Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Теоретическая информатика и компьютерные технологии»

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К КУРСОВОЙ РАБОТЕ

НА ТЕМУ:

«Здесь пишем тему»

Студент _____ _____ _____
(Группа) (Подпись, дата) (И.О. Фамилия)

Руководитель _____ _____
(Подпись, дата) (И.О. Фамилия)

2021 г.

АННОТАЦИЯ

Записка состоит из 25 страниц, содержит 4 разделов, 4 листингов, 1 таблиц и 2 рисунков. Окончания согласованы криво, потому что подсчёт страниц, разделов, таблиц, рисунков и листингов осуществляется автоматически с помощью пакетов LaTeX. Если захочется автоматически посчитать что-то ещё, нужно добавить в преамбулу документа строку `\DeclareTotalCounter{что-то ещё}` и затем воспользоваться командой `\totalчто-то ещёs{}` там, где потребуется.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Главное правило новичка в TEX — абзац начинается с пустой строки. Просто перевод строки новый абзац не определяет.

Важнейший служебный символ в TEX — это обратный слеш, \. Если нужно добавить его в текст, нужно пользоваться тегами `\textbackslash` или `\backslash` (разница — см. Раздел 2). Большинство других служебных символов, таких как $, {, }, _, #, &, ^, %, записываются посредством экранирования их \-префиксом. А вот экранирование обратного слеша обратным слешем \\
действует совсем иначе. Это перевод строки без образования нового абзаца.

Фигурные скобки { } ограничивают действие тега.

Кавычки-«ёлочки» записываются как два идущих подряд уголка, <<, >>. Кавычки-"лапки" записываются собственно знаком кавычки, но пробел после такой кавычки может съедаться, поэтому ограничиваем действие кавычки пустыми фигурными скобками, вот так: `"{}`. Такие же пустые фигурные скобки разрывают комбинацию символов, например, позволяют записать две угловые скобки подряд без преобразования их в «ёлочку», и показывают, что после вызова тега нужно поставить пробел (который автоматически бы подавился тегом), например, { так.

Содержание формируется автоматически. В основном тексте туда попадает всё вплоть до подразделов второго уровня вложенности, в приложениях — только разделы (т.е. собственно приложения).

1 Название раздела

Название раздела идёт под тегом `section`, подраздела — `subsection`. Особые разделы, не имеющие нумерации (например, приложения) идут под тегом `anonsection`. Вот так: `\anonsection{НАЗВАНИЕ}`.

Каждый большой раздел начинается с новой страницы, так же как и все особые разделы. Подразделы LaTeX старается начать так, чтобы заголовок подраздела не был разорван с первым абзацем. Также LaTeX старается не отрывать единственную строчку от абзаца, и с этой целью может немного поменять расстояния между заголовками и текстом и между текстом и плавающими объектами. Обычно это не критично для нормоконтроля, а получается красиво.

Помним, что основной единицей текста в LaTeX является абзац! Если транслятор LaTeX показывает ошибку или предупреждение в строке X, надо искать её внутри всего абзаца, к которому относится эта строка. Если вставляется хак для изменения вертикальных расстояний (см. Раздел 1.1), он будет применён в начале абзаца, а не в месте его расположения.

Таблицы, перечисления, листинги, как правило, считаются принадлежащими одному абзацу.

1.1 Про форматирование

Начертания шрифтов и альтернативные семейства шрифтов:

— *курсив* — `\textit{курсив}`;

— **полужирный** — `\textbf{полужирный}`;

— `моноширинный` — `\texttt{моноширинный}`. Хорошим тоном считается использовать его в листингах и при указании фрагментов кода внутри текста. Впрочем, в листингах обычно он применяется автоматически.

Есть и другие, но в этой болванке они автоматически приводятся к одному из вышеуказанных. Зачёркивания, подчёркивания и прочие украшения к начертаниям, разумеется, не относятся. Обычно начертания используются, чтобы выделить определённые фрагменты текста, и внутри окружений (см. Раздел 1.3) могут сливаться с основным текстом. Директива `\emph` включает изменение начертания в зависимости от окружения. Если окружение стандартное (основной текст), эта

директива тупо включает курсив, если не подключены никакие дополнительные пакеты, и подчёркивает аргумент, если подключён пакет `ulem`.

Хаки для форматирования:

1. `\hspace{мера}`. Увеличить или уменьшить расстояние между объектами по горизонтали. Мера может быть в шрифтонезавимых пойнтах (`pt`, примерно три миллиметра) либо в шрифтозависимых иксах (`ex`, один символ). И ещё во многих разных мерах, но обычно хватает этих двух. Например,
   - БЛА БЛА — стандартное расстояние;
   - БЛА БЛА — добавили полтора пойнта (`\hspace{1.5pt}`);
   - БЛА БЛА — добавили полтора икса (`\hspace{1.5ex}`);
   - БЛА БЛА — удалили половину пойнта (`\hspace{-0.5pt}`);
   - БЛАБЛА — удалили половину икса (`\hspace{-0.5ex}`).
2. То же самое для вертикали: `\vspace{мера}`. С этим хаком желательно быть осторожней, чтобы не изменить размеры полей на странице.
3. Горизонтальные пробельные символы:
   - `\,` — половина пробела (см. бла бла);
   - `\;` — насильственный пробел (см. бла бла);
   - `\quad` — большой пробел, примерно с ширину буквы М (см. бла бла);
   - `\qquad` — два больших пробела (см. бла бла).
4. Неразрывный пробел — символ ~.
5. Вертикальные пробельные символы:
   - `\smallskip` — малый пробел;
   - `\medskip` — средний пробел;
   - `\bigskip` — попробуйте сами догадаться.

   Но вообще, это всё — версии `\vspace`, просто они позволяют не думать, какую меру писать в его аргументе. Пользоваться ими надо обязательно в начале нового абзаца, потому что они применяются к абзацу, внутри которого стоят.
6. `\noindent` — подавить абзацный отступ. Ставим только в самом начале абзаца и только если хорошо понимаем, что делаем, потому что по ГОСТу `\noindent` — зло.

1.2 Перечисления

Как видно из предыдущего подраздела, встроенная структура перечисления может быть нумерованной или маркированной. Нумерованный список обрамляется окружением `\begin{enumerate}` и `\end{enumerate}`; маркированный обрамляется окружением `\begin{itemize}` и `\end{itemize}`. Каждый новый пункт перечисления, неважно, нумерованного или маркированного, начинается с тега `\item`.

Перечисления могут быть вложенными Поскольку каждый новый уровень перечисления подразумевает всё больший отступ, не рекомендуется использовать больше, чем два уровня перечисления, чтобы не получилось вот так.

1. Это первый уровень перечисления.
    (1) Это второй уровень перечисления.
        (i) Это третий уровень перечисления.
            А. Э. Т. О. кошмар нормоконтролёра.

Если маркер перечисления заканчивается точкой (как первый уровень в нумерованных списках), то и в конце предложения — элемента списка ставим точку. Соответственно, очередной элемент пишем с заглавной буквы. В случае маркированных списков (маркер — тире), и если маркер перечисления заканчивается скобкой, в конце каждого элемента списка, кроме последнего, ставим точку с запятой, и начинаем предложения со строчной буквы.

1.3 Математические окружения

В LaTeXсуществуют пакеты, которые позволяют создавать математические окружения, незаменимые для специальности «прикладная математика и информатика». Эти окружения имеют встроенное форматирование и автонумерацию, кроме того, на них удобно расставлять кликабельные ссылки.

Определение 1. *Это окружение типа определение, ограничивается тегами* `\begin{Definition}...\end{Definition}`. *Кстати, здесь использован тег* `\emph`, *кроме того, основной текст внутри окружения стал курсивным.*

Теорема 1. *Это окружение типа теорема. Если слово «теорема» кажется вам слишком громким для утверждения, используемого в работе, хорошим тоном считается пользоваться окружением типа лемма (см. Лемма 1).*

Лемма 1. *Это окружение типа лемма. Если в вашей работе формулируется и доказывается какое-то теоретическое утверждение, лучше всего пользоваться именно этим окружением, а окружение «теорема» оставить для исторически признанных теорем.*

Доказательство. Это — горячо любимое преподавателями математики окружение «доказательство». Основной текст в этом окружении — опять в стандартном начертании. Обратите внимание на знак завершения доказательства по правому краю страницы.

Если вы формулируете собственную лемму в работе, не забудьте воспользоваться окружением типа «доказательство» сразу после формулировки. Можно даже внутри окружения «лемма». □

Лемма 2. *Это лемма со встроенным доказательством.*

Доказательство. Это доказательство внутри окружения лемма. □

Пример 1. *Это — палочка-выручалочка для любого начинающего волшебника, если нужно ~~просто долить воды~~ добавить объём текста, а чем — непонятно.*

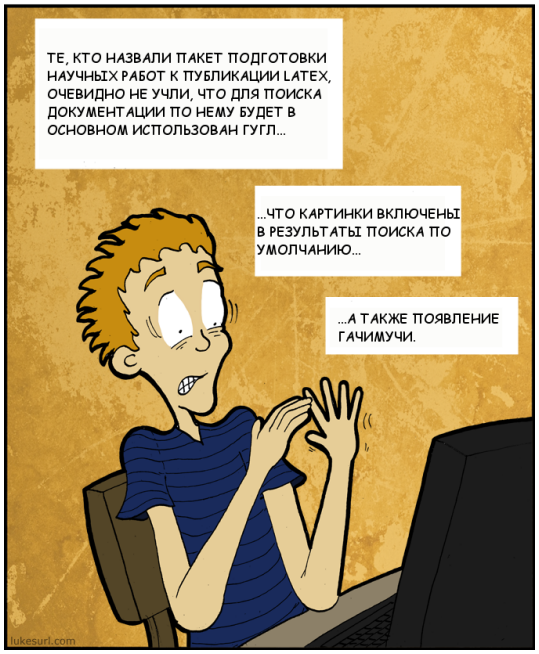
1.4 Таблицы, рисунки, листинги

1.4.1 Рисунки

Окружение типа рисунок обрамляется командами `\begin{figure}[!htb]` и `\end{figure}`. Директива `[!htb]` — это «пытаться разместить рисунок на том месте, где он расположен в исходнике, но если коряво лезет — то вверху страницы, а если и так, и эдак всё равно коряво — то внизу». Сразу над директивой `\end{figure}` размещаем информацию о названии рисунка под тегом `\caption{...}`. Приписывать слово «рисунок» не надо — это стиль сделает автоматически. Если хочется назвать рисунок альтернативно, можно подавить стиль с помощью звёздочки после слова `caption` вот так: `\caption*{...}`.

После объявления `\begin{figure}[!htb]` обязательно включаем центрирование тегом `\centering`, это положено по ГОСТу. А вот центрировать ли элементы составной картинки, решать нужно уже по случаю. Обычно красивее будет центрировать. В примере ниже для размещения двух картинок рядом друг с другом использовано окружение `minipage`.

а) Картинка не отцентрирована.



б) Картинка отцентрирована.

Рисунок 1 — Пример оформления рисунка.

Внешний рисунок вставляется в картинку командой `\includegraphics`. У неё два аргумента: в квадратных скобках указывается ширина рисунка, в фигурных — имя файла. Удобнее всего рассчитывать ширину рисунка исходя из актуальной ширины текстового поля, т.е. в долях от константы `\textwidth`. На рисунке 1 обе картинки включены с параметром `0.75\textwidth`, но разница существенная, поскольку ширина первой колонки (мини-страницы) составляет 40% от ширины общего текстового поля, а ширина второй мини-страницы — 55%.

Окружение `minipage` также формируется с помощью `\begin{minipage}[тип выравнивания]{ширина мини-страницы}...\end{minipage}`. Тип выравнивания определяет вертикальное выравнивание содержимого мини-страницы и может быть `c` (центр), `t` (верх), `b` (низ)[1]. Ширину мини-страницы тоже удобно описывать как долю от актуальной ширины текстового поля. Если не начинать новый абзац между

[1]Иногда выравнивание по верху и по низу даёт одинаковый результат, однако.

описаниями мини-страниц, LaTeX попытается выстроить их на одном уровне. Но получится это у него, только если суммарная ширина всех мини-страниц будет чуть-чуть меньше, чем `\textwidth`. Обычно достаточно оставить зазор в 5% ширины, но в принципе можно подобрать и меньшее значение ε. Например, рисунок 1 формирует две параллельные колонки, если заменить в заголовке первой мини-страницы `.4\textwidth` (исходные 40% ширины) на `.443\textwidth`.

1.4.2 Таблицы

Таблицы оформляются похожим образом. Прежде всего, у достаточно большой таблицы должно быть имя и возможность на неё ссылаться, поэтому такая таблица погружается в окружение `table`. Это окружение имеет уже знакомую по предыдущим разделам `begin-end` структуру. В квадратных скобках после `\begin{table}` — рекомендация для LaTeX, как размещать таблицу относительно страницы. Удобно использовать такую же, как для рисунков, — `htb`.

Название рисунка приводится по ГОСТу под ним, а название таблицы — наоборот, до неё. Поэтому тег `\caption` размещаем на следующей строке после `\begin{table}`. Центрирование внутри окружения `table` также включать обязательно.

Окружение `tabular` формирует саму таблицу. Заголовок у него такой: `\begin{tabular}{список форматов ячеек}`. Формат ячейки — это тип выравнивания её содержимого по ширине (`r` — правый край, `l` — левый край или `c` — середина) и указание, нужны ли вертикальные линии по бокам ячейки. Причём этих линий можно ставить сколько угодно.

Примеры заголовков:

— `\begin{tabular}{||rllc}` — таблица с четырьмя столбцами, первый выровнен по правому краю, последний по центру, остальные по левому краю. Слева от первого столбца двойная вертикальная линия;

— `\begin{tabular}{l|l|l|lll}` — таблица с шестью столбцами, все выровнены по левому краю. Три первых столбца отделены вертикальными линиями.

Сама таблица использует команды & — перейти к ячейке следующего столбца данной строки; и \\ — завершить строку. Если хочется отделить строку линией,

Таблица 1 — Таблицы, рисунки, листинги.

Особенность	Таблицы	Рисунки	Листинги
Окружение	`table`, а внутри `tabular`	`figure`	`listing`, а внутри `minted` или `inputminted`
Название	над таблицей	под рисунком	над листингом
Центрирование	Да		Нет
Ссылка	Непосредственно перед `end`–элементом		

после директивы завершения строки дописываем `\hline`. Делаем это столько раз, сколько параллельных линий хотим провести, но нужно учесть, что при такой многократной отрисовке LaTeX коряво оформляет уголки — точки пересечения дополнительных линий. См. таблицу 1. Слияние строк и столбцов в таблице осуществляется командами `\multirow` и `\multicolumn` соответственно. Про их использование можно почитать в мануалах LaTeX.

1.4.3 Листинги

Комментарии в коде на английском языке временно не работают на территории Российской Федерации, поэтому в рамках импортозамещения в болванке осуществлён переход с пакета `listings` на пакет `minted`. Теперь трансляция `latex`-кода надёжна, как никогда ~~поскольку разрешает запуск сторонних скриптов~~: для обеспечения её скрепности нужно при использовании команды `pdftolatex` добавить ключ исполнения `--shell-escape`.

Листинг мы помещаем в специальное окружение `listing` и стандартно добавляем туда `\caption`- (до самого листинга) и `\label`-параметры. Собственно листинг обрамляется окружением `minted`. Открывающий тег имеет вид `\begin{minted}[`*тут параметры оформления листинга*`]{`*тут название языка*`}`.

Некоторые параметры оформления листинга:

— `fontsize` — определяет размер кода;

— `frame` — задаёт форму рамки;

— `linenos` — включает нумерацию строк;

Листинг 1: Пример листинга

```
int main(){
/* /\---/\ */
/* \ O O / */
/*  \ Y /  */
/*   ---   */
/* Этот комментарий разрешён на территории РФ */
}
```

— `xleftmargin` — увеличивает отступ слева (обязательно нужен, если включена нумерация строк, иначе вас сожрёт нормоконтроль);

— `breaklines` — полезный атрибут, позволяющий окружению вставлять переносы в длинные строки. Знак переноса можно подавить, добавив параметр `breaksymbol = ""`.

Внутри окружения `minted` TEX-овские спецсимволы не распознаются (в частности, все обратные слеши в листинге 1 были записаны просто символами), разметка основного текста не применяется, а каждая новая строка начинается как во всех нормальных языках программирования — не с пустой строки, а просто с символа перевода каретки, в точности как в теховском окружении `Verbatim` (именно так, с заглавной; собственно, это окружение и генерирует python-овский скрипт, вызываемый пакетом `minted`). Кроме того, автоматически применяется моноширинное начертание.

По последней инсайдерской информации, листинги считаются программой нормоконтроля картинками, поэтому цветной код в листинге 1 — не проблема для ВКР.

А ещё листинги можно читать из файла командой `\inputminted[аргументы]{имя файла}`. Аргументы здесь такие же, как и у `minted`. Пример — Листинг 2. Ничему не надо удивляться, здесь рефал-исходник обработан лексером пролога, потому что встроенного лексера для рефала в пакете нет. Хотя можно подключить свой лексер ~~помним же, теперь наш код на LATEX особенно надёжен~~, но это пока future work. Всё равно ВКР с рефаловскими листингами в этом году не предвидится.

Листинг 2: Запрещённый на территории РФ листинг, считанный из файла SCP_new.ref.

```
/* The program takes two shell arguments. */
$ENTRY Go {
    , <Arg 1> : e.input0
    , <Arg 2> : e.output0
     = <Open r 1 e.input0>
       <Open w 2 e.output0>
        <CopyFile 1 2>
        <Close 1><Close 2>;
}

/* A stupid function copying the input stream to the output. */
CopyFile {
   s.Input s.OutPut, <Get s.Input> : e.x 0
         = <Put s.OutPut e.x>
           <Put s.OutPut '/* This program is generated by an efficient
               new supercompiler. '>
           <Put s.OutPut '   It is guaranteed to preserve semantics and
               not to decrease run-time! */'>;
   s.Input s.OutPut, <Get s.Input> : e.x
         = <Put s.OutPut e.x><CopyFile s.Input s.OutPut>;
}
```

1.5 Перекрёстные ссылки

Это — второе, ради чего стоит использовать LaTeX (первое — см. Раздел 2). Система формирования кликабельных ссылок здесь довольно удобна. Большинство объектов, описанных в этом разделе, позволяют создать ссылку на них с помощью команды `\label`. Размещать эту команду в окружении надо аккуратно: в теоремах, леммах, примерах и определениях — сразу после тега `\begin`, без перевода строки; в рисунках и таблицах — непосредственно перед тегом `\end`. Естественно, можно делать ссылки на разделы и подразделы — в этом случае `\label` также ставим сразу после тега раздела, без перевода строки.

Имя ссылки может содержать латинские цифры и буквы и указывается в фигурных скобках: `\label{имя ссылки}`. Принято, что имя ссылки имеет вид `тип объекта:идентификатор`, где тип объекта — это `section`, `example`, `figure`, `table` и т.д. или их сокращения.

Ссылаемся на объект командой `\ref{имя ссылки}`. Обычно требуется скомпилировать исходник дважды, прежде чем корректно расставятся все ссылки. Доказательства по умолчанию не являются объектами, на которые можно делать ссылки.

2 Математика

Кто уже оформлял математические тексты, знает, что формулы удобнее набирать не в формате WYSIWYG, а с помощью простого языка программирования. В LaTeX этот язык отлично разработан и после небольшой практики даёт возможность забыть о проблемах с формулами, невыровненными по тексту или отображающимися кракозябрами. В LaTeX есть специальная «математическая мода» (режим набора математических формул). Включается математический режим символом $ или сочетанием `\(`, выключается — тем же $ или сочетанием `\)`.

Вот некоторые особенности режима записи математики:

— изменены расстояния между символами по умолчанию;

— обычные пробельные знаки игнорируются (те, что перечислены в списке хаков — нет);

— начертание букв по умолчанию — курсивное, причём оно отличается от курсива в обычном тексте. Цифры в математическом режиме пишутся не курсивом, но в улучшенном начертании. Сравним $blahBLAHblah0987$ и *blahBLAHblah*0987.

В математическом режиме есть свои команды для изменения начертания шрифта:

— `\mathit` — математический курсив: $1234XYZxyz$;

— `\mathbf` — математический полужирный: $\mathbf{1234XYZxyz}$;

— `\mathtt` — математический моноширинный: $\mathtt{1234XYZxyz}$;

— `\mathrm` — математический прямой: $\mathrm{1234XYZxyz}$;

— `\mathfrak` — готический: $\mathfrak{1234XYZxyz}$;

— `\mathcal` — каллиграфический: $\infty\in\ni\triangle\mathcal{XYZ}\S\dagger\ddagger$;

— `\mathbb` — прозрачный: $\nVdash\nvDash\nVDash\varsubsetneqq\mathbb{XYZ}\curvearrowleft\curvearrowright F$;

— `\mathscr` — «Ralph Smith formal script»: $\mathscr{X}\mathscr{Y}\mathscr{Z}$. Здесь под тегом такая же строка, как и в предыдущих примерах, но строчные буквы и цифры просто не отображаются в этом начертании.

Как видно, последние три начертания имеет смысл использовать только для заглавных латинских букв, иначе результат может оказаться неожиданным. Впрочем, обычно пользователи именно для них эти начертания и применяют: прозрачным принято записывать значки для числовых множеств, а каллиграфический и RSFS

чаще всего используются для обозначения множеств или структур, вводимых автором. Переменные ими не обозначают, впрочем, как и готическим.

Пример 2. *Формула нормального человека:* $x_i^2 + y_i^2 = \frac{L*\sqrt{S_{i+2}}}{2}$*; формула курильщика:* $\mathbb{X}_{\mathtt{i}}^{\mathtt{2}} + \mathcal{Y}_{\mathtt{i}}^{2} = \frac{\mathscr{L}*\sqrt{\mathfrak{S}_{\mathtt{i+2}}}}{\mathbf{2}}$.

Как видно из Примера 2, чрезмерно креативное использование альтернативных математических начертаний в тексте не приветствуется. Кроме того, существует негласное правило, что имена переменных чувствительны к начертанию: вот такой $\mathtt{x}$ — не то же, что такой $\mathbf{x}$, и не то же, что такой x. Естественно, если начертания нельзя различить на глаз, например, как в x и x (второй записан курсивом), такие переменные считываются как одинаковые. Но вообще, чтобы не запутаться в начертаниях и именах переменных, удобно пользоваться макросами (см. Раздел 2.2).

Нормальной кириллицы в математическом режиме нет, но её можно включить в формулу с помощью тега `\text{...}`.

Если формула достойна того, чтобы быть пронумерованной и выключенной из текста (по ГОСТу это одно и то же), тогда используем окружение `equation`. Включается и выключается оно стандартной `begin`–`end` структурой. Ссылка на формулу (тег `\label`) ставится сразу после `\begin{equation}`, без переноса строки. Автонумерацию, центрирование и переход в математический режим окружение обеспечит автоматически, так что добавлять знаки долларов не надо.

Пример 3. *Формула нормального человека:*

$$x_i^2 + y_i^2 = \frac{L * \sqrt{S_{i+2}}}{2}\,; \tag{1}$$

формула курильщика:

$$\mathbb{X}_{\mathtt{i}}^{\mathtt{2}} + \mathcal{Y}_{\mathtt{i}}^{2} = \frac{\mathscr{L} * \sqrt{\mathfrak{S}_{\mathtt{i+2}}}}{\mathbf{2}} \tag{2}$$

.

В формулу 1 мы были вынуждены добавить точку с запятой, потому что если бы поставили её вне окружения, она бы перенеслась на новую строку, как случилось с формулой 2. На то она и формула курильщика. Также из-за манеры математического режима изменять расстояния между символами пришлось добавить принудительный короткий пробел `\,` перед точкой с запятой.

2.1 Часто используемые фишки

2.1.1 Нижние и верхние индексы

Режим нижних и верхних индексов работает только в математическом режиме. Нижний индекс — это символ или группа символов в фигурных скобках после знака подчёркивания _; верхний индекс — то же после крышки ^.

Если индекс состоит больше чем из одного символа, про фигурные скобки забывать нельзя, иначе все символы, кроме первого, будут уже не считаться индексом. Например, если x хочется снабдить сложным индексом $i + k * m$, то записав это вот так: $x_i + k * m$, получим совсем не то, что нужно. Правильно: x_{i+k*m}.

Нижние и верхние индексы могут быть использованы внутри индексов и образовывать какие угодно сочетания, например: $A^{B^{B^{B}}}_{E_{F_G}}$. С каждым новым уровнем индексов вплоть до третьего размер шрифта уменьшается, а дальше остаётся постоянным.

2.1.2 Логические и теоретико-множественные формулы

С ними довольно часто приходится иметь дело, поэтому здесь будут перечислены основные команды для записи логики:

— `\forall` — знак всеобщности $\forall$;
— `\exists` — знак существования $\exists$;
— `\wedge` или `\&` — конъюнкция $\wedge$ или $\&$;
— `\vee` — дизъюнкция $\vee$;
— `\Rightarrow` — импликация $\Rightarrow$, не путаем с `\rightarrow` — стрелкой $\rightarrow$;
— `\neg` — отрицание $\neg$;
— `\subseteq` — нестрогое отношение подмножества $\subseteq$;
— `\subset` — строгое отношение подмножества $\subset$;
— `\cup` — пересечение множеств $\cup$;
— `\cap` — объединение $\cap$;
— `\backslash` — разность множеств $\backslash$ — в обычном тексте этот тег работать не будет;

— `\in` — знак принадлежности $\in$;

— `\notin` — знак непринадлежности $\notin$.

Далее мы в тему специальных символов не углубляемся, они перечислены в справочниках по LaTeX.

2.2 Макросы

Особенности математического режима таковы, что для красивого ввода имени, состоящего из нескольких латинских букв, приходится использовать хаки. Сравним $BUgs$ и $BUgs$ — в первом имени все четыре буквы записаны просто подряд, и оно выглядит как последовательность из двух двухбуквенных имён; во втором между U и g добавлен хак `\hspace{-.3ex}` и оно уже выглядит нормально. Однако добавлять такие хаки всюду по тексту записки — то ещё удовольствие, к тому же где-нибудь да он забудется. Чтобы минимизировать опечатки в идентификаторах, используемых в формулах, и сделать их внесение в текст менее трудозатратным, можно применить механизм макросов.

Для определения нового макроса–имени нужно в преамбулу документа, желательно после всех прочих записей в преамбуле, добавить команду `\newcommand{имя команды, начинается с бэкслэша}{тело команды}`. После этого тело команды будет инлайниться в код вместо указанного имени.

Пример 4. *Допустим, в преамбулу добавлена директива:*

```
\newcommand{\bug}{\mathbf{B\hspace{-.2ex}U\hspace{-.2ex}G}
\hspace{.1ex}\mathrm{A\hspace{-.2ex}G\hspace{-.2ex}A}}.
```

Она будет рисовать идентификатор $\mathbf{BUG}\mathrm{AGA}$ *везде, где встретит тег* `\bug`*, правда, вне математического режима скомпилируется с ошибкой, поскольку тело макроса использует команды, специфические для него. Теперь, если мы захотим заменить этот идентификатор, например, на* $\bot$*, это делается простейшей заменой макроса на* `\newcommand{\bug}{\bot}`*, и всё, дальше по тексту можно не проверять, все ли нужные идентификаторы заменились, и не заменилось ли что-то лишнее. Только помним, что если исходный идентификатор годился для математического режима, то его замена тоже должна использовать команды этого же режима.*

3 Подключение и сборка библиографии

Библиография готовится в отдельном файле с расширением `bib` и собирается утилитой `biber` со следующими параметрами (см. Листинг 3). Библиографический стиль соответствует требованиям `TestVKR`, а вот стиль цитирования взят из библиографической базы журнала Nature: отечественные стили цитирования могут падать на буржуйских элементах библиографии[1]. `bib`-файл — это база данных, где хранятся записи об использованных источниках. Конкретное имя этого файла указывается командой `\bibliography`. В нашем случае это `\bibliography{biblio}`.

Листинг 3: Параметры оформления библиографии

```
\usepackage[backend=biber,
    bibstyle=gost-numeric,
    citestyle=nature]{biblatex}
```

Порядок записей в `bib`-файле может быть какой угодно, как и полей в них — LaTeX упорядочит всё как надо сам. Для сборки библиографии нужно скомпилировать исходник LaTeX-ом, затем запустить `biber` (в формате `biber <имя исходника без расширения>`) и скомпилировать исходник дважды. Если ссылки всё ещё отображаются некорректно — возможно, LaTeX собрал информацию для вспомогательных файлов ещё не полностью, и рекомпиляция может решить эту проблему.

На примере элемента библиографии в Листинге 4 посмотрим на структуру записи. Имя ссылки — это самое первое поле, сразу после типа записи. Здесь имя `Vybegallo`. Чтобы процитировать элемент библиографии с таким именем в тексте, пишем `\cite{Vybegallo}`. Цитирование указывается в конце текстового отрезка, к которому оно относится. Желательно применять неразрывный пробел для связи команды `cite` с последним словом, ему предшествующим.

На результат можно полюбоваться в списке использованных источников [1]. Отметим некоторые детали:

[1] Кстати, `gost-numeric` тоже может заругаться, поэтому рекомендуется удалить из базы данных в `bib`-файле все вражеские атрибуты типа acmid-ов и прочего, что не поддерживается отечественными бюрократами.

Листинг 4: Пример элемента библиографии

```
@article{Vybegallo,
 author = {Vybegallo, A.A. and A.A. Vybegallo},
 title = {About Reverse Analysis of Gorilla Language},
 journal = {Vestnik NII CHAVO},
 year = {2222},
 pages = {1--111},
 number = {0},
 language = {english}
}
```

— не забываем экранировать служебные символы — библиография не листинг;

— после каждой записи, кроме последней, ставится запятая;

— если вы не уверены, что парсер обработает запись однозначно, лучше заключить её в фигурные скобки;

— при записи диапазона страниц используем короткое тире — записывается так --;

— если источник опубликован на английском языке, язык нужно указать, тогда как русский указывать не обязательно;

— если в имени автора инициалы идут после его фамилии, отделяем их от фамилии запятой;

— если автор не один, перечисляем их через служебное слово `and` (без выделения запятыми).

LaTeX неплохо распознаёт фамилию и инициалы. Как видно по ссылке [1], библиографический стиль поместил фамилию Амвросия Амвруазовича перед именем–отчеством, хотя во втором случае имя–отчество предшествовали фамилии.

Обычная для технических текстов информация с датой обращения к электронному ресурсу помещается в поле `annote`. Она автоматически будет заключена в круглые скобки.

Библиографическую информацию об источниках, опубликованных в индексируемых журналах, почти всегда можно найти на открытых страницах в формате, подходящем для базы данных `bib`. Современные зарубежные статьи чаще всего

есть на dblp.org, русскоязычные — на mathnet.ru (правда, там по умолчанию используется AMSbib-формат, чуть-чуть отличный от стандартного). На elibrary скачать bibtex-образец нельзя, но обычно есть ссылка на doi, которая может вывести на страницу, где он есть.

4 Чтение лога и тестирование

В логе (файл `.log`) можно посмотреть сообщения об ошибках, мешающих собрать документ, и о предупреждениях. Многие из последних — издержки взаимодействия разных пакетов. Перечислим только те, которые со стилем не связаны.

1. Если предупреждение имеет вид `Citation ... on page ... undefined` или `Reference ... on page ... undefined`, значит, есть опечатка в цитировании или обращении к ссылке.
2. Если в логе встретились слова `Underfull` или `Overfull`, это указывает на то, что текст не соответствует установленному формату полей. В целом, на недополнения не стоит обращать особенного внимания, за исключением случая, когда разреженность текста сильно бросается в глаза. Переполнения лучше исключить, кроме того, которое в болванке есть на титульной странице (эмблема МГТУ чуть-чуть вылазит влево за поле прозрачным фоном).

После того, как ошибок, потерянных ссылок и критических переполнений в логе не будет, можно убедиться, что LaTeX не слишком сильно изменил параметры основного шрифта в документе. Для этого открываем Word или его аналог и копируем кусок основного текста из свёрстанного pdf-документа. Имя шрифта должно быть из семейства TimesNewRoman, и обязательно 14-го размера.

ЗАКЛЮЧЕНИЕ

Этот исходник подготовлен с помощью системы LaTeX на базе приложения к положению о нормоконтроле МГТУ им. Н. Э. Баумана [2]. Подробно про возможности языка разметки можно почитать в обстоятельном труде Львовского [3], а большинство мелких вопросов решается поиском, например, в соответствующем разделе StackExchange [4]. Запросы в гугле по слову `'latex'` лучше формулировать как можно конкретнее (см. Рисунок 1).

Установка и настройка LaTeX — не самое приятное занятие, зато реакция преподавателей на результат может быть очень приятной (см. Рисунок 2). Конечно, всё это имеет смысл, если в вашей работе много формул и перекрёстных ссылок. Простой текст легче набивать в WYSIWYG-редакторе.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Vybegallo A. A., Vybegallo A. A. About Reverse Analysis of Gorilla Language // Vestnik NII CHAVO. — 2222. — № 1. — С. 1—111. — DOI: 10.1090/trans2/015.
2. ПРИЛОЖЕНИЕ 1 к положению «О нормоконтроле, размещении текстов в электронно-библиотечной системе и проверке на объем заимствования выпускных квалификационных работ бакалавров». — URL: https://mf.bmstu.ru/info/uu/ot/norm_docs/docs/polozhenie_normcontrol_pril1.pdf.
3. Львовский С. М. Набор и верстка в системе LaTeX. — Москва : Московский центр непрерывного математического образования, 2003.
4. StackExchange [Электронный ресурс]. — URL: https://tex.stackexchange.com/.

ПРИЛОЖЕНИЕ А

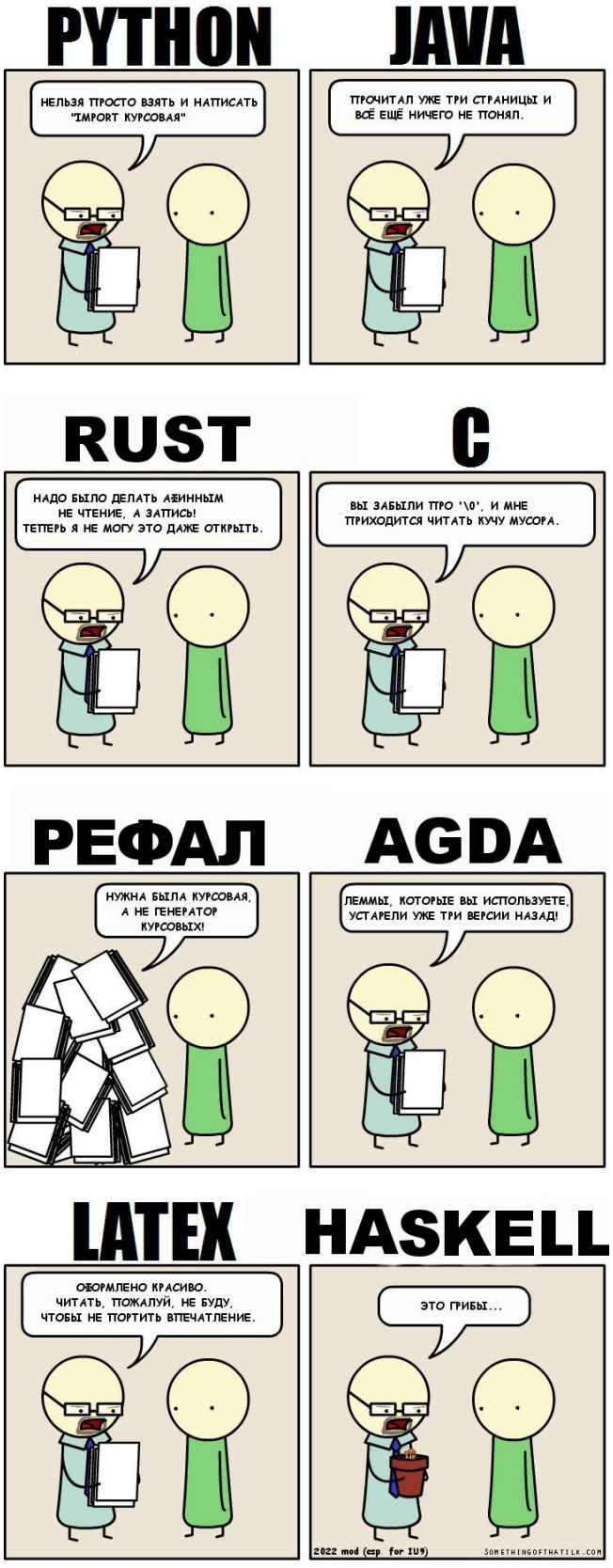


Рисунок 2 — Что было бы, если бы записки принимались на любом языке программирования.