть вложенными друг в друга;
- максимальная глубина вложенности равна четырем;
- отступы и символы перед элементами выбираются автоматически;
- на втором уровне элементы списка отмечаются **полужирными**

**короткими тире**, на третьем – **звездочками**, на четвертом – **точками**.

**Например:**

```
\begin {itemize}
\item На печати каждый ...
\item Перечни могут быть ...
\begin {itemize}
\item максимальная глубина ...
\item отступы и символы ...
\end {itemize}
\item На втором уровне ...
\end {itemize}
```

**Результат:**

- На печати каждый ...
- Перечни могут быть ...
  - максимальная глубина ...
  - отступы и символы ...
- На втором уровне ...

**Нумерованные списки.** Для нумерованных списков используется окружение `\enumerate`. При этом:

1. В окружении `\enumerate` элементы списка нумеруются цифрами или буквами.
2. Нумерация производится автоматически.
3. Списки могут быть вложенными друг в друга.
  - А) Максимальная глубина вложения равна четырем.
  - Б) Отступы и обозначения для элементов выбираются автоматически.
4. На втором уровне элементы обозначаются строчными буквами, на третьем – римскими цифрами, на четвертом – прописными буквами.

**Например:**

```
\begin {enumerate}
```

```
\item В окружении {\tt enumerate}
```

элементы списка нумеруются цифрами или буквами.

```
\item Нумерация ...
```

```
\item Списки могут быть ...
```

```
\begin {enumerate}
```

```
\item Максимальная глубина ...
```

```
\item Отступы и обозначения ...
```

```
\end {enumerate}
```

**Результат:**

1. В окружении `enumerate` элементы списка нумеруются цифрами или буквами.
2. Нумерация ...
3. Списки могут быть ...
  - (а) Максимальная глубина ...
  - (б) Отступы и обозначения ...

**Верстка таблиц.** Используется окружение `tabular` (для набора таблиц с текстом) и `array` (для набора таблиц из формул).

Окружение `tabular` задает таблицу. Окружению необходимо задать обязательный аргумент – преамбулу таблицы. Преамбула, помещаемая в фигурных скобках непосредственно после `\begin {tabular}`, представляет собой, в простейшем случае, последовательность букв, описывающих структуру колонок таблицы (по букве на колонку). Буквы эти могут быть такими:

**l** – означает колонку, выровненную по левому краю

**r** – означает колонку, выровненную по правому краю

**c** - означает колонку, колонку с центрированным текстом.

Между `\begin {tabular}` (с преамбулой) и закрывающей окружение командой `\end {tabular}` располагается текст таблицы. В нем команда `\\` разделяет строки таблицы, а знак `&`, разделяет колонки таблицы внутри одной строки.

**Например:**

Тип нумерация	перечня	<code>\begin {tabular} {lc}</code>
Itemize	нет	Тип перечня & нумерация <code>\\{5pt}</code>
Enumerate	есть	<code>\tt itemize &amp; нет \\</code>
Description	нет	<code>\tt enumerate &amp; есть \\</code>
		<code>\tt description &amp; нет \\</code>
		<code>\end {tabular}</code>

**Программный код** представлен на Рисунке 3.

```

\documentclass[a4paper, 12pt]{article}
\usepackage{cmap}
\usepackage[T2A]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage[english, russian]{babel}
\usepackage{amsmath}
\usepackage{amssymb}
\usepackage{graphicx}
\textheight=240mm
\textwidth=170mm
\begin{document}
\begin{tabular}{lc}
Тип перечня & нумерация \\
\tt itemize & нет \\
\tt enumerate & есть \\
\| \tt description & нет \\
\end{tabular}
\end{document}

```

Рисунок 3. – Расположение текста таблицы

## Результат

Тип перечня	нумерация
5pt itemize	нет
enumerate	есть
tt description	нет

Рисунок 4. – Результат расположения текста таблицы

`\[st]` вставляет дополнительный вертикальный промежуток. Программа дает возможность сверстать и разлинованную таблицу. Горизонтальные отрезки задаются с помощью команды `\hline`. Эта команда может следовать либо после `\` (тогда отрезок печатается непосредственно после строки, завершенной этим `\`), либо после `\begin{tabular}` и преамбулы (тогда отрезок печатается перед началом таблицы).

Задаваемый командой `\hline` горизонтальный отрезок имеет ширину, равную общей ширине таблицы.

Символ `|`, помещенный в преамбулу таблицы между буквами, описывающими колонки, задает вертикальную линейку, разделяющую эти колонки. Можно поставить символ `|` перед 1-ой из этих букв и после последней – тогда вертикальная линейка будет ограничивать таблицу слева и справа. Несколько таких символов могут стоять подряд – тогда колонки будут

разделяться не одинарной, а двойной, тройной и т.д. вертикальной линией.

**Например:**

<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; padding: 5px;">Слон</td> <td style="width: 50%; padding: 5px;">zilonis</td> </tr> <tr> <td style="padding: 5px;">Бегемот</td> <td style="padding: 5px;">nilzirgs</td> </tr> <tr> <td style="padding: 5px;">Лев</td> <td style="padding: 5px;">lauva</td> </tr> </table>	Слон	zilonis	Бегемот	nilzirgs	Лев	lauva	<pre> \begin {tabular}{    L   L     } \hline Слон &amp; zilonis \\\ Бегемот &amp; nilzirgs \\\ Лев &amp; lauva \\\ \hline \end{tabular} </pre>
Слон	zilonis						
Бегемот	nilzirgs						
Лев	lauva						

Рассмотрим другой пример:

<table style="width: 100%; border-collapse: collapse;"> <tr> <th colspan="2" style="text-align: left; padding: 5px;">Западные сладости</th> </tr> <tr> <th style="width: 30%; padding: 5px;">Название</th> <th style="padding: 5px;">Кол-во Цена</th> </tr> <tr> <td rowspan="2" style="padding: 5px;">Сникерс</td> <td style="padding: 5px;">Штука 330</td> </tr> <tr> <td style="padding: 5px;">Десяток 3000</td> </tr> <tr> <td style="padding: 5px;">Марс</td> <td style="padding: 5px;">Штука 270</td> </tr> <tr> <td style="padding: 5px;">Баунти</td> <td style="padding: 5px;">Штука 350</td> </tr> <tr> <td style="padding: 5px;">Твикс</td> <td style="padding: 5px;">В продаже</td> </tr> <tr> <td style="padding: 5px;">Виспа</td> <td style="padding: 5px;">сегодня нет</td> </tr> </table>	Западные сладости		Название	Кол-во Цена	Сникерс	Штука 330	Десяток 3000	Марс	Штука 270	Баунти	Штука 350	Твикс	В продаже	Виспа	сегодня нет	<pre> \begin {tabular}{  L   Lr   } \hline \multicolumn {3}{  c  }{Западные сладости} \hline Название &amp; Кол-во &amp; Цена \\\ \hline Сникерс &amp; штука &amp; 330 \\\ \cline {2-3} &amp; десяток &amp; 3000 \\\ \hline Марс &amp; штука &amp; 270 \\\ \hline Баунти &amp; штука &amp; 350 \\\ \hline Твикс &amp; В продаже &amp; \\\ \cline {1-1} Виспа &amp; сегодня нет &amp; \\\ \hline \end {tabular} </pre>
Западные сладости																
Название	Кол-во Цена															
Сникерс	Штука 330															
	Десяток 3000															
Марс	Штука 270															
Баунти	Штука 350															
Твикс	В продаже															
Виспа	сегодня нет															

В этом примере использована команда `\multicolumn`. У этой команды три обязательных аргумента:

- 1) Количество колонок, охватываемых нестандартной графой.
- 2) Преамбула нашей графы (буква **L**, **r** или **c** с символами `|`).
- 3) Текст, записываемый в графу.

Для рисования горизонтальных отрезков внутри графы используется команда `\cline`. У нее обязательный аргумент – номера 1-ой и последней из колонок, охватываемых горизонтальной чертой, разделенные знаком «минус».



2. Поместите в преамбулу команду `\usepackage{multicol}`

3. Разместите текст в две колонки (исключая заголовки). Для этого воспользуйтесь окружением `\begin{multicols}{2}` текст `\end{multicols}`

4. Сохраните документ.

5. Измените параметры преамбулы, чтобы текст разместился в три колонки.

## Задание 2. Отформатируйте документ

1. Поместите в преамбулу команду `\usepackage{color}`.

2. Перед использованием цвета, его надо определить командой вида

`\definecolor{название цвета}{rgb}{m,n,p}`,

где числа  $m, n, p \in [0, 1]$  указывают на долю красного, зеленого и голубого в определяемом цвете.

Задайте красный цвет командой `\definecolor{red}{rgb}{1,0,0}`. Ее можно поместить в преамбулу.

Задайте красный цвет для всех названий параграфов командой `\color{red}текст`

3. Протранслируйте файл и просмотрите его.

4. Отцентрируйте заголовки по левому краю. Для этого воспользуйтесь окружением

`\begin{flushleft}` текст `\end{flushleft}` или кнопкой  на панели LaTeX.

5. Задайте курсивное начертание для аннотации. Воспользуйтесь кнопкой  на панели LaTeX

6. Для словосочетания «локально-инерциальные системы отсчета» установите неразрывный пробел (ставится, когда нужно, чтобы два соседних слова не попали на разные строки).

Для этого между словами нужно вставить знак `~` : «локально~инерциальные системы отсчета» (строка не должна начинаться с тире).

7. Создайте два нумерованных списка, используя окружение `enumerate` (предварительно сотрите пустые строки перед абзацами).

Для этого: - откройте окружение `\begin{enumerate}`

- введите элементы перечня: `\item` Первая строка списка

`\item` Вторая строка списка

-закройте окружение `\end{enumerate}`

8. Измените размер и начертание текста по собственному усмотрению, используя соответствующие команды.

9. Измените один из перечней, сделав его маркированным. Используйте окружение `itemize`, элементы перечня вводятся командой `\item`.

### Задание 3. Создание документа со списком и сносками

1. Создайте новый документ стиля **article**

2. Наберите в TeXе текст, который после трансляции должен выглядеть так:

*1. Переходите улицу только на зеленый свет*

*2. Стоящий рядом трамвай обходить можно, а автобус – нет*

*Согласно правилу 2, сформулированному на стр. 132, обходить стоящий автобус нельзя.*

Подсказка для создания ссылки: перед текстом пункта 2, но после команды задания списка, нужно поставить метку `\label{tram}`

Следующее предложение должно выглядеть так:

1.Согласно **правилу~\ref{tram}2**, сформулированному на **стр.~\pageref{tram}132**, обходить стоящий автобус нельзя.

2. Дайте тексту заголовок: Правила дорожного движения

3. Подчеркните заголовок: команда `\underline{текст}`

4. Заключите заголовок в рамку: команда `\fbox{текст}`

5. Протранслируйте файл и просмотрите результат

### Задание 4 Создание делового письма

1. Создайте новый документ стиля **letter**

2. Наберите файл-заготовку:

```
\documentclass[11pt]{letter}
\address{129820, Москва, 1-й Рижский пер., д.2 \
Издательство \glqq Мир \grqq, Сур TUG}
\signature{Исполнительный директор \ И.А.Маховая}
\date{30 августа 2005г.}
\begin{document}
\begin{letter}{Всем, кому нужно набирать \
тексты с формулами}
\opening{Дорогие друзья}
Пользуйтесь системой \TeX.
\closing{С наилучшими пожеланиями,}
\end{letter}
\end{document}
```

3. Протранслируйте файл и просмотрите результат.

### Задания для самостоятельной работы

#### Вариант 1

Покупки в магазине				
№	Продукты	Кол-во	цена	стоимость
1	Хлеб	1	32	32
2	Булка	2	15	30
3	Кефир	1	70	70
4	Сыр	0,2	450	90
5	Шоколад	1	75	75
			<b>Итого</b>	297
			<b>Наличные</b>	500
			<b>Сдача</b>	203

#### Вариант 2

Оплата электроэнергии				
Тариф за кВт/час	5,00 руб.			
Месяц	Предыдущие показания	Текущие показания	Расход кВт/час	Сумма к оплате
Январь	2600	2650	50	250,00
Февраль	2650	2700	50	250,00
Март	2700	2750	50	250,00

Апрель	2750	2800	50	250,00
Май	2650	2700	50	250,00
<b>Всего:</b>			<b>250</b>	<b>1250</b>

### Вариант 3

Наименование	Продажа компьютерной техники			Всего
	Количество, шт			
	Магазин 1	Магазин 2	Магазин 3	
Компьютер	2	5	7	14
Принтер	4	0	6	10
Сканер	3	4	8	15
Планшет	1	5	3	9
Ноутбук	3	7	5	15
Смартфон	4	5	8	17
Монитор	1	0	4	5
<b>Итого:</b>	<b>18</b>	<b>26</b>	<b>41</b>	<b>85</b>

### Вариант 4

Водорастворимые и жирорастворимые витамины					
Витамины					
Водорастворимые			Жирорастворимые		
Название	Сут. доза (мг)	Продукты	Название	Сут. доза (мг)	Продукты
Витамин С	75-100	лимон, молоко, смородина	Витамин А	15	Сливочное масло, томаты, желток
Витамин В1	2-3	рожь, пшено, рис	Витамин D	0,025	Рыбий жир
Витамин В2	1-2	молоко, яичный белок	Витамин Е	10 - 15	Подсолнечное масло
Витамин В12	0,001	печень, рыба	Витамин К	1 - 2	Ягоды рябины

### Вариант 5

Сведения об успеваемости студентов								
Учебная дисциплина	группа	Средний балл	Всего сдано	отлично	хорошо	удовл.	неудовл.	неявка
физика	АВП-1	3	23	3	10	6	4	0
химия	АГД-1	4,2	26	6	11	6	3	0
информатика	АД-2	4,5	28	8	10	5	5	0
геометрия	АДП-2	4,7	24	18	4	2	0	1

**Вариант 6**

<b>Посещаемость и выручка кинотеатров города</b>						
<b>Название кинотеатра</b>	<b>январь</b>		<b>февраль</b>		<b>посещаемость</b>	<b>выручка</b>
	<b>посещаемость</b>	<b>выручка</b>	<b>посещаемость</b>	<b>выручка</b>		
	<b>(т.чел)</b>	<b>(т.руб)</b>	<b>(т.чел)</b>	<b>(т.руб)</b>		
Родина	3,99	960	3,09	990	7,08	1950
Заря	9,86	145	11,67	185	21,53	330
Победа	3,15	2110	3,11	2130	6,26	4240
Октябрь	1,11	1530	1,03	1540	2,14	3070
<b>Итого:</b>	<b>18,11</b>	<b>4745</b>	<b>18,9</b>	<b>4845</b>	<b>37,01</b>	<b>9590</b>

**Вариант 7**

<b>Предметы</b>	<b>Успеваемость</b>				
	<b>По полугодиям</b>		<b>годовая</b>	<b>экзамен</b>	<b>итоговая</b>
Русский язык	5	5	5	5	5
Математика	4	4	4	4	4
Физика	3	3	3	3	3
Химия	4,5	4,5	4,5	4,5	4,5
Геометрия	5	5	5	5	5
Информатика	3,5	3,5	3,5	3,5	3,5

**Вариант 8**

<b>Магазин Детская одежда</b>					
<b>Наименование</b>	<b>Объем продаж по месяцам</b>			<b>Всего</b>	<b>Максимум</b>
	<b>январь</b>	<b>Февраль</b>	<b>Март</b>		
Брюки	12	22	10	44	22
Жакет	5	10	12	27	12
Пиджак	22	10	5	37	22
Куртка	7	13	21	41	21
Пальто	15	8	4	27	15
Жакет	7	11	21	39	21
Блузка	15	10	20	45	20
<b>Итого:</b>	<b>83</b>	<b>84</b>	<b>93</b>	<b>260</b>	

**Вариант 9**

<b>№</b>	<b>ФИО студента</b>	<b>Предметы</b>			<b>Средний балл</b>
		<b>Физика</b>	<b>Химия</b>	<b>Математика</b>	
1	Иванов И.И.	4	4	5	4,3
2	Петров П.П.	5	5	4	4,7
3	Сидоров И.К.	3	4	5	4
4	Мальшева И.В.	5	3	5	4,3
5	Захарова Н.Т.	5	5	5	5
	<b>Средний балл</b>	<b>4</b>	<b>4,2</b>	<b>4,8</b>	

**Вариант 10**

Наименование счета	Код счета	На начало месяца		Обороты за месяц	
		Дебет	Кредит		
Основные средства	01	1200		616	300
Амортизация основных средств	05		540	120	261
Нематериальные активы	06	400	50		
Амортизация нематериальных активов	08		160	15	
Вложения во внеоборотные активы	09	180	267	276	

**Вариант 11**

Товар	Курс доллара	Стоимость товара	
		в долларах	в рублях
Лазерный принтер	62	380	442
Компьютер	62	950	1012
Мышь	62	35	97
Защитный экран	62	5	67
Клавиатура	62	100	162
Коврик для мыши	62	0,5	62,5
Жесткий диск	62	25	87

**Вариант 12**

№	Наименование	Цена	Количество	Общая стоимость
1	Компьютер	700	4	2800
2	Ксерокс	200	1	200
3	Кондиционер	300	3	900
4	Картридж	20	6	120
5	Бумага	4	10	40
	<b>Всего</b>			<b>4060</b>

**Вариант 13**

№	Наименование товара	Ед. измерения	Количество	Цена	Стоимость
1	Яблоки	кг	5	80	400
2	Виноград	кг	6	220	1100
3	Бананы	кг	15	150	2250
4	Груша	кг	30	80	2400
5	Слива	кг	6	170	1020
	<b>Итого</b>				<b>7170</b>

**Вариант 14**

Наименование товара	по месяцам				
	январь	февраль	март	апрель	май
Трубы толстые	23	23	34	67	34
Трубы тонкие	23	78	56	34	56
Панели квадратные	1	3	5	2	4

**Вариант 15**

№	Наименование товара	Количество		Остаток	Макс. продано товара	Мин. на складе
		На складе	Продано товара			
1	Монитор LG	23	11	12	13	12
2	Монитор Samsung	15	2	13		
3	Системный блок	38	13	25		
4	Манипулятор мышки	40	13	27		
	Всего:	116	39			

**Вариант 16**

Вид учебной работы	Количество часов		
	Всего по учебному плану	в том числе по семестрам	
Аудиторные занятия:	100	56	44
лекции	40	20	20
Практические занятия	-	-	-
Лабораторные работы	60	36	24
Самостоятельная работа	100	56	44
Всего часов на дисциплину	200	112	88

**Вариант 17**

Название страны	Население млн. чел.		Плотность чел. на кв. км.		Площадь млн. кв. км.
	2010	2020	2010	2020	
Австралия	19	26	2	3	8,5
Африка	361	628	12	21	30,3
Европа	642	701	61	67	10,5
Южная Америка	190	291	11	16	17,8
Северная Америка	320	422	13	17	24,3
Азия	2161	3133	49	71	44,4
Всего:	3693	5201	27	38	135,8

**Вариант 18**

№	Наименование товара	Ед. измерения	Кол-во	Цена	вес	Стоимость
	Телевизор пылесос	шт.	112	18125,5	24	
	Холодильник	шт.	238	2120,3	8	
	Утюг	шт.	177	15674,5	50	
	Миксер	шт.	1174	627,25	1	
	Фен	шт.	139	11345,6	4	
	Чайник	шт.	98	627,5	0,5	
	Итого:					

**Вариант 19**

№	Продукт	Цена	Поставлено	Продано	Осталось	Выручка
1	Молоко	20	100	100	0	2000
2	Сметана	10,2	85	70	15	714
3	Творог	18,5	125	110	15	2035
4	Сливки	5,4	250	225	25	1215
5	Йогурт	15,2	50	45	5	684

**Вариант 20**

Наименование товара	Цена 1 шт.	Кол-во	Общая стоимость
Товар 1	45,65	10	55,65
Товар 2	12,3	60	72,3
Товар 3	480,1	12	492,1
Товар 4	47,1	30	77,1
Товар 5	50,65	4	54,65

## Лабораторная работа 3

### Форматирование текстового документа LaTeX

**Цель:** научиться форматировать текст, в том числе выравнивать текст, создавать отступы, сноски, перекрестные ссылки, библиографические списки.

#### Теоретические сведения

##### Разметка документа

Выравнивание текста:

1. По центру текст можно установить командой **center**, указанной в фигурных скобках

```
\begin{center}
```

Выравнивание по центру

```
\end{center}
```

Можно так же в области декларации указать **\centering** для центрирования текста.

Для выравнивания текста по правому краю можно использовать команду **flushright** или в области декларации указать **\raggedright**.

```
\begin{flushright}
```

Выравнивание по правому краю

```
\end{flushright}
```

Для выравнивания текста по левому краю можно использовать команду **flushleft** или в области декларации указать **\raggedleft**.

```
\begin{flushleft}
```

Выравнивание по левому краю

```
\end{flushleft}
```

##### Отступы

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

Вертикальные и горизонтальные отступы можно выставлять следующим образом:

`\vspace{5mm}` – мягкий вертикальный отступ (в некоторых случаях игнорируется, например, если попадает на начало страницы)

`\vspace*{10mm}` – вертикальный отступ

`\hspace{20mm}` – мягкий горизонтальный отступ (в некоторых случаях игнорируется, например, если попадает на конец строки).

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`).

Переходы внутри абзаца можно сделать так:

`\newline` – переход на новую строку внутри одного абзаца, предыдущая строка не растягивается.

`\linebreak` – переход на новую строку внутри абзаца, предыдущая строка растягивается по ширине.

`\pagebreak` – переход на новую страницу, предыдущая растягивается

`\newpage` – переход на новую страницу

### Сноски

Сноска – это примечание, которое содержит дополнения или пояснения к тексту. В ней могут быть указаны библиографические сведения (в том числе и ссылка), примечания автора, дополнительная информация.

Сноску в тексте можно вставить по команде `\footnote[number]{text}` где `text` — текст сноски, а положительное число `number` – её номер. Маркер сноски печатается на месте команды, а текст – внизу страницы. Если опция `number` опущена, то сноске присваивается порядковый номер из счётчика `footnote`.

Сноски в документе нумеруются подряд на протяжении всей главы или всего документа (в классе `article`).

Пример: вставим в следующий фрагмент текста документа

*... в тени березы. Хорошо, что ...*

сноску:

*... в тени березы\footnote{Береза – это дерево}. Хорошо, что ...*

Внизу страницы появится сноска «Береза – это дерево», а справа у слова «береза» порядковый номер сноски.

Команду `\footnote` можно использовать только для текста или в **minipage**.

Если текст сноски состоит из нескольких абзацев, то нужно использовать пустую строку между ними. Нумерация сноски происходит автоматически в порядке возрастания. Если есть необходимость изменить номер сноски, то он должен быть указан как необязательный параметр команды, т.е. в квадратных скобках.

### Пример:

Трудности, стоящие на пути к достижению цели\footnote[25]{Ильин Е.П. Психология воли. СПб.: Питер, 2009, 368 с.}

Можно не указывать номер, как необязательный параметр, тогда нумерация будет автоматической.

Даже если использовалась команда `\footnote` с необязательным параметром, автоматическая нумерация сносок не будет сбиваться.

Например, предыдущая сноска была 3, затем искусственно создана сноска с номером 25, а следующая сноска будет иметь номер 4.

Автоматическая нумерация заново на каждой странице в LaTeX не предусмотрена.

### Перекрестные ссылки

В книгах, отчётах и статьях часто встречаются ссылки на отдельные элементы документа. Команда `\label{name}` помечает объект, после которого установлена команда, меткой **name** (метка должна быть уникальной). Например, `\section{Глава 1}\label{B_Section1}` – пометит раздел в документе.

Команда `\ref{name}` печатает порядковый номер объекта, помеченного меткой **name**. Для каждого класса объектов ведётся свой счетчик объектов.

Объектом может быть, например: рисунок, таблица, раздел, подраздел и т.д.. Объекты нумеруются в порядке их появления в документе. Например, `\ref{B_Section1}` напечатает порядковый номер раздела "Глава 1".

Для ссылки на источник литературы используется команда `\cite[]{\name}` – печатает в квадратных скобках номер или метку элемента с идентификатором `name`.

Например, источник литературы задан следующим образом:  
`\bibitem{B_Gergel}` Гергель В.П., Стронгин Р.Г. Основы параллельных вычислений для многопроцессорных вычислительных систем. - Н.Новгород, ННГУ, 2001

Команда `\cite{B_Gergel}` заставит LaTeX вставить в основной текст ссылку на данный источник в формате: [**Порядковый номер источника в списке литературы**].

### Список литературы

Список литературы задается окружением:

```
\begin{thebibliography}{99}
```

```
\setlength{\itemsep}{-2pt}
```

...

```
\end{thebibliography}
```

Список литературы оформляется как окружение **thebilibiography**. Оно имеет обязательный аргумент – номер элемента библиографии, который займет больше всего места в печати.

Команда `\setlength{\itemsep}{-2pt}` устанавливает расстояние между источниками.

Для автоматической нумерации в тексте ссылок на литературные источники, необходимо перед каждым источником в списке поставить метку `\bibitem{метка}`, а в тексте обращаться к источнику командой `\cite{метка}`. После компиляции на месте такой команды появится в квадратных скобках порядковый номер источника в списке литературы.

Каждый источник вводится командой `\bibitem`. У нее обязательный аргумент – ваше условное обозначение (любая последовательность букв или цифр).

В тексте ссылка на источник делается с помощью команды `cite`. Ее обязательный аргумент – условное обозначение того источника, на который вы ссылаетесь.

**Например:** ссылки на список литературы (например, [1] или [2, с. 10])

`\cite{метка книги}` или

`\cite[с.~10]{метка книги}`

Рассмотрим другой пример создания библиографического списка.

**Например:**

<p>В книге [3,Глава 1] описана встреча Вини-Пуха с несколькими пчелами.</p> <p>В [2,1] приведены другие сведения о медведях.</p> <p>[1] М.Е. Салтыков-Щедрин Медведь на воеводстве</p> <p>[2] Л.Н. Толстой Три медведя</p> <p>[3] А.А. Милн Вини-Пух</p>	<p>В книге~ <code>\cite</code> [Глава~1] <code>{Winnie}</code> описана встреча Вини-Пуха с несколькими пчелами. В~ <code>\cite {med3,voevoda}</code> приведены другие сведения о медведях.</p> <p><code>\begin{thebibliography}{99}</code></p> <p><code>\bibitem {voevoda}</code> М.~Е.~ Салтыков-Щедрин Медведь на воеводстве</p> <p><code>\bibitem {med3}</code> Л.~Н.~ Толстой Три медведя</p> <p><code>\bibitem {Winnie}</code> А.~А.~ Милн Вини-Пух</p> <p><code>\end {thebibliography}</code></p>
--	---

Пример документа для создания библиографического списка показан на рисунке 7.

```

\documentclass[a4paper, 12pt]{article}
\usepackage{cmap}
\usepackage[T2A]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage[english, russian]{babel}
\usepackage{amsmath}
\usepackage{amssymb}
\usepackage{graphicx}
\textheight=240mm
\textwidth=170mm
\begin{document}
В книге~ \cite [Глава~1] {Winnie} описана
встреча Вини-Пуха с несколькими пчелами. В~
\cite {med3,voevoda} приведены другие сведения
о медведях.
\begin{thebibliography}{99}
  \bibitem {voevoda}
  М.~Е.~ Салтыков-Щедрин
  Медведь на воеводстве
  \bibitem {med3} Л.~Н.~ Толстой Три медведя
  \bibitem {Winnie}
  А.~А.~ Милн Вини-Пух
\end{thebibliography}
\end{document}

```

Рисунок 7. – Текстовый документ, содержащий список источников.

В результате будет создан текстовый документ, представленный на рисунке 8.

В книге [3, Глава 1] описана встреча Вини-Пуха с несколькими пчелами. В [2, 1] приведены другие сведения о медведях.

### Список литературы

- [1] М. Е. Салтыков-Щедрин Медведь на воеводстве
- [2] Л. Н. Толстой Три медведя
- [3] А. А. Милн Вини-Пух

Рисунок 8. – Результат компиляции документа со списком литературы

Рассмотрим другой пример создания списка литературы.

Следующая команда позволяет изменить название библиографического списка: `\renewcommand{\refname}{\Large Список использованных источников}`

**Пример** оформления списка:

```
\begin{thebibliography}{7}
```

```
\bibitem{Najmark} Наймарк~М.~А. Линейные дифференциальные операторы. М. : Наука, 1969.
```

```
\bibitem{Belman} Белман~Р. Дифференциально-разностные уравнения. М. : Мир, 1967.
```

```
...
```

```
\end{thebibliography}
```

В результате компиляции будет получен список источников, представленный на рисунке 9.

#### Список использованных источников

1. Наймарк М. А. Линейные дифференциальные операторы. М. : Наука, 1969.
2. Белман Р. Дифференциально-разностные уравнения. М. : Мир, 1967.
3. Корнев В. В., Хромов А. П. О равносходимости разложений по собственным и присоединенным функциям интегральных операторов с ядрами, допускающими разрывы производных на диагоналях // Матем. сб. 2001. Т. 192. № 10. С. 33–50

Рисунок 9. – Результат компиляции списка источников

#### Задания для самостоятельной работы

Создать LaTeX-документ. Для этого нужно взять любую статью подобную приведенному ниже примеру. Документ должен иметь следующую структуру: название, автор, аннотация, введение, основная часть, состоящая из двух – трех параграфов, заключение и литература. Текст должен содержать сноски, ссылки, перекрестные ссылки.

1. Отформатировать основной текст следующим образом:

- абзацный отступ – 1,25 см;
- интервал между абзацами – брт.
- выравнивание абзаца – по левому краю.

2. Оформление заголовка раздела:

- выравнивание заголовка раздела по левому краю.

- отступ сверху и снизу заголовка раздела – бпт.
- размер шрифта заголовка – 12pt.
- 3. Оформить ссылки по тексту на источники литературы.
- 4. Стил ь заголовка сделать жирным, по центру, 14 pt.
- 5. Стил ь шрифта текста цитаты – полужирный.
- 6. Стил ь шрифта фамилии автора выполнить курсивом.

### **Пример текстового документа**

## **Подготовка специалистов в области высокопроизводительных вычислений на базе межуниверситетской инновационной учебно-исследовательской лаборатории InterUniLab\***

А.С. Абрамова, Н.А. Шехунова, А.В. Бухановский

#### **Аннотация**

Рассматриваются особенности разработки учебно-методического комплекса «Высокопроизводительные вычисления» на основе модульного и компетентностного подходов, ориентированных на слушателей-магистров по специализации «Разработка программного обеспечения» Санкт-Петербургского Государственного университета информационных технологий, механики и оптики. Самостоятельная работа в рамках комплекса ориентирована на участие в учебно-исследовательских проектах, выполняемых в межуниверситетской учебно-исследовательской лаборатории InterUniLab.

### **Введение**

Современный этап развития высокопроизводительных вычислительных технологий характеризуется: широким распространением многоядерных компьютерных архитектур, удешевлением и доступностью кластерных систем на основе стандартных комплектующих, развитием технологий распределенных вычислений, в том числе, Грид [1]. Это требует модификации и развития соответствующих учебных комплексов. Недостаточное внимание сейчас уделяется системному подходу к параллельному математическому и программному обеспечению, как совокупности математических моделей, методов их реализации, параллельных алгоритмов, технологий программирования, тестирования и верификации параллельных программ, хотя именно такой путь позволяет строить эффективные параллельные алгоритмы и проектировать надежные программные системы на их основе [2].

Так как Россия в 2003 году присоединилась к Болонскому процессу, новые УМК должны разрабатываться в соответствии с требованиями, предъявляемыми Европейским союзом. Это позволит включить российские учебные курсы в европейскую систему образования. Главной проблемой перехода к новой системе образования, вызывающей полемику, является переход от квалификационного подхода к компетентностному, а также модульная структура обучения.

---

\* Работа выполняется при финансовой поддержке Министерства образования и науки РФ в рамках приоритетного национального проекта «Образование».

Преподавание высокопроизводительных вычислений (High Performance Computing, HPC), как дисциплины из области компьютерных наук, требует серьезной материальной и информационной базы. HPC быстро развивается, что приводит к необходимости постоянного обновления учебных материалов, которые должны в общем случае содержать мультидисциплинарные сведения (архитектура ЭВМ, теория построения алгоритмов, технологии программирования, коммуникационные технологии и пр.). Требования к педагогическому процессу в области высокопроизводительных вычислений, такие как направленность на конкретный результат образования и гибкость, дают право описать его в виде функциональной системы. Описание педагогического процесса в виде функциональной системы дает возможность эффективно управлять этим сложным процессом [3].

## 1. Разработка курса

Курс «Конструирование и анализ параллельных алгоритмов» разбит на модули: проектирование параллельных алгоритмов, проектирование параллельных программ, прикладные параллельные алгоритмы. По окончании курса студент должен уметь: строить эффективные параллельные алгоритмы, применительно к конкретной вычислительной архитектуре, уметь оценивать и моделировать параллельную производительность алгоритма для определенной вычислительной архитектуры, выбирать алгоритмы для решения поставленной задачи, выбирать технологии параллельного программирования для решения поставленной задачи, реализовывать ПО с помощью технологий параллельного программирования, оценивать эффективность работы параллельной программы и формулировать рекомендации по ее модификации.

Курс «Технологии распределенных вычислений и систем» разбит на 5 модулей: основные виды распределенных вычислительных архитектур, концепция и модели Грид, проектирование приложений для распределенных вычислительных архитектур, разработка приложений в peer-to-peer-системах, разработка приложений в современных Грид-системах. По окончании курса студенты должны уметь выбирать оптимальный способ организации распределенной вычислительной системы, классифицировать распределенные вычислительные архитектуры, оценивать и моделировать параллельную производительность распределенной вычислительной системы, выбирать вычислительную систему для решения поставленной задачи, выбирать программный инструментальный для решения поставленной задачи, пользоваться программным инструментарием распределенных вычислительных систем, разрабатывать эффективное программное обеспечение для распределенных вычислительных систем.

В методическое обеспечение курса входит виртуальная лаборатория, которая формируется на базе разрабатываемых лабораторных работ. Лабораторные работы посвящены построению и оптимизации параллельных алгоритмов: метод Монте-Карло вычисления интегралов, решение систем линейных алгебраических уравнений методом Монте-Карло, генетический алгоритм (глобальная оптимизация), численное интегрирование (квадратуры), поиск на графах, умножение матриц, метод конечных элементов, параллельное LU-разложение.

## 2. Работа студентов

Самостоятельная работа студентов в рамках данных курсов ориентирована на участие в учебно-исследовательских проектах, выполняемых в межуниверситетской инновационной учебно-исследовательской лаборатории InterUniLab. InterUniLab создана совместной инициативой Санкт-Петербургских университетов — Санкт-Петербургского Государственного политехнического университета, СПбГУ ИТМО, Санкт-Петербургского Государственного университета авиаприборостроения и др. — и Фондом содействия развитию малых предприятий в научно-технической сфере при поддержке глобальных IT-компаний, таких как Intel, Microsoft, Cadence. Одной из ее задач является подготовка квалифицированных кадров в области критических технологий (в том числе, технологии распределенных вычислений и систем, высокопроизводительные вычисления) путем вовлечения слушателей в практическую реализацию мотивационных (курсовых) проектов — индивидуального или в составе рабочей группы. Мотивационный проект ориентирован на разработку математического обеспечения высокопроизводительных вычислений в определенной предметной области. Слушателям на выбор будут предложены задачи из области гидрометеорологии, экологии, биомедицины, физики плазмы,

технической диагностики и управления подвижными техническими объектами, основанные на реальных массивах данных.

УМК «Высокопроизводительные вычисления» в СПбГУ ИТМО в настоящий момент находится в состоянии разработки. Однако отдельные его элементы уже прошли апробацию в рамках летних и зимних школ Intel (2006, 2007 гг.), а также в плановом учебном процессе СПбГУ ИТМО. Ввод УМК в опытную эксплуатацию планируется в осеннем семестре 2008 г.

## **Литература**

1. Defining the Grid — a snapshot of the current view // Н. Stockinger, 2006
2. Гергель В.П., Стронгин Р.Г. Основы параллельных вычислений для многопроцессорных вычислительных систем. – Н.Новгород, ННГУ, 2001
3. Анохин П.К. Принципиальные вопросы общей теории функциональных систем. — М, 1973

## Лабораторная работа 4

### Создание математических формул

**Цель:** научиться создавать и редактировать естественно-научные тексты, содержащие математические символы и формулы

#### Теоретические сведения

Для работы с формулами в текстовом документе LaTeX необходимо подключить пакеты **amsmath** и **amssymb** перед пакетом **russian**. Добавить пакеты: **amsfonts**, **amsthm**, **mathrsfs**.

Все математические символы и формулы (математические моды) должны быть ограничены одинарными или парными знаками долларов \$. Кроме \$ и бэкслэш \, управляющими символами являются символы {, }, показывающие область действия команд или ограничивающие аргументы команд.; % - начало комментария в строке. Для набора верхнего индекса применяется знак ^, для нижнего индекса - \_. Если потребуется написать управляющие символы, а также символы #, &, как обычные символы, то необходимо перед символом поставить бэкслэш \. Для написания символов ^, ~ как обычных символов применяются команды  $\wedge$ ,  $\sim$ .

Знак ~, расположенный между двумя словами не позволяет разрывать их при переносе.

Если формула или буква окружена одинарными знаками долларов слева и справа, то LaTeX помещает такую формулу как обычный текст в строке. Для получения формулы, расположенной между строк (выключенный стиль), ее надо окружить двойными знаками долларов. Пробелы в формулах LaTeX расставляет сам и игнорирует набранные пробелы.

Все символы, используемые LATEX'ом в математических формулах можно посмотреть в учебнике Львовского С.М. «Набор и верстка в системе LaTeX».

## Нумерация формул

Естественно-научные текстовые документы как правило формулы нумеруются. При этом нумерацию формул можно организовать двумя способами. Во первых автоматически. Но так можно нумеровать только выключные формулы. Для этого нужно выключенную формулу оформить окружением **equation** и при этом знаков \$\$ быть не должно. Каждая такая формула на печати автоматически получит номер. Чтобы на него можно было ссылаться, надо формулу пометить: в любом месте между `\begin{equation}` и `\end{equation}` поставить команду `\label`, и после этого команда `\ref` будет генерировать номер формулы.

Для этого возможно понадобится повторный запуск LATEX.

Пример нумерации формул показаны на рисунке 10.

<p>Как известно,</p> $7 \times 9 = 63. \quad (1)$ <p>.....</p> <p>Из формулы (1) следует, что  <math>63/9 = 7.</math></p>	<p>Как известно,</p> <pre>\begin{equation} \label{trivial} 7\times9=63. \end{equation} ..... Из формулы~(\ref{trivial}) следует, что \$63/9=7\$.</pre>
---	--

Рисунок 10. – Возможности нумерации формул

Второй способ связан с тем, что можно не использовать автоматическую генерацию номеров формул, а ставить их в ручную с помощью команды `\eqno`.

Пример использования автоматической генерации номеров формул показан на рисунке 11.

<p>Простое тождество</p> $7 \times 9 = 63 \quad (3.2)$ <p>известно каждому школьнику.</p>	<p>Простое тождество</p> <pre>\$\$ 7\times9=63\eqno (3.2) \$\$ известно каждому школьнику.</pre>
---	--

Рисунок 11. – Автоматическая генерация номеров формул

### Смена шрифтов в формуле

По умолчанию все латинские буквы в формулах набираются курсивом. Для смены шрифта можно воспользоваться командами `\bfseries` или `\itshape`. Но можно сделать это иначе. Пусть, например, вам нужна буква  $P$ , набранная прямым жирным шрифтом (Рисунок 12).

$$P^n \qquad \qquad \qquad \$\mathbf{P}^n\$$$

Рисунок 12. – Смена шрифта

### Включение текста в формулы

В математическую формулу можно включить фрагмент текста. Для этого можно использовать команду `\mbox`. В следующем примере используется команда `\qquad`, делающая в тексте или формуле пробел размером `2em` (Рисунок 13).

Пример:

$$\sqrt{x^3} = x \quad \text{для всех } x.$$

```

$$
\sqrt{x^3}=x\qquad
\mbox{для всех }x.
$$

```

Рисунок 13. – Включение текста в формулы

### Скобки переменного размера

Если заключенный в скобки фрагмент формулы занимает много места по вертикали (за счет дробей, степеней и тому подобного), то и сами скобки должны быть большего размера, чем обычные. Для этого можно использовать механизм автоматического выбора размера скобок.

Например:

$$e = \lim_{n \rightarrow \infty} \left( 1 + \frac{1}{n} \right)^n$$

При наборе такой формулы надо поставить команду `\left` перед открывающей скобкой и команду `\right` перед закрывающей.

```


$$e = \lim_{n \rightarrow \infty} \left(1 + \frac{1}{n}\right)^n$$


```

### Практические задания

**Задание 1.** Набрать в LaTeX простейшие математические формулы.

- Создайте документ в любом стиле (например, **article**)
- Введите следующие формулы (они будут расположены внутри строки):
  - Сложение, вычитание  $a+b$   $a-b$
  - Степень, индексы  $a_{b+c}^{de}$
  - Деление  $\frac{b+c}{de}$
  - или  $b+c \over de$
  - Корень квадратный  $\sqrt{a}$
  - Корень n-ой степени  $\sqrt[n]{a}$
- Введите следующую букву и формулу, которые будут расположены между строк
  - $A,$   
 $a_{b+c}^{de}$
  - $a_{b+c}^{de}$
- Замените двойные знаки долларов (для написания формулы в выключенном стиле) на бакслэши с квадратными скобками.
  - $[a_{b+c}^{de}]$

В математической моде пробелы игнорируются. Их нужно устанавливать самостоятельно командами – бэкслэш с пробелом  $\$ , бэкслэш с запятой  $\,$  или бэкслэш с точкой с запятой  $\;$ . Текст в математической моде отмечается командами  $\text{}$ ,  $\textbf{}$ ,  $\textit{}$  и так далее (соответствующие кнопки

расположены в нижней строке панели меню в закладке “LaTeX **a a a A a a a ?**”).

Отличие в написании выключенной формулы с помощью двойных знаков доллара и с бэкслэш с квадратными скобками состоит в следующем: если в команде `\documentclass` установлена опция `fleqn`:

```
\documentclass[12pt, fleqn]{article},
```

то выключенные формулы, полученные парой команд `\[` и `\]`, не будут центрированы на странице – они будут отстоять от левого поля на расстояние, содержащееся в параметре длины `\mathindent`. Задать такой отступ можно в любом месте; например, после команды `\mathindent=1.2in` все формулы, обрамленные квадратными скобками, будут отстоять от левого края на `1.2in`. При этом выключенные формулы, полученные с использованием двойных знаков доллара, по-прежнему будут центрироваться.

### Задания для самостоятельной работы

<p><b>Вариант 1</b></p> $c = \frac{Q}{m(t_2 - t_1)}$	<p><b>Вариант 6</b></p> $(a - b)^3 = a^3 - 3a^2b + 3ab^2 - b^3$
<p><b>Вариант 2</b></p> $x_1 = \frac{-p - \sqrt{D}}{2n}$	<p><b>Вариант 7</b></p> $\sqrt{\frac{a^2 + 6ab + 25b^2}{a - 2\sqrt{ab} + 5b}} = \sqrt{a} + \sqrt{b}$
<p><b>Вариант 3</b></p> $\frac{a^2 + bc}{2} \geq \sqrt{a^2bc}$	<p><b>Вариант 8</b></p> $y = \sqrt{x^2 + 2\sqrt{2x} + 2} + \sqrt{x^2 - 2\sqrt{2x} + 2}$
<p><b>Вариант 4</b></p> $a^2 + b^2 = c^2 \quad (\alpha + \beta)^2 = 1$	<p><b>Вариант 9</b></p> $(a - b)^4 = a^4 - 4a^3b + 6a^2b^2 - 4ab^3 + b^4$
<p><b>Вариант 5</b></p> $x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$	<p><b>Вариант 10</b></p>

	$X = \frac{\left(p^2 - \frac{1}{q^2}\right)^2 \left(p - \frac{1}{q}\right)^{q-p}}{\left(q^2 - \frac{1}{p^2}\right)^2 \left(q - \frac{1}{p}\right)^{p-q}}$
--	---

## Задание 2. Научиться изменять размеры символов в математических формулах

1. Создайте новый документ в стиле **article**
2. Введите математическую формулу командами

$$b^2 + \frac{a^1}{b_1 + \frac{a^2}{b_2 + \frac{a^3}{b_3}}}$$

`$$b^2+\frac{a^1}{b_1+\frac{a^2}{b_2+\frac{a^3}{b_3}}}`

Обратите внимание, что буквы в дробях, верхних и нижних индексах получились меньших размеров. В этой формуле первое слагаемое написано в стиле `\displaystyle` (стиль для выключенных формул), числитель дроби в знаменателе – в стиле `\scriptstyle` (стиль верхних и нижних индексов), а маленькие индексы – в стиле `\scriptscriptstyle` (стиль индексов к индексам). Формулы в тексте пишутся в стиле `\textstyle`.

3. Замените в формуле все стили одним – для выключенных формул. Для упрощения записи переименуйте команду `\displaystyle` в команду `\D`. Для этого:
  - Наберите в строке команду `\newcommand{\D}{\displaystyle}`
  - Ввести команду

`$$b^2+\frac{a^1}{\D b_1+\frac{a^2}{\D b_2+\frac{a^3}{b_3}}}`
4. Протранслируйте программу и посмотрите получившийся результат.
5. Создайте новый документ, введите в нем формулы в двух видах: с уменьшенными размерами индексов, с одинаковыми размерами индексов:

$$\frac{1}{a_1 + \frac{1}{a_3 + \frac{1}{a_5 + \frac{1}{a_7}}}}$$

## Уравнения и системы уравнений

Окружение  $\begin{cases} . . \end{cases}$  устанавливает знак системы.

Такое окружение применяется при определении функций.

**Задание 3.** Ввести функцию  $\varphi(x) = \begin{cases} 0, & \text{если } x \leq 0 \\ e^{-1/x}, & \text{если } x > 0 \end{cases}$

1. Создайте новый документ в стиле **article**.
2. Подключите пакеты **amsmath** и **amssymb** перед пакетом **russian**
3. Введите команду  $\varphi(x)=\begin{cases}$

$0, & \text{\text{если}} \ x \leq 0, \ \backslash$

$e^{-1/x}, & \text{\text{если}} \ x > 0$

$\end{cases}$

4. Самостоятельно введите команды для задания следующих функций (для ввода знака  $\geq$  используйте команду **\geq**):

$$\text{а) } y = \begin{cases} 0, & \text{если } x \leq -1 \text{ или } x \geq 1 \\ \sqrt{1-x^2}, & -1 < x < 1 \end{cases} \quad \text{б) } y = \begin{cases} \sin(x), & \text{если } x \leq a \\ x^2, & \text{если } a < x \leq b \\ \sqrt{x}, & \text{если } x > b \end{cases}$$

$$\text{в) } y = \begin{cases} e^{\sin x} + \sqrt[3]{x^2 + 3}, & \text{если } x > 2 \\ x^4 + \cos e^x, & \text{если } 1 < x \leq 2 \\ \sqrt{x^4 + 3} + e^{-x + \cos x}, & \text{если } 0 < x \leq 1 \\ 0, & \text{если } x \leq 0 \end{cases}$$

## Задание 4. Создание систем уравнений

1. Создайте новый файл в стиле **article**
2. Подключите пакеты **amsmath** и **amssymb**

3. Для создания системы  $\begin{cases} a_{11}x_1 + a_{12}x_2 = b_1 \\ a_{21}x_1 + a_{22}x_2 = b_2 \\ a_{31}x_1 + a_{32}x_2 = b_3 \end{cases}$  введите следующие команды:

$\$$

$\begin{cases}$

$a_{\{11\}} x_1 + a_{\{12\}} x_2 = b_1 \backslash$

$$\begin{cases} a_{21}x_1 + a_{22}x_2 = b_2 \\ a_{31}x_1 + a_{32}x_2 = b_3 \end{cases}$$

- Для нумерования системы добавьте команду `\eqno` (1) перед закрывающимися знаками долларов.
- Самостоятельно введите команды для задания следующих систем, пронумеруйте их:

$$\begin{cases} a_{11}x + a_{12}y = b_1 \\ a_{21}x + a_{22}y = b_2 \end{cases} \quad \begin{cases} -x + y - z = 1 \\ 2x - y - z = 2 \\ x + 2y + 4z = 0 \end{cases} \quad \begin{cases} -x_1 + x_2 - x_3 + 2x_4 = 1 \\ 2x_1 - x_2 - 2x_3 + x_4 = 0 \\ 3x_1 + x_2 + x_3 - x_4 = -1 \\ 4x_1 + 2x_2 - x_3 - 2x_4 = -2 \end{cases}$$

Для автоматической нумерации формул или систем уравнений их надо поместить в окружение `\begin{equation}` уравнение или система уравнений `\end{equation}`

В стиле документа **article** формулы, системы уравнений нумеруются последовательно во всем документе, начиная с (1), метка состоит просто из одного номера. В стилях **book**, **report** формулы нумеруются последовательно внутри каждой главы, начиная с (X.1), где X – номер главы, а все метки состоят из двух чисел.

Окружение `\begin{eqnarray} . . . \end{eqnarray}` позволяет записывать несколько уравнений без знака системы с нумерацией каждого уравнения

Команда `\nonumber` запрещает нумеровать уравнение своей строки, а замена `eqnarray` на `eqnarray*` запретит нумерацию уравнений.

### Задание 5. Создание систем уравнений

- Создайте новый файл в стиле **article**.
- Подключите пакеты **amsmath** и **amssymb**.

- Для создания системы  $\begin{cases} a_{11}x_1 + a_{12}x_2 = b_1 \\ a_{21}x_1 = b_2 \\ a_{31}x_1 + a_{32}x_2 = b_3 \end{cases}$  введите следующие команды

`\begin{eqnarray}`

```

a_{11} x_1+a_{12} x_2=b_1 \\
a_{21} x_1=b_2 \\
a_{31} x_1+a_{32} x_2=b_3
\end{eqnarray}

```

4. Протранслируйте файл и посмотрите результат.
5. Для отключения нумерации второго уравнения добавьте перед ним команду `\nonumber`.
6. Протранслируйте файл и посмотрите результат

**Справка.** Знак системы можно ставить справа от системы. Для этого надо применить окружение `\begin{array} . . . \end{array}`

#### Задание 6. Создание систем уравнений

1. Создайте новый файл в стиле article
2. Подключите пакеты `amsmath` и `amssymb`

3. Для создания системы уравнений
 
$$\left. \begin{array}{l} a_{11}x_1 + a_{12}x_2 = b_1 \\ a_{21}x_1 = b_2 \\ a_{31}x_1 + a_{32}x_2 = b_3 \end{array} \right\}, \text{ когда знак системы}$$

ставится справа введите следующие команды:

```

$$
\left .
\begin{array}{l}
a_{11} x_1+a_{12} x_2=b_1 \\
a_{21} x_1=b_2 \\
a_{31} x_1+a_{32} x_2=b_2
\end{array}
\right \}
\right .
$$

```

**Справка.** Точка (разделитель) «.» обязательна.

4. Скопируйте вышенаписанные команды и измените их таким образом, чтобы система стала уравнениями с одним номером

```

$$

```

```

\begin{array}{l}
a_{11} x_1 + a_{12} x_2 = b_1 \\
a_{21} x_1 = b_2 \\
a_{31} x_1 + a_{32} x_2 = b_2
\end{array}
\leqno (4)
$$

```

5. Уравнения последней системы выделите полужирными буквами и увеличьте их размер до `\Large`:

```

$$
{\Large \begin{array}{l}
{\bf a}_{11} x_1 + {\bf a}_{12} x_2 = {\bf b}_1 \\
{\bf a}_{21} x_1 = {\bf b}_2 \\
{\bf a}_{31} x_1 + {\bf a}_{32} x_2 = {\bf b}_2
\end{array}}
\leqno (4)
$$

```

6. Скопируйте систему из задания 4. Выделите уравнения цветом. При этом необходимо добавить в преамбулу пакеты для работы с цветом:

```

\usepackage[usenames]{color}
\usepackage{colortbl}
\begin{equation*}
\begin{cases}
\colorbox{green}{\color{blue}$a_{11} x_1 + a_{12} x_2 = b_1$}} \\
\color{blue} a_{21} x_1 + a_{22} x_2 = b_2 \\
\colorbox{red}{ $a_{31} x_1 + a_{32} x_2 = b_2$ }
\end{cases}
\end{equation*}

```

## Задания для самостоятельной работы

<b>Вариант 1</b>	<b>Вариант 6</b>
$\begin{cases} 13x + 6y = 7 \\ 2x - 4y = 6 \end{cases}$	$\begin{cases} 3y - 4x = -13 \\ 3x + 7y = 56 \end{cases}$
<b>Вариант 2</b>	<b>Вариант 7</b>
$\begin{cases} 6x - 5y = 23 \\ y + 3x = 8 \end{cases}$	$\begin{cases} 7x + 3y = 21 \\ 4y - 5x = -15 \end{cases}$
<b>Вариант 3</b>	<b>Вариант 8</b>
$\begin{cases} 2x + 5y = 10 \\ 8y - 5x = 57 \end{cases}$	$\begin{cases} \frac{6}{x} - \frac{8}{y} = -2 \\ \frac{9}{x} + \frac{10}{y} = 8 \end{cases}$
<b>Вариант 4</b>	<b>Вариант 9</b>
$\begin{cases} 2x + 3y = 13 \\ 4x + 5y = 23 \end{cases}$	$\begin{cases} 2x + 3y = 11 \\ 3x + 2y = 9 \end{cases}$
<b>Вариант 5</b>	<b>Вариант 10</b>
$\begin{cases} 2x + 5y = 10 \\ 3x - 2y = 1 \end{cases}$	$\begin{cases} \frac{y}{4} - \frac{x}{5} = 6 \\ \frac{x}{15} + \frac{y}{12} = 0 \end{cases}$

**Многострочные формулы**

Окружение **split** применяется внутри окружения **equation** или **aline** и служит для разбивки длинных формул в несколько строк. При этом нумерация формулы ставится по центру блока, занимаемого формулой.

**Задание 7.** Набрать следующий пример.

```
\begin{equation}
```

```
\begin{split}
```

```
S \leq \int_0^{\phi_0} \int_0^{\rho_0} |
```

```
{\partial \over {\partial \phi}}}
```

$$\begin{aligned}
& \left| \frac{\partial}{\partial \rho} \right| \\
& d\phi d\rho = \int_0^{\phi_0} d\phi \int_0^{\rho_0} \\
& \left| \frac{\partial}{\partial \phi} \right| d\rho \leq \\
& \int_0^{\phi_0} d\phi \int_0^{\rho_0} \left\{ \frac{1}{\sqrt{K_0}} \right\} \\
& \left\{ \sin \left( \sqrt{K_0} \rho \right) \right\} d\rho \\
& = \left\{ \frac{2\phi_0}{K_0} \right\} \sin^2 \left\{ \frac{\sqrt{K_0} \rho_0}{2} \right\} \\
& \end{aligned}$$

### Результат:

$$\begin{aligned}
S & \leq \int_0^{\phi_0} \int_0^{\rho_0} \left| \frac{\partial}{\partial \phi} \wedge \frac{\partial}{\partial \rho} \right| d\phi d\rho = \int_0^{\phi_0} d\phi \int_0^{\rho_0} \left| \frac{\partial}{\partial \phi} \right| d\rho \leq \\
& \int_0^{\phi_0} d\phi \int_0^{\rho_0} \frac{1}{\sqrt{K_0}} \sin \sqrt{K_0} \rho d\rho = \frac{2\phi_0}{K_0} \sin^2 \frac{\sqrt{K_0} \rho_0}{2}
\end{aligned}$$

### Интегралы

Для набора интегралов служат команды **\int** – для одинарного интеграла, **\iint** – для двойного, **\iiint** – для тройного, **\iiiiint** – для четверного и **\idotsint** – произвольного кратного интеграла. Интеграл по контуру обозначается **\oint**. **Примеры:**

$$\int_V f(x) dx \quad \text{\texttt{\$}\$ \int \limits_V f(x) dx \text{\texttt{\$}\$}}$$

$$\int_a^b f(x) dx \quad \text{\texttt{\$}\$ \int_a^b f(x) dx \text{\texttt{\$}\$}}$$

$$\int_a^b f(x) dx \quad \text{\texttt{\$}\$ \int \limits_a^b f(x) dx \text{\texttt{\$}\$}}$$

$$\iint_V f(x) dx \quad \text{\texttt{\$}\$ \iint \limits_V f(x) dx \text{\texttt{\$}\$}}$$

$$\iiint_V f(x) dx \quad \text{\texttt{\$}\$ \iiint \limits_V f(x) dx \text{\texttt{\$}\$}}$$

$$\oint_V f(x) dx \quad \text{\texttt{\$}\$ \oint \limits_V f(x) dx \text{\texttt{\$}\$}}$$

### Задания для самостоятельной работы

<b>Вариант 1</b> $\int_V (x^3 + 2) dx$	<b>Вариант 6</b> $\iint_{(P)} \sqrt{xy - y^2}$
<b>Вариант 2</b> $\int_{-5}^{10} \frac{5x^2 + 2}{4 - x} dx$	<b>Вариант 7</b> $\iiint_{(V)} (x + y + z) dx dy dz$
<b>Вариант 3</b> $\iint_{(P)} \frac{x^2}{y^2} dx dy$	<b>Вариант 8</b> $\iiint_{(V)} (2x + y) dx dy dz$
<b>Вариант 4</b> $\iint_{(P)} \cos(x + y) dx dy$	<b>Вариант 9</b> $\iiint_{(V)} (3x + 2y - z^3) dx dy dz$
<b>Вариант 5</b> $\iint (x^2 + y^2) dx dy$	<b>Вариант 10</b> $\oint_{\Gamma} 2x dx - dy + 4dz$

**Задание 9.** Построить схему командами.

`$$\begin{array}{ccc}`

`S^{\mathcal{W}} \otimes T`

`\stackrel{j}{\longrightarrow} T`

`\Big\downarrow \searrow`

`\Big\downarrow \vcenter{\%`

`\rlap{\scriptstyle{\mathrm{End}}\,P}`

`(S \otimes T)/I = (S \otimes T)/J`

`\end{array}$$`

Команда `\otimes` задает знак  $\otimes$

Команда `\downarrow` задает стрелку вниз  $\downarrow$

#### Верхние и нижние пределы

Знаки суммы `\sum`, интеграла `\int`, предела `\lim` имеют похожее написание пределов. Существует три варианта написания пределов:

1. Команда `$m=\sum_{t=1}^{10}x(t)$` включает формулу в строку в виде

$$\sum_{t=1}^{10} x(t)$$

2. Команда  $\displaystyle m=\sum_{t=1}^{10}x(t)$  включает формулу в строку, но формула записана в выключенной форме:

$$m = \sum_{t=1}^{10} x(t)$$

3. Команда  $m=\sum \limits_{t=1}^{10}x(t)$  аналогична предыдущей, она также включает формулу в строку, но знак суммы меньше

4. В выключенном стиле (межстрочном) команда

$$\$ m=\sum_{t=1}^{10}x(t) \$$$

Во всех примерах, кроме 3, вместо слова `\sum` может стоять любая команда, выводящая математический символ.

### Задания для самостоятельной работы

<b>Вариант 1</b> $\sum_{n=1}^{\infty} \frac{1}{3^{n+1}}$	<b>Вариант 6</b> $\sum_{n=1}^{\infty} \frac{n+3}{5n-1}$
<b>Вариант 2</b> $\sum_{n=1}^{\infty} \frac{1}{(2n+1)(2n+3)}$	<b>Вариант 7</b> $\sum_{n=1}^{\infty} \frac{n^2+3}{2n}$
<b>Вариант 3</b> $\sum_{n=0}^{\infty} \frac{2^n+3}{(2n+1)!}$	<b>Вариант 8</b> $\sum_{n=1}^{\infty} \frac{1}{n^2+1}$
<b>Вариант 4</b> $\sum_{n=1}^{\infty} \frac{2n+1}{n^2(n+1)^2}$	<b>Вариант 9</b> $\sum_{n=1}^{\infty} \frac{3n^2-2}{2^{n+1}}$
<b>Вариант 5</b> $\sum_{n=1}^{\infty} \left( \frac{1}{n^2} - \frac{1}{(n+1)^2} \right)$	<b>Вариант 10</b> $\sum_{n=1}^{\infty} \frac{2n-3}{5n+1}$

**Задание 10.** Научиться задавать верхние и нижние пределы.

1. Создать новый документ в стиле **article**
2. Подключить все необходимые пакеты
3. Ввести всеми способами следующие формулы:

$$\text{a) } \lim_{x \rightarrow 0} \left( \frac{\sin(x)}{x} + x^4 \right) dx$$

$$\text{б) } \sum_{i=1}^k \frac{x^k}{4x^3 + 2x + 8}$$

$$\text{в) } \lim_{\lambda \rightarrow 0} \sum_{i=0}^{n-1} \omega_i \Delta x_i = 0$$

*Подсказка:* стрелка вправо задается командой `\rightarrow`, буква  $\Delta$  - командой `\Delta`, буква  $\omega$  - командой `\omega`, буква  $\lambda$  - командой `\lambda`.

### Задания для самостоятельной работы

<p><b>Вариант 1</b></p> $\lim_{x \rightarrow a} f(x) = b, \lim_{x \rightarrow a} g(x) = c,$	<p><b>Вариант 6</b></p> $\lim_{x \rightarrow 4} \left( \frac{1}{2}x^3 - x + 2 \right)$
<p><b>Вариант 2</b></p> $\lim_{x \rightarrow a} f(x) \pm \lim_{x \rightarrow a} g(x) = b \pm c$	<p><b>Вариант 7</b></p> $\lim_{x \rightarrow 4} \left( \frac{1}{2}x^3 - x + 2 \right) = \frac{1}{2} \cdot 4^3 - 4 + 2 = 30$
<p><b>Вариант 3</b></p> $\lim_{x \rightarrow a} f(x) \cdot \lim_{x \rightarrow a} g(x) = b \cdot c$	<p><b>Вариант 8</b></p> $\lim_{x \rightarrow 3} \frac{x^2 + x + 2}{x^2 + 2x + 8}$
<p><b>Вариант 4</b></p> $\lim_{x \rightarrow a} f(x) = b, \lim_{x \rightarrow a} g(x) = c, (c \neq 0)$	<p><b>Вариант 9</b></p> $\lim_{x \rightarrow 3} \frac{x^2 + x + 2}{x^2 + 2x + 8} = \frac{3^2 + 3 + 2}{3^2 + 2 \cdot 3 + 8} = \frac{14}{23}$
<p><b>Вариант 5</b></p> $\lim_{x \rightarrow a} \frac{f(x)}{g(x)} = \frac{\lim_{x \rightarrow a} f(x)}{\lim_{x \rightarrow a} g(x)} = \frac{b}{c}$	<p><b>Вариант 10</b></p> $\lim_{x \rightarrow 3} \frac{x^2 + x + 2}{x^2 + 2x + 8} = \frac{14}{23}$

### Матрицы

Матрицы задаются следующим окружением:

`$$ \begin{array}{ccc}`

элементы первой строки через знак `&` `\`

элементы второй строки через знак `&` `\`

элементы третьей строки через знак `&`

`\end{array}$$`

Число букв «с» указывает на число столбцов, при этом элементы на каждом месте центрируются. Если вместо буквы «с» написать «r» или «l», то элементы соответствующего столбца примут крайнее правое или левое положение соответственно.

$$\begin{matrix} & a & b & c \\ \text{Задание 11. Ввести матрицу} & d & e & f \\ & t & m & z \end{matrix}$$

1. Создать новый документ в стиле article
2. Подключить все необходимые пакеты
3. Ввести команды

```


$$\begin{array}{ccc}
a&b & c \\
d&e & f \\
t & m & z
\end{array}$$


```

Для получения ровной матрицы знаки **&** желательно ставить в каждой строке друг под другом.

4. Протранслировать файл и посмотреть результаты
5. Самостоятельно создать матрицы:

$\begin{matrix} 5 & -3 & 2 & 1 \\ \text{a)} & 9 & 0 & -17 & 87 \\ 5 & 7 & 3 & 12 \end{matrix}$	$\begin{matrix} \Phi & \Psi & B \\ \text{б)} & \Gamma & \Theta & A \\ T & \Omega & Z \end{matrix}$
$\begin{matrix} a_{13} & a_{12} & a_{13} \\ \text{в)} & a_{23} & a_{22} & a_{23} \\ a_{33} & a_{23} & a_{33} \end{matrix}$	

Для выполнения задания б) воспользуйтесь командами **\Phi**, **\Psi**, **\Gamma**, **\Theta**, **\Omega**, **\Gamma**, **\Theta**, **\Omega**.

Отцентрировать элементы, выровнять их по правому краю, по левому краю.

## Задания для самостоятельной работы

<b>Вариант 1</b> $B_{3 \times 2} \cdot A_{2 \times 3} = D_{3 \times 3} = \begin{pmatrix} 0 & 9 & 6 \\ 2 & 16 & 11 \\ -2 & 2 & 1 \end{pmatrix}$	<b>Вариант 6</b> $A^2 = \begin{pmatrix} 2 & 0 & 1 \\ 1 & -1 & 0 \\ 0 & 3 & 1 \end{pmatrix} * \begin{pmatrix} 2 & 0 & 1 \\ 1 & -1 & 0 \\ 0 & 3 & 1 \end{pmatrix}$
<b>Вариант 2</b> $C^{-1} \cdot C = \frac{1}{13} \cdot \begin{pmatrix} 13 & 0 & 0 \\ 0 & 13 & 0 \\ 0 & 0 & 13 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = E_{(3)}$	<b>Вариант 7</b> $A^3 = A^2 * A = \begin{pmatrix} 4 & 3 & 3 \\ 1 & 1 & 1 \\ 3 & 0 & 1 \end{pmatrix} * \begin{pmatrix} 2 & 0 & 1 \\ 1 & -1 & 0 \\ 0 & 3 & 1 \end{pmatrix}$
<b>Вариант 3</b> $A^{-1} = \begin{pmatrix} \frac{1}{6} & -\frac{1}{2} & \frac{1}{3} \\ \frac{1}{3} & 0 & -\frac{1}{3} \\ \frac{2}{3} & 1 & \frac{2}{3} \end{pmatrix}$	<b>Вариант 8</b> $A^{-1} = \frac{-1}{\det(A)} A^T = \frac{1}{6} = \begin{pmatrix} 1 & -3 & 2 \\ 2 & 0 & -2 \\ -4 & 6 & 4 \end{pmatrix}$
<b>Вариант 4</b> $A^2 = A * A = \begin{pmatrix} 1 & 0 & 2 \\ 4 & -2 & 2 \\ 0 & 1 & -1 \end{pmatrix} * \begin{pmatrix} 1 & 0 & 2 \\ 4 & -2 & 3 \\ 0 & 1 & -1 \end{pmatrix}$	<b>Вариант 9</b> $A E = \begin{pmatrix} 2 & 4 & 1 & 1 & 0 & 0 \\ 0 & 2 & 1 & 0 & 1 & 0 \\ 2 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}$
<b>Вариант 5</b> $A^4 = \begin{pmatrix} 1 & 2 & 0 \\ -4 & 7 & -1 \\ 4 & -3 & 4 \end{pmatrix} * \begin{pmatrix} 1 & 2 & 0 \\ -4 & 7 & -1 \\ 4 & -3 & 4 \end{pmatrix}$	<b>Вариант 10</b> $A^{-1} = -\frac{1}{2} \begin{pmatrix} 4 & -3 \\ -2 & 1 \end{pmatrix}$

## Лабораторная работа 5

### Титульный лист, подчеркивание, рамки, переносы

**Цель:** научиться в LaTeX создавать титульные листы и аннотации, устанавливать интервалы между строками, форматировать абзацы, горизонтальные и вертикальные интервалы, подчеркивание, рамки, переносы слов, разрыв строки

### Теоретические сведения

#### Титульный лист и аннотация

Стандартный заголовок, состоящий из названия, имён авторов и даты создания документа, печатает команда `\maketitle`. Ей должны предшествовать две команды: `\title{title}` и `\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 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 **ru** 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.

Титульный лист приходится создавать вручную, т.к. в LaTeX не предусмотрены специальные команды. При создании титульной страницы следует учитывать, что нумерация страниц начинается с титульной страницы, но на самой титульной странице номер не указывается.

Для форматирования текста использовать разные типы выравнивания текста по горизонтали и вертикальные пробелы «бесконечной» длины.

Для создания информации о студенте и научном руководителе используйте таблицу.

### **Интервалы между строками**

Если нужны большие интервалы между строками, то их значение можно изменить помещением в преамбулу команды

`\linespread{коэффициент}` Для печати «через полтора интервала» пользуйтесь `\linespread{1.3}`, для печати «через два интервала» – `\linespread{1.6}`. По умолчанию этот коэффициент равен 1.

### **Форматирование абзацев**

Два параметра в LaTeX влияют на верстку абзацев. Для изменения внешнего вида абзацев нужно поместить в преамбулу определения вида:

**`\setlength{parindent}{0pt}`**

**`\setlength{parskip}{1ex plus 0.5ex minus 0.2ex}`**

Эти две строчки увеличивают расстояние между абзацами и устанавливают абзацный отступ равным нулю. Однако, имейте в виду, что это влияет также и на оглавление: его строки тоже становятся теперь более разреженными. Чтобы этого избежать, эти команды можно перенести из преамбулы документа куда-нибудь после `\tableofcontents`, или не использовать их совсем, потому что в профессиональной книжной верстке используется выделение абзацев красной строкой, а не пробелами.

Если вы хотите сделать абзацный отступ в не имеющем его абзаце, то вставьте в начало абзаца команду `\indent`. Но эффект от нее будет только если `\parindent` не установлен равным нулю.

Для создания абзаца без отступа первой командой абзаца можно сделать `\noindent`.

Первая строка каждого абзаца печатается с отступом, величина которого задается командой `\parindent`.

Команда `\noindent` подавляет отступ в начале абзаца, перед которым она стоит. Команда `\indent` делает горизонтальный пробел, равный значению `\parindent`. Команда `\indent` не работает в первом абзаце. В этом случае отступ нужно делать командой `\hspace*{\parindent}`

### Горизонтальные интервалы

LaTeX автоматически определяет пробелы между словами и предложениями. Чтобы добавить горизонтальный пробел, пользуйтесь командой `\hspace{длина}`. Если такой интервал должен быть выдержан, даже если он приходится на начало или конец строки, используйте `\hspace*`, а не `\hspace`. В простейшем случае длина – это просто число и единица измерения. Например: `\hspace{1.5cm}` пробел

Команда `\stretch{n}` генерирует специальный «резиновый» пробел. Он растягивается, заполняя все оставшееся места на строке. Если на одной строке

встречаются две команды  $\backslash\text{hspace}\{\text{stretch}\{n\}\}$ , то они растягиваются пропорционально своим коэффициентам.

$x\backslash\text{hspace}\{\text{stretch}\{1\}\}$

$x\backslash\text{hspace}\{\text{stretch}\{3\}\}$

### Вертикальные интервалы

Интервалы между абзацами, разделами, подразделами, ... определяются LaTeX автоматически. При необходимости дополнительный пробел *между двумя абзацами* можно добавить командой  $\backslash\text{vspace}\{\text{длина}\}$ .

Обычно эта команда вставляется между двумя пустыми строчками. Если это пространство должно сохраняться вверху или внизу страницы, используйте вариант команды со звездочкой:  $\backslash\text{vspace}^*$ .

Команду  $\backslash\text{stretch}$  вместе с  $\backslash\text{pagebreak}$  можно применять для верстки текста на последней строке страницы или для вертикального центрирования текста на странице.

Некий текст  $\backslash\text{dots}$

$\backslash\text{vspace}\{\text{stretch}\{1\}\}$

А это окажется на последней строке страницы  $\backslash\text{pagebreak}$ .

Дополнительный пробел между двумя строками *одного* абзаца или внутри таблицы указывается командой  $\backslash\backslash[\text{длина}]$ .

Команда  $\backslash\text{baselinestretch}$  можно использовать для изменения междустрочного интервала. Эта команда задает минимальное расстояние: при вставке внутрь абзаца больших боксов строки автоматически раздвигаются.

### Блоки

LaTeX выстраивает страницы, передвигая блоки. Сначала каждая буква является маленьким блоком, который приклеивается к другим буквам, формируя слово. Слова склеиваются с другими словами, но специальным «эластичным клеем», который может растягиваться или сжиматься, так, чтобы в точности заполнить строку.

Но не только буква может быть блоком. Можно поместить в блок практически все, что угодно, не исключая и другие блоки. Каждый блок затем обрабатывается LaTeX, как если бы это была отдельная буква.

Примерами могут быть окружение **tabular** или **includegraphics**, оба производящие блок. Это значит, что легко можно сверстать рядом две таблицы или иллюстрации. Только убедитесь, что их общая ширина не превышает **\textwidth**.

Вы также можете упаковать любой абзац в блок или командой **\parbox[*поз*]{*ширина*}{*текст*}** или окружением **\begin{minipage}[*поз*]{*ширина*} текст \end{minipage}**

**\begin{minipage}[*поз*]{*ширина*} текст \end{minipage}**

Параметр *поз* может принимать одну из букв **c**, **t** или **b**, контролируя вертикальное выравнивание блока по отношению к базовой линии окружающего текста. **Ширина** принимает аргументом длину, определяющую ширину блока.

В то время, как **\parbox** упаковывает целый абзац, разбивая строчки и прочее, существует класс блоковых команд, работающих только на горизонтально расположенном материале. Одну из них называют **\mbox** и просто упаковывает последовательность блоков, что можно использовать для предотвращения переноса двух слов. Так как вы можете помещать одни блоки в другие, эти упаковщики горизонтальных блоков чрезвычайно гибки.

**\makebox[*ширина*][*поз*]{*текст*}**

**Ширина** определяет ширину результирующего блока так, как его видно снаружи. Кроме выражений длины, вы тут можете использовать **\width**, **\height**, **\depth** и **\totalheight**. Они устанавливаются равными значениям, полученным измерением параметров *текста*. Параметр *поз* принимает однобуквенное значение: **c**: центрировать, **l**: отжать влево, **r**: отжать вправо или **s**: равномерно заполнить блок текстом.

Команда **\framebox** работает в точности так же, как **\makebox**, но рисует рамку вокруг текста.

Следующий пример показывает некоторые возможности использования команд `\makebox` и `\framebox`. Пример:

```
\makebox[\textwidth]{%
  ц е н т р}\par
\makebox[\textwidth][s]{%
  р а с т я н у т ы й}\par
\framebox[1.1\width]{Я теперь
  в рамке!} \par
\framebox[0.8\width][r]{Ой,
  я слишком толстый} \par
\framebox[1cm][l]{ничего,
  я тоже}
```

Результат показан на Рисунке 14.

ц е н т р

р а с т я н у т ы й

Я теперь в рамке!

Ой, я слишком толстый

ничего, я тоже

Рисунок 14. – Использование команд `\makebox` и `\framebox`

Теперь посмотрим как управлять вертикалью. Команда `\raisebox{сдвиг}[глубина][высота]{текст}`

позволяет определить вертикальные характеристики блока. В первых трех параметрах можно использовать `\width`, `\height`, `\depth` и `\totalwidth`, чтобы получить размеры аргумента *текст*.

```
\raisebox{0pt}[0pt][0pt]{\Large%
\textbf{Aaaa}\raisebox{-0.3ex}{a}%
\raisebox{-0.7ex}{aa}%
\raisebox{-1.2ex}{a}%
\raisebox{-2.2ex}{a}%
\raisebox{-4.5ex}{a}}
```

кричал он, но даже стоящий рядом

не заметил, что с ней случилось  
что-то ужасное.

**Аааааааа** кричал он, но даже стоящий рядом не заметил, что  
с ней случилось что-то ужасное.  
а

## Переносы слов

Команда `\-` указывает, в каком месте слова `LaTeX` имеет право делать перенос. Она незаменима, когда в русском тексте используются иностранные слова. Например: `ter\ra in\cog\ni\ta`. Команда `\-` разрешает делать переносы даже в словах, которые печатаются машинописным шрифтом. Для часто используемых слов правила переноса можно объявить в преамбуле входного файла декларацией `\hyphenation{words}`. Слова в аргументе декларации отделяются друг от друга пробелом, а допустимые места переноса указываются дефисом. Например, `\hyphenation{Micro-soft office}` разрешает переносить «soft» на новую строку в слове «Microsoft» и запрещает любые переносы в слове «office».

## Разрыв строки

Команды `\linebreak[n]` и `\nolinebreak[n]` позволяют регулировать разбиение текста между словами на строки. `\linebreak[n]` стимулирует `LaTeX` завершить текущую строку, а `\nolinebreak[n]` – продолжить её. Опция `n` – целое число от 0 до 4 – задаёт «силу» команды: чем больше значение `n`, тем выше вероятность выполнения команды. Значение 0 равносильно отсутствию команды, а значение 4 принуждает `LaTeX` выполнить её. По умолчанию действует значение 4.

Разрыв строки между словами можно запретить, связав их пробелом нормального размера, который делает команда `~`.

Команды `\[length]` и `\*[length]` обрывают верстку текущей строки и начинают новую строку, причем `*`-форма запрещает перенос новой строки на следующую страницу. Опция `length` задает величину дополнительного вертикального пробела перед новой строкой.

### Задания для самостоятельной работы

1. Создать, отформатировать реферат по теме, связанной с информационными технологиями.

Реферат должен содержать:

- титульный лист;
- оглавление;
- введение (1-2 страницы);
- основную часть (15-20 страниц);
- заключение (1-2 страницы);
- список использованных источников.

Каждая из этих частей (и каждая глава в основной части) начинается с новой страницы.

1. Страницы реферата нумеруются справа в верхней части страницы; титульный лист входит в общую нумерацию, но сам номер на нём не ставится.

Титульный лист должен содержать:

- название министерства (МИНПРОСВЕЩЕНИЯ РОССИИ);
- название организации;
- слово «реферат», название предмета;
- название реферата (шрифт 28, полужирный);
- фамилия и имя автора;
- фамилия, имя и отчество руководителя;
- в последних двух строчках – город и год, выравнивание по центру.

6. Оглавление строится автоматически на основе включенных в реферат заголовков разных уровней (*Заголовок 1, Заголовок 2* и т.д.).

7. Во введении дается краткая характеристика изучаемой темы, обосновывается ее актуальность и практическая значимость (где можно использовать). Часто бывает удобно писать введение уже после того, как реферат будет готов.

8. Основная часть делится на главы (разделы), которые могут в свою очередь делиться на подразделы. Здесь нужно изложить различные точки зрения на проблему и собственную позицию автора реферата.

9. В заключении подводятся итоги исследования, делаются выводы, формулируются новые результаты, полученные в ходе выполненной работы.

10. В списке использованных источников перечисляются все материалы, использованные при составлении реферата: книги, статьи, интернет-сайты, электронные ресурсы и др. Работы в списке перечисляются в алфавитном порядке по фамилии автора, работы одного автора – по возрастанию года издания. В конце списка перечисляются источники на иностранных языках и интернет-ресурсы.

## Лабораторная работа 6

### Рисунки и графика

**Цель:** научиться создавать псевдорисунки (линии, отрезки, стрелки, окружности, круги и овалы, кривые и пр.), построение графа и импортировать готовые изображения, графику.

### Теоретические сведения

#### Псевдорисунки

LaTeX позволяет использовать примитивные рисунки, состоящие из прямых, наклонных линий, стрелок и окружностей. Для псевдорисунков используется окружение `{picture}`:

```
\begin{picture}(110,50)
```

...

```
\end{picture}
```

В скобках задаётся размер канвы – ширина, высота (по умолчанию размеры измеряются в пунктах, поменять можно в преамбуле документа командой `\unitlength`).

Если нужно, чтобы длины измерялись в миллиметрах, надо написать в преамбуле `\unitlength=1mm`.

**Пример:**

```
\begin{picture}(110,50)
```

```
\put(55,15){Стрелка}
```

```
\put(55,15){\vector(-2,1){40}}
```

```
\put(0,0){\line(1,0){20}}
```

```
\put(0,0){\line(0,1){20}}
```

```
\end{picture}
```

**Результат:**



На каждый псевдорисунок LATEX должен отвести в тексте определенное место. Эти размеры задаются в круглых скобках через запятую немедленно после `\begin{picture}`, сначала ширина, затем высота. Между скобками, запятой и числами, задающими размеры псевдорисунка, не должно быть пробелов.

Чтобы поместить что-то на этот псевдорисунок, используется команда `\put`. После `\put` в круглых скобках через запятую следуют координаты того объекта, который мы размещаем на псевдорисунке (сначала абсцисса, затем ордината; началом координат по умолчанию считается левый нижний угол псевдорисунка), а затем, без пробела, в фигурных скобках, – тот объект, который надо нанести.

Для первой из команд `\put` этот объект был просто текстом, и соответственно в фигурных скобках только этот текст и был; для остальных трех команд, размещавших на рисунке стрелку и два отрезка, в фигурных скобках помещается нечто более сложное: описание этой стрелки и отрезков.

Уголок в приведенном примере – не что иное, как левый нижний угол псевдорисунка (точка с координатами (0,0)).

Когда мы говорили о координатах объекта, имелись в виду координаты так называемой точки отсчета. на этом объекте. Если объект – текст, то точка отсчета — его левый нижний угол. Иногда при размещении текста удобнее задать координаты его правого, а не левого нижнего угла. Чтобы так сделать, можно воспользоваться командой `\llap` с одним аргументом — текстом, чья точка отсчета будет в правом нижнем углу. В следующем примере точка отсчета .полужирной. кошки будет в левом нижнем углу, а .рубленной. — в правом нижнем. Например:

```
\begin{picture}(110,40)
\put(82,20){\textbf{Кошка}}
\put(80,20){\llap{\textsf{Кошка}}}}
\end{picture}
```

Результат:

**Кошка** Кошка

Внутри окружения `picture` не должно быть пустых строк!

Чтобы поместить что-либо на канву, используется команда `\put(x,y){<объект>}`.  $(x,y)$  – координаты объекта (началом координат считается левый нижний угол канвы) В фигурных скобках – тот объект, который нужно нанести. Можно использовать следующие объекты:

- **Надпись.** Любой текст, например, `\put(10,15){Пример текста}`
- **Линия.** Пример: `\line(1,-2){20}`. Здесь  $1/-2$  – угловой коэффициент отрезка, 20 – длина проекции на ось абсцисс.
- **Стрелка.** Стрелка задаётся командой `\vector`. Параметры те же, что и у линии.
- **Окружность.** Команда: `\circle{<радиус>}`.
- **Круг.** Команда: `\circle*{<радиус>}`.
- **Овал** – прямоугольник с закруглёнными краями: `\oval(<ширина>,<высота>)`.
- **Кривые Безье.** Пример: `\qbezier(22,2)(120,20)(20,77)` – в скобках координаты опорных точек.

### Отрезки и стрелки

Отрезки задаются с помощью команды `\line`. Надо установить *наклон* и *размер отрезка*. Вот пример команды `\put`, выводящей отрезок:

<code>\begin{picture}(100,50)</code>	<b>Результат:</b>
<code>\put(60,50){\line(1,-2){20}}</code>	
<code>\end{picture}</code>	

Здесь на рисунок размера  $100 \times 50$  пунктов наносится отрезок с началом в точке  $(60,50)$ . Наклон отрезка задается парой целых чисел, расположенных в круглых скобках через запятую непосредственно после `\line`. Отношение этих чисел должно быть равно угловому коэффициенту отрезка (тангенсу угла наклона к горизонтали); в нашем случае эти числа суть  $(1,-2)$ , это означает, что отрезок отклоняется на одну единицу вправо и на две единицы вниз. Если эти числа  $(1,0)$ , то отрезок горизонтален, если  $(0,1)$ , то отрезок вертикален.

Размер отрезка задается в фигурных скобках после круглых скобок, в которых задан наклон. Этот размер не его длина, но длина его проекции на горизонтальную ось (кроме случаев, когда отрезок вертикален — тогда задается его длина по вертикали).

Длину отрезка можно (если она не слишком мала) задавать произвольно, а вот наклон — нет. Каждое из целых чисел, задающих наклон, не должно превосходить **6** по абсолютной величине, и, кроме того, эти два числа не должны иметь общих делителей, кроме 1 (это последнее условие репертуар возможных наклонов не ограничивает).

Стрелки задаются с помощью команды `\vector`. Синтаксис этой команды совершенно такой же, как у команды `\line`: в круглых скобках пишется пара чисел, задающая наклон стрелки, а затем в фигурных скобках параметр, задающий ее размер (длина проекции на горизонтальную ось, если стрелка не вертикальна, и длина проекции на вертикальную ось, если стрелка вертикальна). Отличие от команды `\line` в том, что репертуар возможных наклонов стрелок еще более ограничен, чем у отрезков: целые числа, задающие наклон, не должны превосходить **4** по абсолютной величине (и по-прежнему не должны иметь общих делителей). Точкой отсчета стрелки является ее начало.

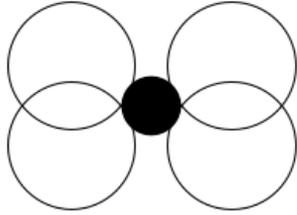
### Окружности, круги и овалы

Окружность задается командой `\circle`, а круг (сплошной черный кружок) — ее вариантом со звездочкой. `\circle*`. У этих команд единственный аргумент — диаметр круга или окружности. Как обычно, он задается в единицах, равных значению параметра `\unitlength` (по умолчанию — в пунктах). Точкой отсчета окружности или круга является центр. Вот пример картинki с окружностями и кругами:

```
\begin{picture}(100,80)
\put(30,30){\circle{30}}
\put(70,30){\circle{30}}
\put(30,50){\circle{30}}
\put(70,50){\circle{30}}
```

```
\put(50,40){\circle*{20}}
\end{picture}
```

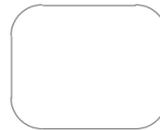
**Результат:**



Наряду с окружностями и кругами, на псевдорисунок можно нанести также овал. Он задается командой `\oval`, аргументы которой — ширина и высота овала. Эти аргументы задаются в круглых скобках через запятую. Точка отсчета овала — его центр. Пример:

```
\begin{picture}(100,80)
\put(50,40){\oval(100,80)}
\end{picture}
```

**Результат:**



Кроме того, возможны и неполные овалы, представляющие собой половины или четверти от полных. Чтобы задать такой неполный овал, надо задать команде `\oval` необязательный аргумент (в квадратных скобках, после обязательного). Для задания половины овала этот аргумент должен быть одной из следующих букв:

- **t** верхняя половина;
- **b** нижняя половина;
- **r** правая половина;
- **l** левая половина.

Для задания четверти овала необязательный аргумент команды `\oval` должен быть сочетанием двух из этих букв (например, `tr` для верхней правой четверти). Точка отсчета усеченного овала расположена там же, где точка отсчета соответствующего ему полного овала. Вот пример картинки с усеченными овалами:

```
\begin{picture}(100,80)
```

```
\put(50,40){\oval(80,60)[t]}
```

```
\put(50,40){\oval(80,60)[br]}
\end{picture}
```

Результат:



## Кривые

При пользовании окружением **picture** вы имеете возможность нанести на псевдорисунок кривую более или менее произвольной формы (эти кривые — так называемые квадратичные сплайны Безье). Это делается с помощью команды **\qbezier**. Вот пример ее работы:

```
\begin{picture}(80,80)
\qbezier(22,2)(120,20)(20,77)
\put(22,2){\circle*{5}}
\put(120,20){\circle*{5}}
\put(20,77){\circle*{5}}
\end{picture}
```

Результат:



После **\qbezier** надо указать (без пробелов) координаты трех точек: начальной, опорной и конечной. Из начальной точки кривая выходит, устремляется к опорной, но, как правило, до нее не доходит, поскольку сворачивает к конечной точке, в которой и заканчивает свой путь. В нашем примере мы для ориентировки нанесли на псевдорисунок черные кружки в этих трех точках.

Можно попросить LaTeX не так густо ставить квадратики, из которых состоит кривая. Для этих целей у команды **\qbezier** предусмотрен необязательный аргумент — количество этих квадратиков. Он ставится перед всеми обязательными в квадратных скобках:

```
\begin{picture}(80,80)
\qbezier(22,2)(120,20)(20,77)
\qbezier[60](58,2)(-40,20)(60,77)
\end{picture}
```

Результат:



Кстати, обратите внимание, что опорная точка второй из наших кривых находится где-то за пределами текста. Это не страшно, поскольку ее координаты используются LaTeX только для расчетов.

Какой бы необязательный аргумент команды `\qbezier` мы ни задавали, количество квадратиков, из которых составляется кривая, не превысит числа 500. Если вы решили увеличить этот максимум, допустим, до тысячи, надо написать так:

```
\renewcommand{\qbeziermax}{1000}
```

Если так вы напишете в преамбуле, то предел 1000 будет относиться ко всем кривым в вашем тексте, а если внутри группы (например, внутри окружения), то изменение этого параметра забудется по выходе из группы.

### **Дополнительные возможности**

Иногда бывает нужно нанести на псевдорисунок несколько регулярно расположенных объектов. В этом случае, вместо того чтобы много раз писать `\put`, удобно воспользоваться командой `\multiput`. Она располагает на псевдорисунке несколько одинаковых объектов на равных расстояниях. Синтаксис этой команды таков:

```
\multiput(x,y)(_x,_y){n}{объект}
```

Здесь `x` и `y` – координаты первого из размещаемых объектов (как и в обычной команде `\put`), `_x` и `_y` – расстояния, на которые каждый следующий объект будет сдвинут относительно предыдущего по горизонтали и вертикали, `n` – количество объектов, которые надо разместить, и, наконец, объект – это, как и у команды `\put`, описание того, что мы размещаем на рисунке. Пример:

```
\begin{picture}(100,80)  
\multiput(10,70)(8,-6){8}{%  
{\circle*{3}}  
\end{picture}
```

**Результат:**



Кстати, знак процента нужен для переноса строки – окружение **picture** не допускает пустых строк, т.е. либо весь код должен идти одной строкой, либо в конце каждой ставится процент.

Вот еще один пример; здесь с помощью команды **\multiput** рисуется решеточка:

```
\begin{picture}(100,50)
\multiput(0,0)(10,0){10}%
{\line(1,5){10}}
\multiput(0,0)(2,10){6}%
{\line(1,0){90}}
\end{picture}
```

Результат:



Иногда, когда псевдорисунок достаточно сложен, удобно применить следующий прием: задать в качестве аргумента одной из команд **\put** целое окружение **picture** (точкой отсчета будет служить левый нижний угол). При этом можно отсчитывать координаты объектов на подрисунке относительно самого подрисунка, а не внешнего рисунка, что часто бывает проще; кроме того, если понадобится сдвинуть этот подрисунок, как единое целое, то для этого будет достаточно изменить аргументы в одной-единственной команде **\put**.

Вот пример рисунка с подрисунком (будем считать, что это классная доска, на которой нарисованы оси координат Рисунок 15):

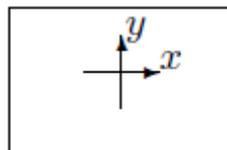


Рисунок 15. – Создание рисунков и подрисунков

Этому рисунку соответствовал такой исходный текст:

```
\begin{picture}(120,80)
% Края доски:
\put(0,0){\line(1,0){120}} \put(0,80){\line(1,0){120}}
```

```

\put(0,0){\line(0,1){80}} \put(120,0){\line(0,1){80}}
% Оси координат:
\put(40,25){\begin{picture}(40,40)%
\put(20,0){\vector(0,1){40}}
\put(0,20){\vector(1,0){40}}
\put(40,22){$x$} \put(22,40){$y$}
\end{picture}}
\end{picture}

```

Размеры внутренней картинке можно задать совершенно произвольно, например, (200,200) или даже (0,0) — команда `\put` бездумно размещает объекты таким образом, чтобы их точки отсчета имели указанные координаты, и при этом не интересуется, сколько места они реально занимают и не наложится ли на текст или другие объекты.

Нередко требуется сдвигать не какую-то часть псевдорисунка, а весь псевдорисунок как целое (например, если вы ищете оптимальное расположение иллюстрации по отношению к тексту). Для этого удобно использовать еще одну возможность окружения `picture`: можно задать его таким образом, чтобы начало координат было не в левом нижнем углу, а в любой другой точке. Для этого после `\begin{picture}` надо задать не одну, а две пары чисел в круглых скобках. В этом случае первая пара чисел будет, как и прежде, обозначать ширину и высоту места, выделяемого на псевдорисунок, а вторая пара будет указывать, каковы координаты левого нижнего угла этого псевдорисунка (по умолчанию, т. е. если второй пары чисел в круглых скобках нет, они были бы просто (0,0)). Главное только — не напутать со знаками: если вы сказали `\begin{picture}(a,b)(x,y)`, то это значит, что левый нижний угол псевдорисунка имеет координаты (x, y), стало быть, по сравнению со случаем, когда  $x = y = 0$ , весь псевдорисунок сдвинется на  $-x$  по горизонтали и на  $-y$  по вертикали.

Пример, в котором второй псевдорисунок сдвигается на 20 единиц вправо и на 10 единиц вверх по отношению к первому:

```

\begin{picture}(150,80)

```

```

\put(0,0){\line(1,0){140}}
\put(0,70){\line(1,0){140}}
\put(0,0){\line(0,1){70}}
\put(140,0){\line(0,1){70}}
\put(25,30){\Huge Сдвиг}
\end{picture}\[25pt]
\begin{picture}(150,80)(-20,-10)
\put(0,0){\line(1,0){140}}
\put(0,70){\line(1,0){140}}
\put(0,0){\line(0,1){70}}
\put(140,0){\line(0,1){70}}
\put(25,30){\Huge Сдвиг}
\end{picture}

```

Результат работы программного кода представлен на рисунке 16.



Рисунок 16. – Сдвиг рисунков

### **XY-pic**

XY-pic – это пакет для создания графов и диаграмм.

Графы строятся в виде матрицы, где каждый элемент матрицы соответствует вершине графа. Рёбра графа строятся с помощью специальных команд.

### **Подключение пакета XY-pic**

В преамбуле документа написать:

```

\input xy
\xyoption{all}

```

Либо `\usepackage[all]{xy}`.

### Построение графа

Использовать команду `\xymatrix{ ... }`. Внутри окружения описывается матрица. Элементы матрицы в строке разделены символом "&". Строки разделены "\". Пример:

```
\xymatrix{ U \ar@/_/[ddr]_y
\ar@/^/[drr]^x
\ar@{>}[dr]|-(x,y) \\\
& X \times Z Y \ar[d]^q \ar[r]_p \\
& X \ar[d]_f \\\
& Y \ar[r]_g & Z }
```

Результат показан на Рисунке 17.

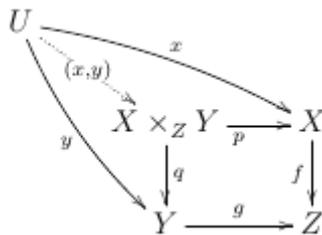


Рисунок 17. – Результат программного кода построения графа

Из примера видно, что *стрелки* строятся командой `\ar`. У команды много модификаций:

В квадратных скобках можно задать *направление стрелки* – `\ar[hop]`. Варианты u, d, l, r, ur, ul, dl, dr, drr и так далее. Пример: `\ar[ur]`.

Можно задать стиль стрелки – `\ar@style[hop]`. Некоторые варианты: `@{=>}`, `@{>}`, `@{:>}`, `@{~>}`, `@{-->}`, `@{-}`, `@{}`. Не путать со смайлами)).

Над стрелками (или под ними) можно размещать текст или другие объекты. `"^"` – *метка сверху*, `"_"` – *метка снизу*, `"|"` – *разорвёт стрелку*. Пример: `\xymatrix@1{ X \ar[r]^a_b & Y & Z \ar[l]_A_B }`

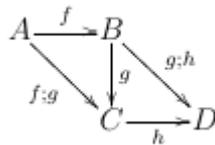
*Дуги:* `@/^/`, `@/_/`, `@/_1pc/` и т.п.

Несколько примеров:

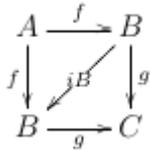
```
\xymatrix{ \bullet \var@{-}[r] & \bullet \var@{.}[d] \\ \bullet \var@{--}[u] & \bullet \var@{->}[l] \var@{=}[ul] }
```



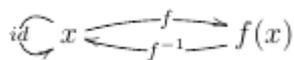
```
\xymatrix@1{ A \var[r]^f \var[dr]_{f;g} & B \var[d]^g \var[dr]^{g;h} \\ & C \var[r]_h & D }
```



```
\xymatrix{ A \var[d]_f \var[r]^f & B \var[d]^g \var[dl]_{iB} \\ B \var[r]_g & C }
```



```
\xymatrix{ x \var@(ul,dl)[id] \var@/^/[rr]f & f(x) \\ \var@/^/[ll]{f^{-1}} & }
```



### Импортированная графика

В документ можно импортировать готовые изображения (как растровые, так и векторные). Но для этого сначала нужно подключить драйвер (**dvips**, **pdftex** или др.). Для подключения драйвера используем следующую конструкцию в преамбуле документа:

```
\ifx\pdfoutput\undefined
\usepackage{graphicx}
\else
\usepackage[pdftex]{graphicx}
\fi
```

Чтобы вставить изображение, используем команду `\includegraphics`.

Синтаксис команды таков:

`\includegraphics[keyval-list]{file}`,

где **file** – имя файла, а **keyval-list** – список ключей, которые задаются в виде **key=value** через запятую.

Расширение файла с рисунком в команде `\includegraphics` можно не указывать, поскольку драйвер сам знает, какие типы файлов он может обработать, а какие нет. Для драйвера **dvips** это файлы с расширением **eps**, **ps**, **eps.gz**, **ps.gz**, **eps.Z**, а для драйвера **pdftex** – **png**, **pdf**, **jpg**, **mps**, **tif**. Когда расширение файла в команде не указано, драйвер последовательно добавляет к имени файла все известные ему расширения, пока не найдёт первый подходящий файл. Пример:

`\includegraphics{01}`

**Возможные ключи:**

**width=length** – устанавливает в любых LaTeX'овских единицах длины ширину области, выделяемой для рисунка. Пример:

`\includegraphics[width=1in]{01}`

**height=length** – устанавливает высоту рисунка. Пример:

`\includegraphics[width=1in,height=10mm]{01}`

**totalheight=length** – полная высота.

**Keepaspectratio** – сохраняет пропорцию рисунка при изменении ширины и/или высоты. Пример:

`\includegraphics[width=1in,%  
height=1cm,keepaspectratio]{01}`

**scale=scale** – масштабирование рисунка в **scale** раз. Пример:

`\includegraphics[scale=0.5]{01}`

**viewport=llx lly urx ury**

**trim=dl db dr du** – эти ключи задают так называемую видимую область рисунка. Здесь **llx lly urx ury** — это x- и y-координаты левого нижнего и правого верхнего углов видимой области рисунка относительно точки отсчёта, а **dl db dr du** – это расстояния между левыми, нижними, правыми и верхними границами видимой области рисунка и самого рисунка. **Пример:**

`\includegraphics[viewport=-5 -5 50 50]{01}`

**clip=boolean** – если значение равно true, то данный ключ отсекает часть рисунка, выходящую за границы видимой области. **Пример:**

`\includegraphics[viewport=-5 -5 50 50,clip]{01}`

**angle=angle** – этот ключ поворачивает рисунок на **angle** градусов против часовой стрелки. Ось вращения проходит через точку отсчёта бокса. **Пример:**

`\includegraphics[scale=0.3,angle=30]{01}`

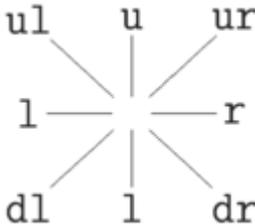
**draft** – на стадии подготовки документа можно использовать этот ключ. Он указывает, что вместо рисунка надо начертить рамку и напечатать внутри неё имя файла. **Пример:**

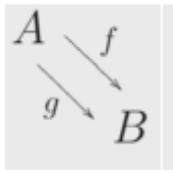
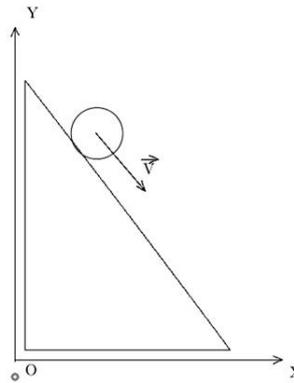
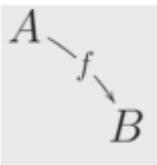
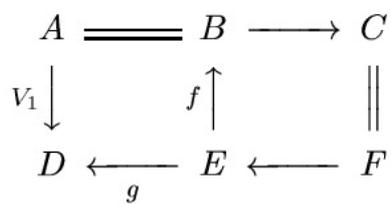
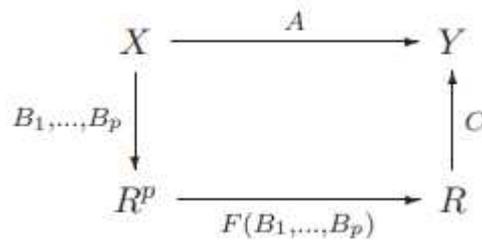
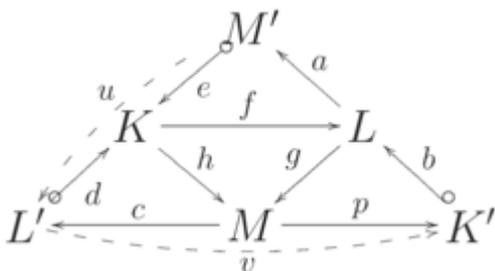
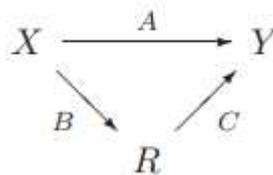
`\includegraphics[scale=0.5,draft]{01}`

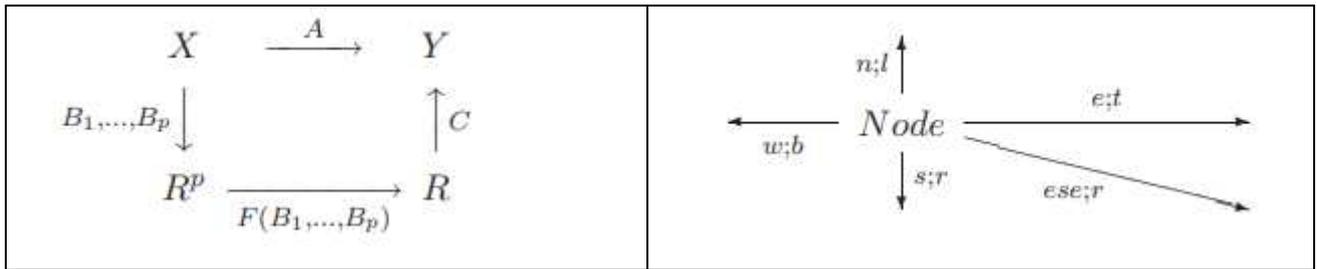
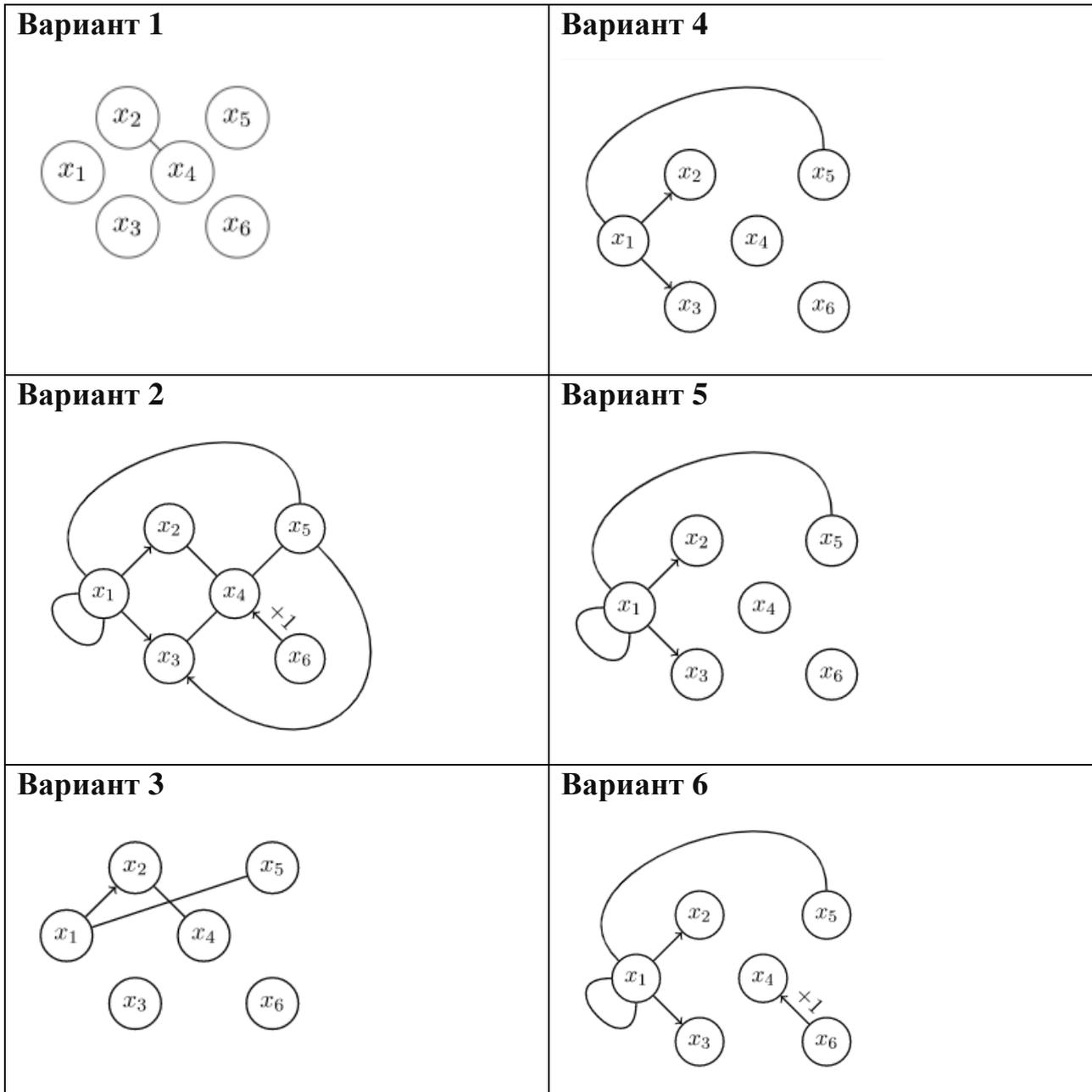
### Задания для самостоятельной работы

Создать схему, коммуникативную диаграмму в соответствии с вариантом.

#### Задание 1

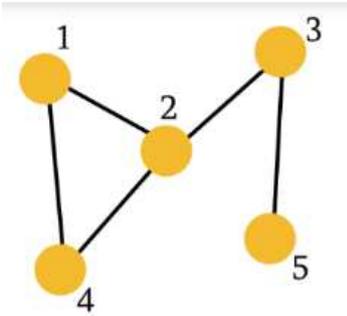
Вариант 1	Вариант 4
	

**Вариант 2****Вариант 5****Вариант 3****Задание 2:****Вариант 1****Вариант 4****Вариант 2****Вариант 5****Вариант 3****Вариант 6**

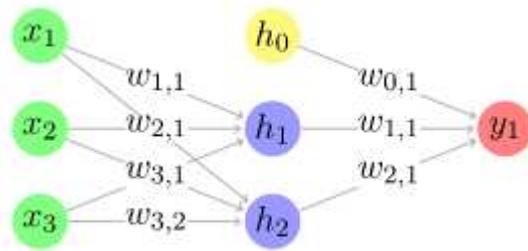
**Задание 3:**

## Задание 4:

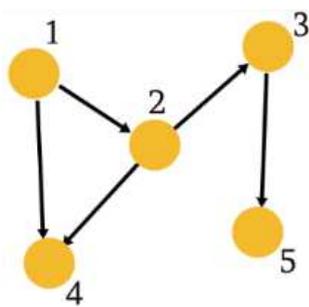
Вариант 1



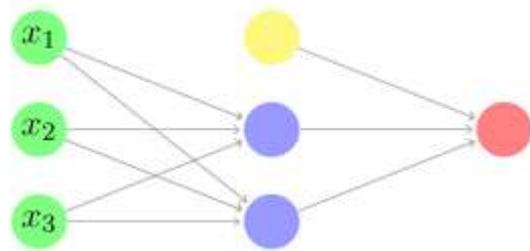
Вариант 4



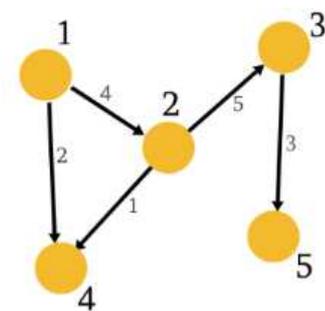
Вариант 2



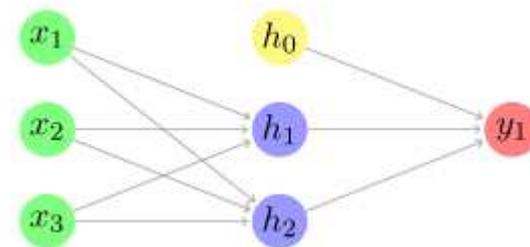
Вариант 5



Вариант 3



Вариант 6



## Лабораторная работа 7

### Сервисы для документов LaTeX

**Цель:** научиться работать с готовыми онлайн приложениями, позволяющими автоматизировать ввод символов, создание формул и разработку таблиц в LaTeX документе.

#### Теоретические сведения

Существуют множество различных приложений, позволяющих автоматизировать разработку LaTeX документов. Рассмотрим три наиболее значимые из них: **Detexify** (ввод символов), **mathURL** (редактирование уравнений) и **Tables Generator** (создания таблиц).

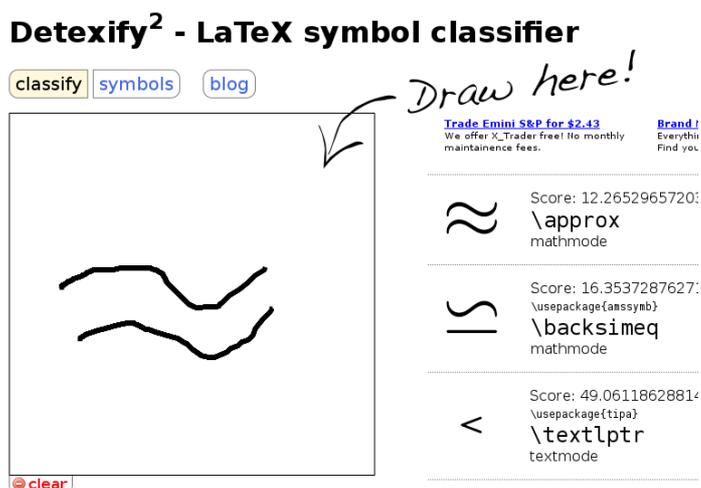
#### Detexify

В LaTeX слишком много символов, поэтому найти нужный символ сложно. Если вы не можете выразить символ словами, вы не можете найти его в Google. Для быстрого поиска достаточно записать, как выглядит типизируемый символ, и искать символ через символ можно с помощью онлайн приложения Detexify. Detexify позволяет нарисовать нужный символ и показывает LaTeX код для него! Возможности поиска символа с помощью приложения Detexify показана на рисунке 18.

**Detexify<sup>2</sup> - LaTeX symbol classifier**

classify symbols blog

*Draw here!*



$\approx$	Score: 12.2652965720: <code>\approx</code> mathmode
$\backsimeq$	Score: 16.3537287627: <code>\usepackage{ansymb}</code> <code>\backsimeq</code> mathmode
$\lt$	Score: 49.0611862881: <code>\usepackage{tipa}</code> <code>\textlptr</code> textmode

clear

Рисунок 18. – Приложение Detexify

**Detexify** – по сути, является упражнением в машинном обучении. В конечном итоге вы изучаете команды для символов, о которых даже не подозревали.

Вы можете щелкнуть по какому либо символу, чтобы сказать системе: «Эта закорючка, которую я нарисовал – это то, что я имел в виду», тем самым помогая обучить приложение.

Однако рисовать даже простые символы с помощью мыши довольно сложно, можно это сделать и пальцем или стилусом, поэтому что выпустили версию **Texify** для мобильных телефонов **Android**. Приложение для Android работает точно так же, как и веб-версия, и подключается к серверу для фактического распознавания. Ссылка на приложение **DeTeXify!** – <http://detexify.kirelabs.org/classify.html>.

## mathURL

Для редактирования уравнений в реальном времени рекомендуется приложение **mathURL** – набор формул. Ссылка на приложение: <http://mathurl.com>. Приложение представлено на рисунке 19.

**mathURL** редактирование уравнений в реальном времени · постоянные короткие ссылки · ввод LaTeX+AMS

$$\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s}$$

`\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s}`

макет	
$a^b$	$\frac{a}{b}$
$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$	$a = b + c$ $= d$ $a = \begin{cases} b & \text{if } x \\ c & \text{if } y \end{cases}$
• bulleted	1. numbered
• list	2. list
буквы и символы	
операторы и отношения	
пунктуация и акценты	
функции	
форматирование	
общие формы	

Рисунок 19. – Приложения для создания формул mathURL

## Приложение для создания таблиц Tables Generator

**Tables Generator** – приложение для создания таблиц. Ссылка: [http://www.tablesgenerator.com/latex\\_tables](http://www.tablesgenerator.com/latex_tables).

Интерфейс приложения показан на рисунке 20.

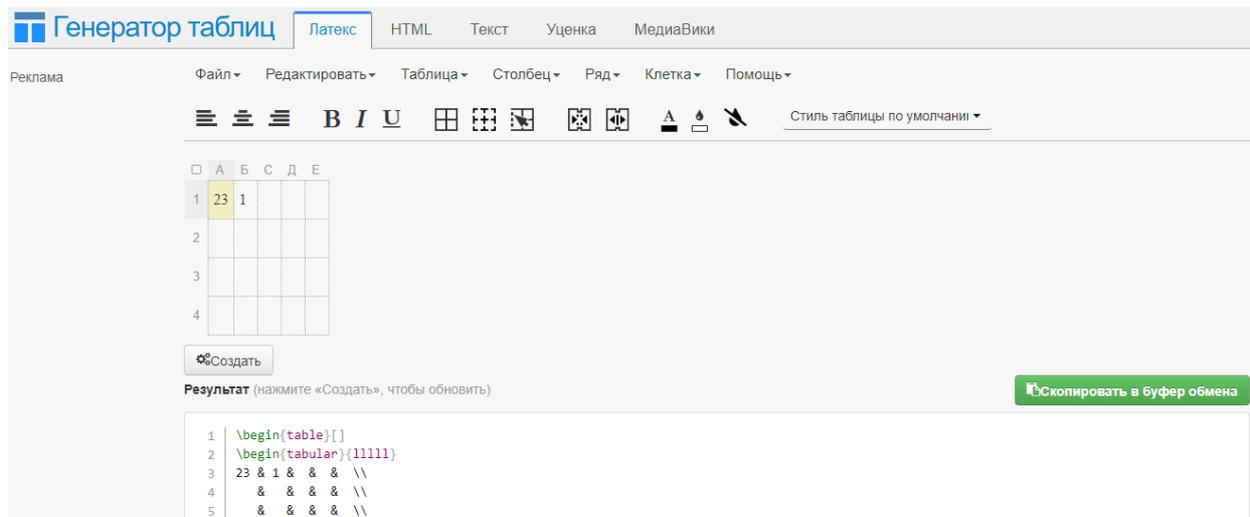


Рисунок 20. – приложение для создания таблиц

## Практические задания

### Вариант 1

#### Задание 1

**Определение 2.1.** Множество  $Y$  называется *подмножеством* множества  $X$ , если любой элемент множества  $Y$  является элементом множества  $X$ . Это обозначается записью  $Y \subseteq X$ .

В кванторах это определение можно записать следующим образом:

$$\forall y \in Y : y \in X.$$

Например,  $\{1, 2\} \subseteq \{1, 2, -\frac{1}{2}\}$ ,  $\mathbb{N} \subseteq \mathbb{Z}$ ,  $\mathbb{Z} \subseteq \mathbb{Q}$ ,  $\mathbb{Q} \subseteq \mathbb{R}$ .

Из определения подмножества следует, что всякое множество является подмножеством самого себя, а пустое множество является подмножеством любого множества.

#### Задание 2

**Определение 6.1.** Число  $a$  называется *пределом* последовательности  $\{x_n\}$  при  $n \rightarrow \infty$ , если

$$\forall \varepsilon > 0 \quad \exists N(\varepsilon) \in \mathbb{N} \quad \forall n > N(\varepsilon) \Rightarrow |x_n - a| < \varepsilon.$$

Тот факт, что  $a$  – предел последовательности  $\{x_n\}$  при  $n \rightarrow \infty$ , обозначается следующим образом:

$$\lim_{n \rightarrow \infty} x_n = a.$$

Определение также может быть сформулировано на языке окрестностей:

$$\forall \varepsilon > 0 \quad \exists N(\varepsilon) \in \mathbb{N} \quad \forall n > N(\varepsilon) \Rightarrow x_n \in O_\varepsilon(a).$$

### Задание 3

Таблица Брадиса для синуса и косинуса

sin	0'	6'	12'	18'	24'	30'	36'	42'	48'	54'	60'	cos	1'	2'	3'
											0.0000	90°			
0°	0.0000	0017	0035	0052	0070	0087	0105	0122	0140	0157	0175	89°	3	6	9
1°	0175	0192	0209	0227	0244	0262	0279	0297	0314	0332	0349	88°	3	6	9
2°	0349	0366	0384	0401	0419	0436	0454	0471	0488	0506	0523	87°	3	6	9
3°	0523	0541	0558	0576	0593	0610	0628	0645	0663	0680	0698	86°	3	6	9
4°	0698	0715	0732	0750	0767	0785	0802	0819	0837	0854	0.0872	85°	3	6	9

### Вариант 2

#### Задание 1

**Определение 2.3.** Пусть  $X$  и  $Y$  – некоторые множества. Тогда *пересечением* множеств  $X$  и  $Y$  называется множество

$$X \cap Y = \{x \mid x \in X \text{ и } x \in Y\},$$

*объединением* множеств  $X$  и  $Y$  называется множество

$$X \cup Y = \{x \mid x \in X \text{ или } x \in Y\}.$$

Например, если  $X = \{2, 4, 6, 8, 9\}$ ,  $Y = \{1, 3, 6, 9\}$ , то  $X \cap Y = \{6, 9\}$ , а  $X \cup Y = \{1, 2, 3, 4, 6, 8, 9\}$ .

#### Задание 2

**Пример 6.1.** Доказать, что

$$\lim_{n \rightarrow \infty} \frac{n^3 + 7n + 3}{2n^3 + 5n + 4} = \frac{1}{2}.$$

**Решение.** По определению предела последовательности надо по любому числу  $\varepsilon > 0$  найти такое число  $N$ , чтобы для любых  $n > N$  выполнялось неравенство

$$\left| \frac{n^3 + 7n + 3}{2n^3 + 5n + 4} - \frac{1}{2} \right| < \varepsilon.$$

Преобразуем его левую часть:

$$\begin{aligned} \left| \frac{n^3 + 7n + 3}{2n^3 + 5n + 4} - \frac{1}{2} \right| &= \left| \frac{2n^3 + 14n + 6 - 2n^3 - 5n - 4}{4n^3 + 10n + 8} \right| = \\ &= \frac{9n + 2}{4n^3 + 10n + 8}. \end{aligned}$$

### Задание 3

Таблица Брадиса для тангенса и котангенса

tg	0'	6'	12'	18'	24'	30'	36'	42'	48'	54'	60'	ctg	1'	2'	3'
											0	90°			
0°	0,000	0017	0035	0052	0070	0087	0105	0122	0140	0157	0175	89°	3	6	9
1°	0175	0192	0209	0227	0244	0262	0279	0297	0314	0332	0349	88°	3	6	9
2°	0349	0367	0384	0402	0419	0437	0454	0472	0489	0507	0524	87°	3	6	9
3°	0524	0542	0559	0577	0594	0612	0629	0647	0664	0682	0699	86°	3	6	9
4°	0699	0717	0734	0752	0769	0787	0805	0822	0840	0857	0,0875	85°	3	6	9

### Вариант 3

#### Задание 1

**Определение 2.5.** Пусть  $\mathcal{E}$  – универсальное множество для данной системы множеств,  $X$  – некоторое множество этой системы. *Дополнением* множества  $X$  называется множество  $C_{\mathcal{E}}X$  (или просто  $CX$ ) тех элементов универсального множества, которые не принадлежат  $X$ .

Например, если  $X = \{1, 2, 4, 6, 7, 8\}$ , а универсальное множество – множество всех десятичных цифр, то  $CX = \{0, 3, 5, 9\}$ .

Справедливы равенства:

- 1)  $X \cap \mathcal{E} = X$ ;
- 2)  $X \cup \mathcal{E} = \mathcal{E}$ ;
- 3)  $C\mathcal{E} = \emptyset$ ;
- 4)  $C\emptyset = \mathcal{E}$ .

#### Задание 2

**Свойство 1.** Если последовательность  $\{x_n\}$  сходится, то ее предел единственный.

Доказательство проведем от противного. Пусть

$$\lim_{n \rightarrow \infty} x_n = a < b = \lim_{n \rightarrow \infty} x_n.$$

Возьмем  $\varepsilon = \frac{b-a}{3} > 0$ , тогда  $O_{\varepsilon}(a) \cap O_{\varepsilon}(b) = \emptyset$  в силу выбора  $\varepsilon$ . По определению сходимости, для выбранного  $\varepsilon$

$$\exists N_1 : \forall n > N_1 \quad x_n \in O_{\varepsilon}(a),$$

$$\exists N_2 : \forall n > N_2 \quad x_n \in O_{\varepsilon}(b).$$

Следовательно, для  $n > N_1 + N_2$   $x_n \in O_{\varepsilon}(a) \cap O_{\varepsilon}(b)$ , что означает непустоту этого пересечения. Получено противоречие.  $\square$

### Задание 3

Таблица синусов углов от  $0^\circ$  до  $180^\circ$

$\sin(0^\circ) = 0$	$\sin(46^\circ) = 0.71934$	$\sin(91^\circ) = 0.999848$	$\sin(136^\circ) = 0.694658$
$\sin(1^\circ) = 0.017452$	$\sin(47^\circ) = 0.731354$	$\sin(92^\circ) = 0.999391$	$\sin(137^\circ) = 0.681998$
$\sin(2^\circ) = 0.034899$	$\sin(48^\circ) = 0.743145$	$\sin(93^\circ) = 0.99863$	$\sin(138^\circ) = 0.669131$
$\sin(3^\circ) = 0.052336$	$\sin(49^\circ) = 0.75471$	$\sin(94^\circ) = 0.997564$	$\sin(139^\circ) = 0.656059$
$\sin(4^\circ) = 0.069756$	$\sin(50^\circ) = 0.766044$	$\sin(95^\circ) = 0.996195$	$\sin(140^\circ) = 0.642788$
$\sin(5^\circ) = 0.087156$	$\sin(51^\circ) = 0.777146$	$\sin(96^\circ) = 0.994522$	$\sin(141^\circ) = 0.62932$
$\sin(6^\circ) = 0.104528$	$\sin(52^\circ) = 0.788011$	$\sin(97^\circ) = 0.992546$	$\sin(142^\circ) = 0.615661$
$\sin(7^\circ) = 0.121869$	$\sin(53^\circ) = 0.798636$	$\sin(98^\circ) = 0.990268$	$\sin(143^\circ) = 0.601815$

### Вариант 4

#### Задание 1

**Определение 2.6.** Пусть  $X$  и  $Y$  – некоторые множества. Разностью множеств  $X$  и  $Y$  называется множество  $X \setminus Y = \{x \mid x \in X \text{ и } x \notin Y\}$ .

Например, если  $X = \{1, 3, 7, 8, 9\}$ , а  $Y = \{5, 6, 7, 8\}$ , то  $X \setminus Y = \{1, 3, 9\}$ .

Разность легко выражается через введенные ранее операции. Действительно, нетрудно показать, что

$$X \setminus Y = X \cap CY.$$

#### Задание 2

**Свойство 3.** Если  $x_n \leq y_n \leq z_n$  для всех  $n$  и

$$\lim_{n \rightarrow \infty} x_n = \lim_{n \rightarrow \infty} z_n = a, \quad \text{то} \quad \lim_{n \rightarrow \infty} y_n = a.$$

Доказательство. Возьмем любое  $\varepsilon > 0$ . Тогда из условия  $\lim_{n \rightarrow \infty} x_n = a$  следует, что

$$\exists N_1 \forall n > N_1 \quad a - \varepsilon < x_n < a + \varepsilon,$$

а из условия  $\lim_{n \rightarrow \infty} z_n = a$  следует, что

$$\exists N_2 \forall n > N_2 \quad a - \varepsilon < z_n < a + \varepsilon.$$

Следовательно, при всех  $n > N_1 + N_2$

$$a - \varepsilon < x_n \leq y_n \leq z_n < a + \varepsilon,$$

## Задание 3

Таблица IX. ТАНГЕНСЫ

A	0'	6'	12'	18'	24'	30'	36'	42'	48'	54'	60'		1'	2'	3'	
0°	0,0000	0017	0035	0052	0070	0087	0105	0122	0140	0157	0175	0,0000	90°			
1°	0175	0192	0209	0227	0244	0262	0279	0297	0314	0332	0349	89°	3	6	9	
2°	0349	0367	0384	0402	0419	0437	0454	0472	0489	0507	0524	88°	3	6	9	
3°	0524	0542	0559	0577	0594	0612	0629	0647	0664	0682	0699	87°	3	6	9	
4°	0699	0717	0734	0752	0769	0787	0805	0822	0840	0857	0,0875	85°	3	6	9	
5°	0,0875	0892	0910	0928	0945	0963	0981	0998	1016	1033	1051	84°	3	6	9	
6°	1051	1069	1086	1104	1122	1139	1157	1175	1192	1210	1228	83°	3	6	9	
7°	1228	1246	1263	1281	1299	1317	1334	1352	1370	1388	1405	82°	3	6	9	
8°	1405	1423	1441	1459	1477	1495	1512	1530	1548	1566	1584	81°	3	6	9	
9°	1584	1602	1620	1638	1655	1673	1691	1709	1727	1745	0,1763	80°	3	6	9	

## Вариант 5

## Задание 1

**Определение 2.8.** Пусть  $X, Y, Z$  – некоторые множества. *Прямым (декартовым) произведением*  $X \times Y \times Z$  называется множество всевозможных упорядоченных троек, первая компонента которых принадлежит  $X$ , вторая –  $Y$ , третья –  $Z$ , т. е.

$$X \times Y \times Z = \{(x, y, z) | x \in X, y \in Y, z \in Z\}.$$

Прямое произведение одинаковых множеств  $X \times X \times X$  обозначают  $X^3$ .

## Задание 2

**Свойство 4.** Если последовательности  $\{x_n\}$  и  $\{y_n\}$  сходятся, то сходится последовательность  $\{x_n + y_n\}$  и

$$\lim_{n \rightarrow \infty} (x_n + y_n) = \lim_{n \rightarrow \infty} x_n + \lim_{n \rightarrow \infty} y_n.$$

**Доказательство.** Пусть  $\lim_{n \rightarrow \infty} x_n = a$ ,  $\lim_{n \rightarrow \infty} y_n = b$ . Возьмем любое  $\varepsilon > 0$ . Тогда из первого условия следует, что

$$\exists N_1 \quad \forall n > N_1 \quad |x_n - a| < \frac{\varepsilon}{2},$$

а из второго:

$$\exists N_2 \quad \forall n > N_2 \quad |y_n - b| < \frac{\varepsilon}{2}.$$

### Задание 3

Таблица VIII. СИНОУСЫ

$A$	$0'$	$6'$	$12'$	$18'$	$24'$	$30'$	$36'$	$42'$	$48'$	$54'$	$60'$		$1'$	$2'$	$3'$
$70^\circ$	0,9397	9403	9409	9415	9421	9426	9432	9438	9444	9449	0,9455	$19^\circ$	1	2	3
$71^\circ$	9455	9461	9466	9472	9478	9483	9489	9494	9500	9505	9511	$18^\circ$	1	2	3
$72^\circ$	9511	9516	9521	9527	9532	9537	9542	9548	9553	9558	9563	$17^\circ$	1	2	3
$73^\circ$	9563	9568	9573	9578	9583	9588	9593	9598	9603	9608	9613	$16^\circ$	1	2	2
$74^\circ$	9613	9617	9622	9627	9632	9636	9641	9646	9650	9655	0,9659	$15^\circ$	1	2	2

### Вариант 6

#### Задание 1

##### I. АКСИОМЫ СЛОЖЕНИЯ

1. *Существование нейтрального элемента:*

$$\exists 0 \in X \quad \forall x \in X \quad x + 0 = x.$$

2. *Существование обратного элемента:*

$$\forall x \in X \quad \exists x' \in X \quad x + x' = 0.$$

3. *Ассоциативность сложения:*

$$\forall x, y, z \in X \quad (x + y) + z = x + (y + z).$$

4. *Коммутативность сложения:*

$$\forall x, y \in X \quad x + y = y + x.$$

Множество  $X$ , удовлетворяющее условиям  $I_1$ – $I_3$ , называется *группой* относительно операции сложения (*аддитивной группой*).

Множество  $X$ , удовлетворяющее условиям  $I_1$ – $I_4$ , называется *коммутативной (абелевой) группой* относительно операции сложения.

#### Задание 2

**Определение 6.4.** Последовательность  $\{x_n\}$  называется *бесконечно малой*, если  $\lim_{n \rightarrow \infty} x_n = 0$ .

Развернутое определение:

$$\forall \varepsilon > 0 \quad \exists N(\varepsilon) \in \mathbb{N} \quad \forall n > N(\varepsilon) \Rightarrow |x_n| < \varepsilon.$$

**Свойство 1.** Сумма бесконечно малых последовательностей есть бесконечно малая последовательность.

**Доказательство.** Пусть  $\{x_n\}$  и  $\{y_n\}$  – бесконечно малые последовательности. Возьмем произвольное  $\varepsilon > 0$ . Для него

$$\begin{aligned} \exists N_1 \quad \forall n > N_1 \quad |x_n| < \frac{\varepsilon}{2}, \\ \exists N_2 \quad \forall n > N_2 \quad |y_n| < \frac{\varepsilon}{2}. \end{aligned}$$

### Задание 3

Таблица IX. ТАНГЕНСЫ

$\alpha$	0'	6'	12'	18'	24'	30'	36'	42'	48'	54'	60'		1'	2'	3'
40°	0,8391	8421	8451	8481	8511	8541	8571	8601	8632	8662	0,8693	49°	5	10	15
41°	8693	8724	8754	8785	8816	8847	8878	8910	8941	8972	9004	48°	5	10	16
42°	9004	9036	9067	9099	9131	9163	9195	9228	9260	9293	9325	47°	6	11	16
43°	9325	9358	9391	9424	9457	9490	9523	9556	9590	9623	0,9657	46°	6	11	17
44°	9657	9691	9725	9759	9793	9827	9861	9896	9930	9965	1,0000	45°	6	11	17

### Вариант 7

#### Задание 1

##### II. АКСИОМЫ УМНОЖЕНИЯ

1. *Существование нейтрального элемента:*

$$\exists 1 \in X \quad \forall x \in X \quad x \cdot 1 = x.$$

2. *Существование обратного элемента:*

$$\forall x \in X \setminus \{0\} \quad \exists x' \in X \quad x \cdot x' = 1.$$

3. *Ассоциативность умножения:*

$$\forall x, y, z \in X \quad (x \cdot y) \cdot z = x \cdot (y \cdot z).$$

4. *Коммутативность умножения:*

$$\forall x, y \in X \quad x \cdot y = y \cdot x.$$

Множество  $X$ , удовлетворяющее условиям II<sub>1</sub>–II<sub>3</sub>, называется *группой* относительно операции умножения (*мультипликативной группой*).

Множество  $X$ , удовлетворяющее условиям II<sub>1</sub>–II<sub>4</sub>, называется *коммутативной (абелевой) группой* относительно операции умножения.

#### Задание 2

**Определение 6.5.** Последовательность  $\{x_n\}$  называется *бесконечно большой*, если

$$\forall E > 0 \quad \exists N(E) \in \mathbb{N} \quad \forall n > N(E) \Rightarrow |x_n| > E.$$

Этот факт записывается так:

$$\lim_{n \rightarrow \infty} x_n = \infty.$$

**Свойство 3.** Последовательность  $\left\{\frac{1}{x_n}\right\}$ , обратная к бесконечно большой последовательности  $\{x_n\}$ , есть бесконечно малая последовательность.

## Задание 3

Таблица X. ТАНГЕНСЫ УГЛОВ, БЛИЗКИХ К 90°

A	0'	1'	2'	3'	4'	5'	6'	7'	8'	9'	10'	
76°00'	4,011	4,016	4,021	4,026	4,031	4,036	4,041	4,046	4,051	4,056	4,061	50'
10'	4,061	4,066	4,071	4,076	4,082	4,087	4,092	4,097	4,102	4,107	4,113	40'
20'	4,113	4,118	4,123	4,128	4,134	4,139	4,144	4,149	4,155	4,160	4,165	30'
30'	4,165	4,171	4,176	4,181	4,187	4,192	4,198	4,203	4,208	4,214	4,219	20'
40'	4,219	4,225	4,230	4,236	4,241	4,247	4,252	4,258	4,264	4,269	4,275	10'
50'	4,275	4,280	4,286	4,292	4,297	4,303	4,309	4,314	4,320	4,326	4,331	13°00'

## Вариант 8

## Задание 1

III. СВЯЗЬ МЕЖДУ СЛОЖЕНИЕМ И УМНОЖЕНИЕМ

$$\forall x, y, z \in X \quad (x + y) \cdot z = x \cdot z + y \cdot z.$$

Аксиома III носит название *дистрибутивность* или *распределительный закон*.

Множество  $X$ , удовлетворяющее аксиомам I–III, называется *полем*.

IV. АКСИОМЫ ПОРЯДКА

1.  $\forall x, y \in X \Rightarrow x \leq y$  или  $y \leq x$ .

2. *Рефлексивность*:

$$\forall x \in X \quad x \leq x.$$

## Задание 2

**Свойство 4.** Пусть  $\{x_n\}$  – бесконечно малая последовательность и такая, что  $x_n \neq 0$  при  $n > n_0$ . Тогда последовательность  $\left\{\frac{1}{x_n}\right\}$ , обратная к  $\{x_n\}$ , есть бесконечно большая последовательность.

*Доказательство.* Положим  $y_n = \frac{1}{x_n}$  при  $n > n_0$ . Возьмем  $E > 0$  и рассмотрим  $\varepsilon = \frac{1}{E} > 0$ . Для него по определению бесконечно малой последовательности найдется номер  $N(\varepsilon)$  такой, что

$$|x_n| < \varepsilon, \quad \text{при } n > N(\varepsilon).$$

## Задание 3

Таблица X. ТАНГЕНСЫ УГЛОВ, БЛИЗКИХ К 90°

A	0'	1'	2'	3'	4'	5'	6'	7'	8'	9'	10'	
83°00'	8,144	8,164	8,184	8,204	8,223	8,243	8,264	8,284	8,304	8,324	8,345	50'
10'	8,345	8,366	8,386	8,407	8,428	8,449	8,470	8,491	8,513	8,534	8,556	40'
20'	8,556	8,577	8,599	8,621	8,643	8,665	8,687	8,709	8,732	8,754	8,777	30'
30'	8,777	8,800	8,823	8,846	8,869	8,892	8,915	8,939	8,962	8,986	9,010	20'
40'	9,010	9,034	9,058	9,082	9,106	9,131	9,156	9,180	9,205	9,230	9,255	10'
50'	9,255	9,281	9,306	9,332	9,357	9,383	9,409	9,435	9,461	9,488	9,514	6°00'

## Вариант 9

### Задание 1

**V. СВЯЗЬ МЕЖДУ СЛОЖЕНИЕМ И ОТНОШЕНИЕМ ПОРЯДКА**

$$\text{Если } x \leq y, \text{ то } \forall z \in X \Rightarrow x + z \leq y + z.$$

**VI. СВЯЗЬ МЕЖДУ УМНОЖЕНИЕМ И ОТНОШЕНИЕМ ПОРЯДКА**

$$\text{Если } x \leq y, \text{ то } \forall z \geq 0 \Rightarrow x \cdot z \leq y \cdot z.$$

**VII. АКСИОМА НЕПРЕРЫВНОСТИ**

Пусть  $A \subset X$ ,  $B \subset X$ ,  $A \neq \emptyset$ ,  $B \neq \emptyset$  и для любых элементов  $a \in A$  и  $b \in B$  выполняется  $a \leq b$ . Тогда найдется такой элемент  $c \in X$ , что  $a \leq c \leq b$  при любых  $a \in A$  и  $b \in B$ .

Множество  $X$ , удовлетворяющее аксиомам **I–VII** и содержащее более одного элемента, называется *множеством действительных (вещественных) чисел*. Его принято обозначать  $\mathbb{R}$ .

### Задание 2

**Определение 7.6.** Пусть функция  $f$  определена на интервале  $(a, x_0)$ . Число  $A$  называется *пределом функции  $f$  слева в точке  $x_0$* ,

$$A = \lim_{x \rightarrow x_0 - 0} f(x),$$

если

$$\forall \varepsilon > 0 \quad \exists \delta > 0 \quad \forall x \in (a, x_0) \\ (x_0 - \delta < x < x_0) \Rightarrow (|f(x) - A| < \varepsilon).$$

*Предел функции  $f$  справа* определяется аналогично.

### Задание 3

**Т а б л и ц а XI. РАДИАННАЯ МЕРА**  $\left( \text{Арг } A^\circ = \frac{\pi A}{180^\circ} \right)$

$A$	0'	6'	12'	18'	24'	30'	36'	42'	48'	54'	1'	2'	3'
0°	0,0000	0017	0035	0052	0070	0087	0105	0122	0140	0157	3	6	9
1°	0,0175	0192	0209	0227	0244	0262	0279	0297	0314	0332	3	6	9
2°	0,0349	0367	0384	0401	0419	0436	0454	0471	0489	0506	3	6	9
3°	0,0524	0541	0559	0576	0593	0611	0628	0646	0663	0681	3	6	9
4°	0,0698	0716	0733	0750	0768	0785	0803	0820	0838	0855	3	6	9

## Вариант 10

### Задание 1

**Определение 5.3'.**  $M = \sup X$ , если

- 1)  $\forall x \in X \quad x \leq M$ ;
- 2)  $\forall \varepsilon > 0 \exists x_\varepsilon \in X \quad x_\varepsilon > M - \varepsilon$ .

**Определение 5.4.** Множество  $X$  называется *ограниченным*, если оно ограничено сверху и снизу, т. е.

$$\exists a, b \in \mathbb{R} \quad \forall x \in X \Rightarrow a \leq x \leq b.$$

Множество называется *неограниченным*, если оно неограничено хотя бы с одной стороны (сверху или снизу).

### Задание 2

#### Арифметические свойства предела функции

Пусть функции  $f$  и  $g$  определены на интервале  $(a, b)$ , кроме, быть может, точки  $x_0$ . Если существуют пределы  $\lim_{x \rightarrow x_0} f(x)$ ,  $\lim_{x \rightarrow x_0} g(x)$ , то существуют пределы суммы, произведения и отношения этих функций и имеют место равенства:

- 1)  $\lim_{x \rightarrow x_0} [f(x) + g(x)] = \lim_{x \rightarrow x_0} f(x) + \lim_{x \rightarrow x_0} g(x)$ ;
- 2)  $\lim_{x \rightarrow x_0} [f(x) \cdot g(x)] = (\lim_{x \rightarrow x_0} f(x)) \cdot (\lim_{x \rightarrow x_0} g(x))$ ;
- 3)  $\lim_{x \rightarrow x_0} \frac{f(x)}{g(x)} = \frac{\lim_{x \rightarrow x_0} f(x)}{\lim_{x \rightarrow x_0} g(x)}$ , при условии  $\lim_{x \rightarrow x_0} g(x) \neq 0$ .

### Задание 3

Таблица XI. РАДИАННАЯ МЕРА  $\left( \text{Arc } A^\circ = \frac{\pi A}{180^\circ} \right)$

$A$	$0'$	$6'$	$12'$	$18'$	$24'$	$30'$	$36'$	$42'$	$48'$	$54'$	$1'$	$2'$	$3'$
$35^\circ$	0,6109	6126	6144	6161	6178	6196	6213	6231	6248	6266	3	6	9
$36^\circ$	0,6283	6301	6318	6336	6353	6370	6388	6405	6423	6440	3	6	9
$37^\circ$	0,6458	6475	6493	6510	6528	6545	6562	6580	6597	6615	3	6	9
$38^\circ$	0,6632	6650	6667	6685	6702	6720	6737	6754	6772	6789	3	6	9
$39^\circ$	0,6807	6824	6842	6859	6877	6894	6912	6929	6946	6964	3	6	9

## Лабораторная работа 8

### Презентации LaTeX

**Цель:** научиться создавать презентации LaTeX, использовать в них такие возможности, как: применение тем, структурирование информации на слайдах, вставка рисунков и анимации, готовых звуковых файлов, создание оглавления, использование эффектов переходов между слайдами и для отдельных объектов на слайде, гиперссылок и кнопок для перехода между слайдами и прочее.

Создание презентации в LaTeX, с использованием пакета **beamer** необходимо, как и для любого другого документа начать с преамбулы.

#### Теоретические сведения

Пакет **beamer** определяет класс документов, который переопределяет формат страницы (ширина страницы – 12,8 см, высота – 9,6 см, при соотношении сторон 4:3). Пакет можно подключить командой **`\documentclass{beamer}`**.

#### Структура презентации

Презентация в LaTeX состоит из набора фреймов, каждый из которых может иметь один слайд или набор слайдов (слоев фрейма). Каждый слой имеет свой порядковый номер (начиная с 1).

#### Синтаксис окружения для создания фрейма

```
\begin{frame}<слои> [<слои по умолчанию>] [параметры]
    {заголовок}{подзаголовок}
```

#### Содержание фрейма

```
\end{frame}
```

В данном окружении параметр **<слои>** показывает, какие слои должны отображаться на фрейме. Если параметр слои заключен в угловые скобки, то это означает - **<слои по умолчанию>**, иначе это аргумент [**параметры**].

Параметр «**заголовок**» выполняет ту же функцию, как и команда `\frametitle{заголовок}`, а именно вывод названия слайда. Подзаголовок выводится командой `\framesubtitle{подзаголовок}`.

Назначение <параметр> аналогичны `allowdisplaybreaks` и принимает значения от 0 до 4. При этом 0 означает, что длинные формулы не будут переноситься на другой слайд, а 4 означает противоположное, что такие формулы будут перенесены на следующий слайд фрейма.

При этом разбиение на слайды можно выполнить с помощью одной из команд: `\break`, `\nobreak`, `\pagebreak`, `noimagebreak`.

Рассмотрим ряд команд для работы со слайдами:

**label** – определяет метку слайда, на которую можно ссылаться в программном коде;

**t, c, b** – вертикальное выравнивание слайда (по верхнему краю, по центру, по нижнему краю);

**plain** – используется для отображения больших объектов на слайде, отключая колонтитулы, область заголовка и другие отступы.

Слайд имеет следующую структуру:

- верхний и нижний колонтитулы;
- левое и правое поле;
- панель навигации;
- управляющие знаки;
- логотипы;
- название слайда;
- фон и др. компоненты.

### Отступы

Для управления расположением данных на слайде предназначены отступы, значения которых можно установить с помощью команды `\setbeamersize{отступы}`. Отступы задаются парами **ключ=значение**.

**Поддерживаемые отступы:**

**text margin left** – величина левого отступа (от левого поля или от левого края), по умолчанию – 1 см;

**text margin right** – величина правого отступа;

**sidebar width left** – ширина левого поля;

**sidebar width right** – ширина правого поля;

**Пример:**

```
\setbeamersize{text margin left=3cm, sidebar width left=2cm}
```

Устанавливает отступ слева – 2 см, ширину левого поля – 1 см.

**Титульный слайд**

Титульный слайд можно создать автоматически. Для этого существует команда **\titlepage**. Вывести титульный слайд можно двумя способами:

1. **\frame{\titlepage}**

2. **\begin{frame}**

**\titlepage**

**\end{frame}**

Для создания титульного слайда можно использовать команды:

**\title[короткий заголовок]{заголовок}** – создание названия презентации (короткий заголовок отображается в колонтитулах, «заголовок» можно разделить на несколько строк командой **\\**).

**\subtitle[короткий заголовок]{заголовок}** – создает подзаголовок.

**\author[сокращ. авторы]{авторы}** – задается список авторов, который разделяется командой **\and**. Если авторы из разных организаций, то после каждого имени автора ставится команда **\inst** с параметром.

**Пример:**

**\institute{\inst{1}Нижегородский государственный архитектурно-строительный университет \and**

**\inst{2} Нижегородский государственный университет им.~Н.И. ~Лобачевского};**

**\date[короткая дата]{дата}** – хранит дату мероприятия, если в параметре «дата» указать команду **\today**, то отобразится дата компиляции документа.

`\titlegraphic{текст}` – будет выводиться «текст» в качестве логотипа и чаще всего к этой команде подключается картинка.

**Пример** создания заглавного слайда показан на рисунке 21.

```
\documentclass{beamer}
\usepackage[english,russian]{babel}
\usepackage[utf8]{inputenc}
% Стиль презентации
\usetheme{Warsaw}
\begin{document}
\title{Издательская система TeXstudio}
\author{Создание презентаций в TeXstudio}
\institute{Нижегородский государственный архитектурно-строительный
университет}
\date{Нижний Новгород, 2023}
% Создание заглавной страницы
\frame{\titlepage}
% Автоматическая генерация содержания
\frame{\frametitle{Содержание}\tableofcontents}
\end{document}
```

Рисунок 21. – Создание заглавного слайда

**Результат** работы программного кода показан на рисунке 22.

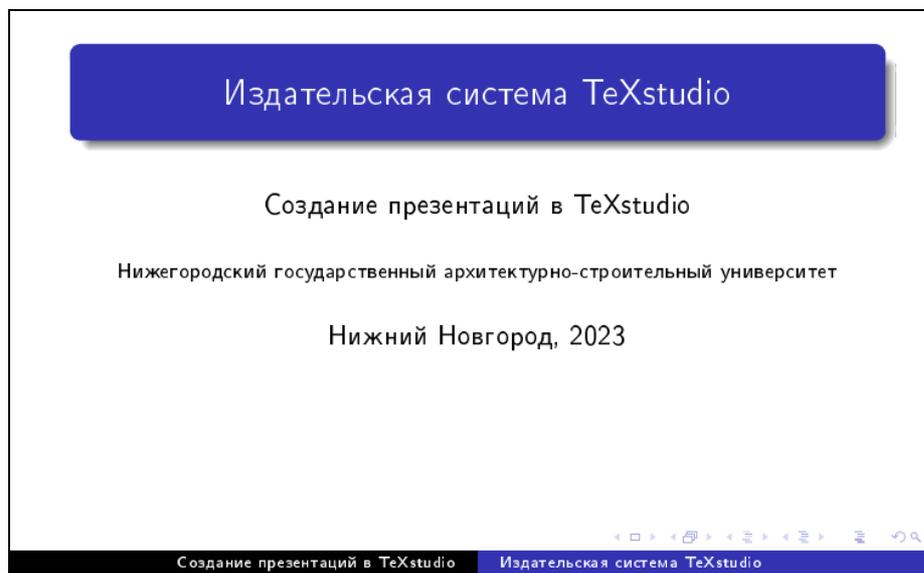


Рисунок 22. – Пример создания заглавного слайда

Вывод содержания осуществляется командой `\tableofcontents[список параметров]`.

Создание заглавной страницы слайда можно организовать с помощью переменных окружения `\frame{}` или же конструкцией:

```
\begin {frame}
```

.....

**\end {frame}**

## Разделы презентации

Презентацию можно разделить на логически завершённые части с помощью команд: **\part(часть)**, **\chapter(глава)**, **\section(раздел)** и пр. А для создания списка лекций можно ввести команду **\lecture[короткое название]{название лекции}{метка}**. Текст параметра «название лекции» выводится с помощью команды **\insertlecture**, а параметра «короткое название» с помощью команды **\insertshortlecture**.

Если в преамбуле указать команду **\includeonlylecture{метка}**, то в презентацию будут включены только лекции с заданной меткой.

## Слои

Каждый слой фрейма имеет свой порядковый номер. Для размещения информации на соответствующем слое нужно указать его номер. Для создания слоёв в презентации нужно использовать команду **\pause[номер слайда]**. В этом случае на фрейме будет создано ряд слайдов. Параметр «номер слайда» указывает с какого слайда будет отображаться текст, следующий за командой **\pause**. На первом слайде будет отображаться вся информация, которая идет до команды **\pause**, на втором тоже та информация, которая идет до команды **\pause** и т.д. пока не встретится команда **\onslide**.

Все часто используемые команды и окружения можно переопределить, добавив к ним параметр в угловых скобках, например: **\textbf<слайды>{текст}**. В этом случае в угловых скобках указываются номера слайдов, для которых будет действовать эта команда.

**Пример:** **\textbf<1,3-5>{текст}** при этом текст на слайдах 1, 3, 4 и 5 будет выводиться полужирным начертанием.

Команда **\only<слайды>{текст}** выводит текст только на слайдах с заданным номером.

## Внешний вид презентации

Для оформления внешнего вида презентации используются темы. Рассматривается пять тем: основная, внутренняя, внешняя, цветовая и темы шрифтов.

**Основная тема** – определяет внешний вид всех базовых элементов презентации (формулы, нумерация параграфов и пр.).

**Внутренняя тема** – создает отображение элементов в рабочей области презентации (не входят колонтитулы, напели навигации и пр.).

**Внешняя тема** – формирует вид внешних элементов (колонтитулов, полей и пр.).

**Цветовая схема** – задает цветовую гамму презентации.

**Тема шрифтов** – задает используемые шрифты и их атрибуты.

**Тема** для оформления презентации задается при помощи `\usetheme`, а в скобках указывается название самой темы. Синтаксис команды:

```
\usetheme[опции]{тема}
```

Рассмотрим основные темы.

`\usetheme{default}` – наиболее простая тема, которая устанавливается по умолчанию (без колонтитулов, названий слайдов и др. элементы).

`\usetheme[headheight=<высота>, footheight=< высота >]{boxes}` – определяет высоту колонтитулов.

`\usetheme[опции]{Bergen}` – используется для перечислений, но не очень хорошо отображает колонки и отступы справа.

`\usetheme[опции]{Boadilla}` – позволяет отобразить максимум информации на фиксированном пространстве. Параметр **опции**, содержащий **secheader** позволяет добавить верхний колонтитул, содержащий название раздела и подраздела.

`\usetheme{Pittsburgh}` – тема отображается в спокойных тонах, заголовки слайдов выровнены по правому краю.

`\usetheme[опции]{Rochester }` – основная тема без навигации. Параметр **опции** может иметь значение `height=<высота>`, который задает высоту заголовка слайдов.

`\usetheme{Antibes}` – главная тема с навигацией в виде дерева. Её можно изменить с помощью другой цветовой схемой. При этом в верхнем колонтитуле отображается дерево разделов документа, в котором выводятся название документа, текущего раздела и подраздела.

`\usetheme[опции]{Berkeley }` – основная тема с оглавлением на полях. Отображается на левом поле, подсвечивается текущий пункт, можно вывести логотип, имеет фиксированную высоту заголовка слайда.

Для настройки внутренних элементов презентации используются внутренние и внешние темы. Внутренние темы управляют внешним видом следующих элементов: титульной страницы, списков, блоков, теорем и доказательств, рисунков и таблиц, сносок и библиографии. Для внутренних тем используется команда `\useinnertheme[опции]{тема}`.

Тема `{default}` используется по умолчанию. Элементы отображаются как в обычном тексте, за исключением нумерованных списков. В них вместо кружочков используются треугольники.

**Внешние темы** определяют общую разметку слайда, колонтитулов, левого и правого полей, логотипа и названия слайда. Для внешних тем используется команда `\useoutertheme[опции]{тема}`.

Тема `{default}` используется по умолчанию. Выравнивает заголовок по левому краю, отключает колонтитулы.

Тема `{infolines}` задает верхний колонтитул, содержащий название текущей секции и подсекции. В верхнем колонтитуле выводятся автор, организация, название презентации, дата и номер слайда.

**Цветовая схема** позволяет настроить цвет для любого элемента презентации. Изменить цветовую гамму можно командой `\usecolortheme[опции]{тема}`.

Тема `{default}` используемая по умолчанию, использует минимум цветов.

Тема `{sidebarfab}` текущий раздел в оглавлении на полях выделяется цветом.

Тема `{albatross}` использует основные цвета схемы синий и желтый.

Тема `{beetle}` использует черный текст на сером фоне и белый текст для заголовков.

### Шрифты

Тему со шрифтами можно подключить командой `\usefonttheme[опции]{тема}`.

Тема по умолчанию `{default}` устанавливает шрифт **Sans Serif** для всех элементов презентации.

Тема `{structurebold}` выводит названия и текст в колонтитулах полужирным шрифтом. Опция может принимать значение `{onlysmall}` устанавливает полужирный шрифт только для колонтитулов и полей, а `{onlylarge}` устанавливает полужирный шрифт для названий.

Размером шрифта во всем документе можно управлять с помощью параметров класса `beamer`. Например, команда `\documentclass[8pt]{beamer}` устанавливает базовый размер шрифта 8 пунктов, а остальные рассчитываются на его основе. Поддерживаются 9pt, 10pt, 11pt, 12pt, 14pt, 17pt, 20pt.

### Графические и мультимедийные возможности

Для вставки в презентацию картинок из внешних файлов можно, используя пакеты **graphic** или **pgt**. Возможности этих пакетов аналогичны, но **graphic** дает возможность экономить память при использовании одного и того же рисунка в нескольких местах, а **pgt** позволяет работать с полупрозрачными изображениями.

Документ формата **pdf** может использовать только графические файлы типа: **.pdf, .jpg, .jpeg, .png**.

Добавить рисунки в презентацию можно следующими способами:

Командой `\includegraphics [список ключей]{имя файла}` из пакета **graphics** или командами `\pgfuseimage{имя файла}`, или командой `\pgfimage{имя файла}` из пакета **pgf**.

Например, для размещения рисунка достаточно добавить следующий код (Рисунок 23).

```
\begin{frame}{Домашние животные}
  \begin{center}
    \includegraphics{Cat.jpg}
  \end{center}
\end{frame}
```

Рисунок 23. – Размещение рисунка

С помощью пакета **beamer** можно **включить в презентацию анимацию**, созданную сторонней программой. Для этого используется пакет `multimedia`. Добавить видеофайл можно командой `\movie[опции]{подпись}{имя файла}`. Где параметр «подпись» может быть обычным текстом или картинкой. Осуществляется сначала их показ, а затем на этом месте перекрывается анимацией. Параметр «опции» может быть представлен комбинацией следующий значений:

**autostart** – запускается анимация сразу при показе слайда;

**borderwidth=<размер>** устанавливает ширину рамки вокруг анимации.

**duration=<время>** устанавливает время показа анимации в секундах.

**height=<высота>** устанавливает высоту области анимации (включая подпись).

В презентацию можно так же включить звуковой ролик с помощью команды: `\sound[опции]{подпись}{имя файла}`.

Параметр «подпись» аналогичен такому же параметру в команду `\movie`. Параметр «опции» может иметь следующие значения:

**autostart** – начинает проигрывать звук как только отобразится соответствующий слайд.

**automute** – звук останавливается как только будет пролистан текущий слайд.

**bitspersample=<8 или 16>** указывает качество звука (по умолчанию 16).

**channels=<1 или 2>** определяет моно или стереозвучание (по умолчанию 1 – моно).

С целью добавления в презентацию **дополнительных эффектов** можно использовать плавное переключение между слайдами. Для этого можно использовать такие команды, как:

**\transdissolve<слайды>[параметры]** – применяется эффект исчезновения слайда и перекрытия его следующим.

Команды визуальных эффектов относятся к следующему слайду и могут быть указаны в любом месте данного слайда.

### Пример:

#### 1-й вариант

```
\begin{frame}
```

Первый слайд

```
\end{frame}
```

```
\begin{frame}
```

```
\transdissolve
```

Второй слайд

```
\end{frame}
```

#### 2-й вариант

```
\begin{frame}
```

```
\only<1>{Первый слайд}
```

```
\only<2>{Второй слайд}
```

```
\transdissolve<2>
```

```
\end{frame}
```

В обоих случаях первый слайд будет плавно сменяться вторым. Во втором случае опция *<слайды>* показывает на каком по счету слайде будет выведена данная команда. «*Параметр*» может управлять эффектом переключения слайдов парой «*ключ=значение*». «*Значения*» могут быть следующие:

**duration** – длительность перехода между слайдами в секундах (по умолчанию 1 секунда).

**direction** – направление движения эффекта в градусах. Поддерживаются значения: 0, 90, 180, 270 и 315 для эффекта мерцания (по умолчанию значение равно 0).

Команды

**\transblindshorizontal**<слайды>[параметры],

**\transblindsvertical**< слайды >[ параметры]

Позволяют осуществлять переключение между слайдами горизонтальными или вертикальными полосами.

Команды

**\transboxin**< слайды >[ параметры] – выполняет эффект сжимающегося к центру слайда.

**\transboxout**< слайды >[ параметры] – раскрывает следующий слайд от центра экрана.

**\transglitter**< слайды >[ параметры] – выполняет эффект мерцания в заданном направлении.

Можно добавить на слайды эффекты анимации, связанные с последовательным динамичным появлением информации на слайде. С помощью окружения **block** можно визуально выделить блоки на слайде, которые будут динамично на нем появляться (Рисунок 24).

```

\begin{block}{Текстовый процессор}
  "Microsoft Word, OpenOffice Writer"
\end{block} \pause

\begin{block}{Табличный процессор}
  "Microsoft Excel, OpenOffice Calc"
\end{block} \pause

\begin{block}{Графический редактор}
  "Adobe Photoshop, 3D Studio Max"
\end{block}
\end{frame}

```

Рисунок 24. – Добавление на слайды эффектов анимации

Использование в программном коде **\pause** даст возможность создать эффект последовательного появления информации на слайде. Сначала отобразится пустой слайд. После нажатия клавиши **пробел** или **Enter** появится

первая цитата, при следующем нажатии пробела появится вторая, аналогично третья цитата. Будет при этом сгенерировано четыре слайда. Осуществляя просмотр с помощью «**Внешнего просмотрщика**» (пиктограмма  на панели инструментов) можно наблюдать созданный динамический эффект появления информации на слайде (Рисунок 25).

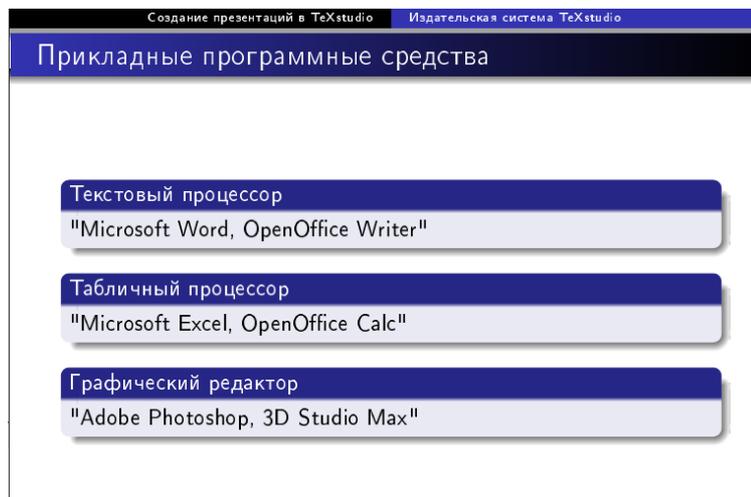


Рисунок 25. – Пример использования эффекта последовательного появления информации на слайде

Другой способ в **beamer** появления информации на слайде – это использование оверлеев. Если считать, что оверлеи пронумерованы то можно передать номер нужной команде. Например, команда `\item<5->` соответствует тому, что на 5-м оверлее появится пункт перечисления, а по команде `\color<5>;{red}` на пятом оверлее текст будет красного цвета (Рисунок 26).

```

\begin{frame}
% еще один способ "озаглавить" слайд
\frametitle{Рассмотрим классификацию программного обеспечения}
%подзаголовок
\framesubtitle{Прикладные программные средства}
\begin{itemize}
\item<5-> \color<5>;{red} Текстовые процессоры.
\item<3-> \color<4>;{yellow} Табличные процессоры.
\item<2-> \color<3>;{green} Программы демонстрационной
графики.
\item<4-> \color<4->;{blue} Системы управления базами данных.
\item<1-> \color<1-3,5>;{brown} Программы видеомонтажа
\end{itemize}
\end{frame}

\end{document}

```

Рисунок 26. – Появление информации на слайде

Слайд с библиографией можно создать так как показано на Рисунке 27.

```
\begin{frame}{Список литературы по теме "Классификация программного
обеспечения"}
\begin{thebibliography}{10}

\bibitem{LurkHorse}
{\sc Орлов С.А.}, {\em Технологии разработки программного
обеспечения / С.А. Орлов. - М.: Питер, 2017. - 464 с.}.
\bibitem{AbsHorse}
{\sc Советов Б.Я.}, {\em Моделирование систем. Практикум / Б.Я.
Советов, С.А. Яковлев. - М.: Юрайт, 2018. - 296 с.}.
\end{thebibliography}
\end{frame}
```

Рисунок 27. – Создание слайда с библиографией

Результат работы программного кода показан на рисунке 28.

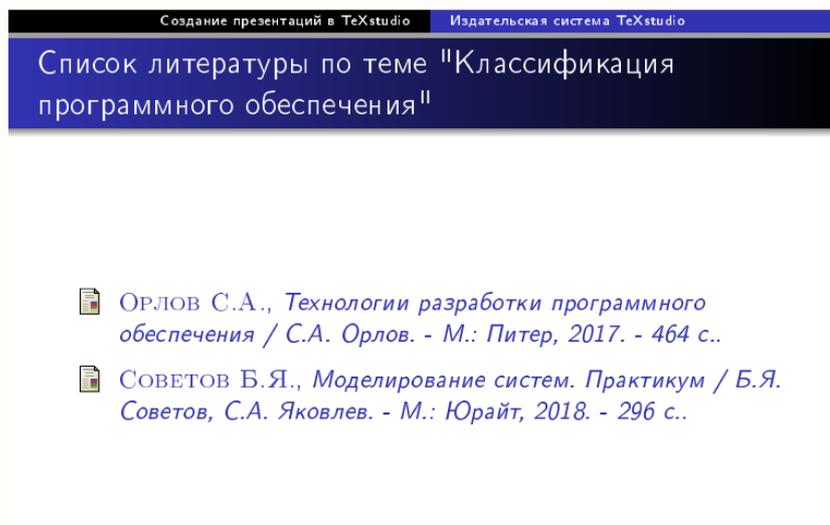


Рисунок 28. – Пример создания библиографического списка литературы

### Интерактивные элементы управления

В LaTeX существуют широкие возможности по использованию элементов управления, гиперссылок и пр. в документе .pdf.

### Гиперссылки

В целях использования в презентации гиперссылок нужно подключить пакет **hyperref**. При этом мы получаем возможность ссылаться как на элементы внутренние или другие документы или ресурсы Интернет, но при этом для организации перехода к некоторому элементу внутри текущего документа –

этот элемент должен иметь уникальную метку, которая создается командами **\hypertarget** или **\label**.

Команда **\hypertarget<слайд>{метка}{текст}** выводит «*текст*», являющийся мишенью для ссылки на слайде, указанном в параметре «*слайд*». Параметр «*текст*» может быть пустым. Параметр «*метка*» может состоять только из латинских символов, цифр и символов разделителей (двоеточие, подчеркивание и пр.)

Метка, создаваемая командой **\hypertarget**, размещается только на одном слайде. Если параметр «*слайд*» не указан явно, то метка будет размещена на первом слайде фрейма.

Команда **\label{метка}** создает метку на каждом слайде фрейма. Имя метки задается в формате метка «*номер слайда*».

Для перехода к слайду, помеченному командой **\hypertarget** или **\label**, используется команда **\hyperlink<слайды>{метка}{текст}**.

При этом выводится «*текст*» на слайдах <*слайд*>, при нажатии на который осуществляется переход на указанный слайд.

Однако можно переходить на слайды и по командам без создания меток. Рассмотрим ряд из них:

**\hyperlinkslideprev<слайды>{текст}** – переход на один слайд назад;

**\hyperlinkslidenext<слайды>{текст}** – переход на один слайд вперед;

**\hyperlinkframestart<слайды>{текст}** – переход на первый слайд текущего фрейма;

**\hyperlinkframeend<слайды>{текст}** – переход на последний слайд текущего фрейма.

И ряд аналогичных команд управления переходом.

### **Кнопки**

Команда **\beamerbutton{текст}** рисует на слайде кнопку. Для того, чтобы нажатие на кнопку приводило к каким-либо действиям, команда рисования кнопки должна быть использована внутри гиперссылки.

**Пример**

```

\begin{frame}[label=newframe]
  \begin{itemize}
    \item<+--> Один
    \item<+--> Два
    \item<+--> Три
  \end{itemize}
  \hyperlink<3>{newframe<1>}{\beamerbutton{В начало}}
\end{frame}

```

Помимо стандартной команды для рисования кнопки существует набор дополнительных команд:

**\beamerbutton{текст}** – показывает, что нажатие на кнопку переводит в другую область презентации;

**\beamerreturnbutton{текст}** – возвращает к предыдущему тексту.

**Задание для самостоятельной работы**

Разработать и оформить презентацию по теме. Презентация должна состоять из шести слайдов: титульный слайд, содержание, тематические слайды. В презентации должны присутствовать следующие возможности: установленная тема, переход между слайдами с помощью кнопок и гиперссылок, рисунки, таблицы, анимационная картинка, эффекты перехода между слайдами и эффекты появления объектов на слайде, звуковой ролик, список литературы.

Тематика презентаций:

1. Основные сведения о мультимедиа. Особенности и классификация мультимедиа компонентов.
2. Аппаратные средства мультимедиа технологии. Особенности и классификация мультимедиа компонентов.
3. Программные средства мультимедиа технологии. Классификация, назначение и функциональные возможности.

4. Мультимедиа продукты учебного назначения.
5. Инструментальные средства для создания интерактивной анимации.
6. Flash – технологии. Назначение и функциональные возможности программных средств реализации Flash – технологии.
7. Звуковые системы персонального компьютера. Основные характеристики.
8. Инструментальные средства для работы со звуком. Назначение и функциональные возможности.
9. Основы компьютерных видеотехнологий.
10. Компьютерная обработка видеоданных.
11. Интернет технологии для работы с мультимедиа технологиями.
12. Сетевые возможности передачи видеoinформации образовательного назначения
13. Классификация и области применения мультимедиа технологий.
14. Анимация. Программные средства создания анимации.
15. Технология поп-арт. Обрисовка фотографии в технологии поп-арт.
16. Форматы обработки текстовой, графической, аудио и видео информации.
17. Физические основы, аппаратные средства мультимедиа технологий.
18. Проявление мультимедиа технологий в инструментальных интегрированных средах разработчика мультимедиа продуктов.
19. Проявление мультимедиа технологий в продуктах профессионального назначения, электронных образовательных ресурсах и программных средствах компьютерного обучения.

20. Области применения мультимедиа технологий. Примеры использования.

21. Компьютерная графика, как компонент мультимедиа технологии.

22. Программные средства обработки аудио информации. Классификация, назначение и функциональные возможности.

23. Программные средства обработки звуковой информации. Классификация, назначение и функциональные возможности.

24. История развития вычислительной техники.

25. История развития мультимедиа технологии.

26. Фрактальная графика.

27. Функциональные возможности программы PowerPoint.

### Список литературы

1. Львовский С.М. Набор и верстка в пакете LaTeX. – 5-е изд., перераб. / С.М. Львовский – М.: МЦНМО, 2014. – 400 с. – ISBN 978-5-4439-0239-5
2. Насыров И.А. Установка и настройка MikTeX. / Насыров И.А. – Казань. Институт физики КФУ, 2016. – 50 с. – URL: [http://shelly.kpfu.ru/e-ksu/docs/F1581959532/Inst\\_MiKTeX.pdf](http://shelly.kpfu.ru/e-ksu/docs/F1581959532/Inst_MiKTeX.pdf) (Дата обращения 12.01.23)
3. Кнут, Дональд Эрвин. Все про TEX / Дональд Э. Кнут; (Пер. с англ. Л.Ф. Козаченко). – М. (и др.): Вильямс, 2003, ГПП Печ. Двор. – 549 с.: ил., табл.; 24 см. – (Серия: Компьютеры и вёрстка) – ISBN 5-8459-0382-3
4. Ширяева Е.В., Ширяева И.В. Введение в TeX. Часть 1. Набор и верстка текста. Учебное пособие / Е.В. Ширяева, И.В. Ширяева. – Ростов н/Д: Изд-во ЮФУ, 2015, 100 с., ил. – URL: [https://edu.mmcs.sfedu.ru/pluginfile.php/37265/mod\\_resource/content/1/\\_LabTeX\\_Main\\_1.pdf](https://edu.mmcs.sfedu.ru/pluginfile.php/37265/mod_resource/content/1/_LabTeX_Main_1.pdf) (Дата обращения 14.01.23)
5. Котельников И.А., Чеботаев П.З. LaTeX по-русски. – 3-е изд. Перераб. и доп.. / И.А. Котельников, П.З. Чеботаев – Новосибирск: Сибирский Хронограф, 2004. – 496 с., ил. – ISBN 5–87550–195–2

## Оглавление

Введение.....	4
Лабораторная работа 1. Основные приемы набора текста LaTeX.....	6
Лабораторная работа 2. Создание и редактирование списков и таблиц.....	27
Лабораторная работа 3. Форматирование текстового документа LaTeX.....	42
Лабораторная работа 4. Создание математических формул.....	52
Лабораторная работа 5. Титульный лист, подчеркивание, рамки, переносы.....	69
Лабораторная работа 6. Рисунки и графика.....	78
Лабораторная работа 7. Сервисы для документов LaTeX.....	95
Лабораторная работа 8. Презентации LaTeX.....	107
Список литературы.....	124

Суханова Надежда Тимофеевна

# ПОДГОТОВКА ТЕКСТОВЫХ ДОКУМЕНТОВ С ПОМОЩЬЮ ИЗДАТЕЛЬСКОЙ СИСТЕМЫ LaTeX

Учебное пособие

Редактор:  
Н. В. Викулова

Подписано в печать      Формат 60x90 1/16 Бумага газетная. Печать трафаретная.  
Уч. изд. л. 7,6. Усл. печ. л. 7,8. Тираж 300 экз. Заказ №

---

---

Федеральное государственное бюджетное образовательное учреждение высшего образования «Нижегородский государственный архитектурно-строительный университет»  
603950, Нижний Новгород, ул. Ильинская, 65.

Полиграфический центр ННГАСУ, 603950, Н.Новгород, Ильинская, 65  
<http://www.nngasu.ru>, [srec@nngasu.ru](mailto:srec@nngasu.ru)