РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ № 3

дисциплина: Архитектура компьютера

Студент: Земцов Роман

Группа: НКАбд-02-25

МОСКВА

2025 г.

Содержание

Цель работы

Целью работы является освоение процедуры оформления отчетов с помощью легковесного языка разметки Markdown.

Теоретическое введение

2.1 Базовые сведения о Markdown

Markdown - это облегченный язык разметки, который позволяет форматировать текст с помощью простых символов. Он преобразуется в валидный HTML. Его можно открывать и изменять в любом редакторе текста. Широко используется для написания документаций и README-файлов.

Он содержит базовые элементы, которые можно найти почти в любом README.md:

• заголовок первого уровня для названия;

• выделение жирным шрифтом важных частей в описании;

• ссылка с понятным текстом;

• заголовок второго уровня для подпунктов;

• маркированный список для перечисления преимуществ. Несмотря на то, что Markdown достаточно удобно читать в исходном виде, его часто переводят в HTML.

2.1.1 Параграф

Параграф — это одна или несколько подряд идущих строчек текста, отделённых одной или несколькими пустыми строчками. Если строка содержит только пробелы или табы, то она всё равно считается пустой.
Подряд идущие строчки будут склеены в одну, если не добавить жёсткий перенос. Существует несколько способов, как это можно сделать:

• добавить два (или больше) пробелов в конце строки ;
• добавить обратную косую черту в конце строки \;
• добавить HTML-тег переноса строки

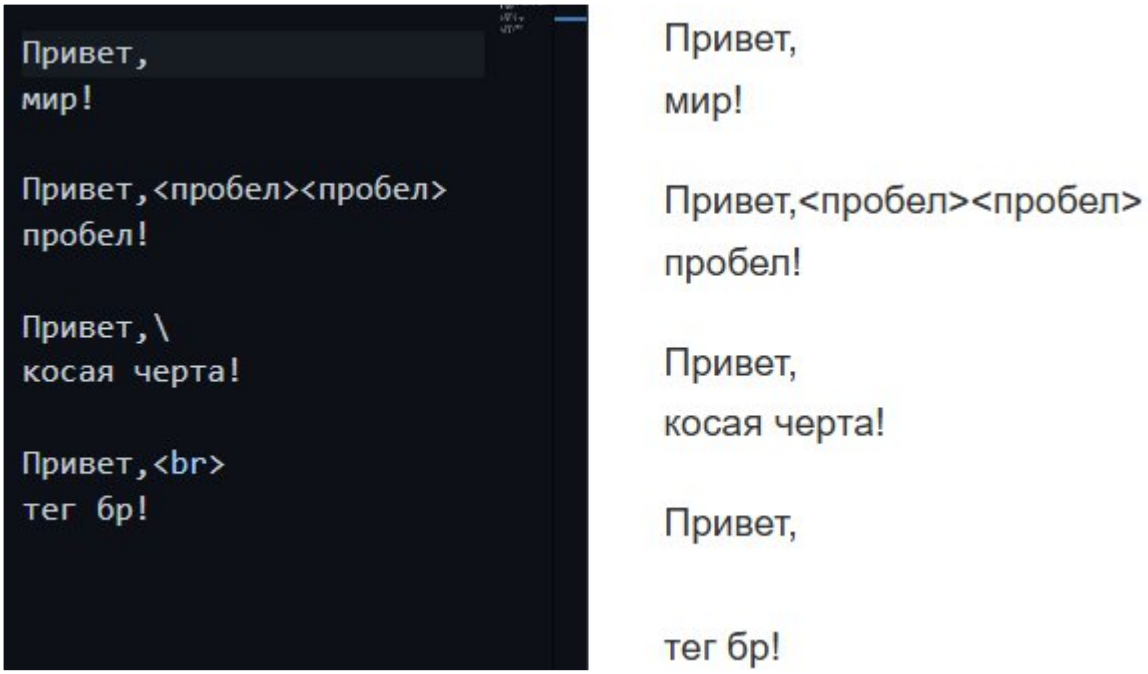


Рисунок 1.Создание параграфа в MD

2.1.2 Заголовки

Markdown предлагает два стиля написания заголовков: через решётки (#) и через подчёркивания (====). Можно использовать до шести уровней заголовков, но подчёркивания позволяют создавать только первые два (<h1> и <h2>).

Для того чтобы выделить заголовок, необходимо поставить от 1 до 6 решёток (#) и пробел в самом начале строки. Уровень заголовка зависит только от количества решёток.

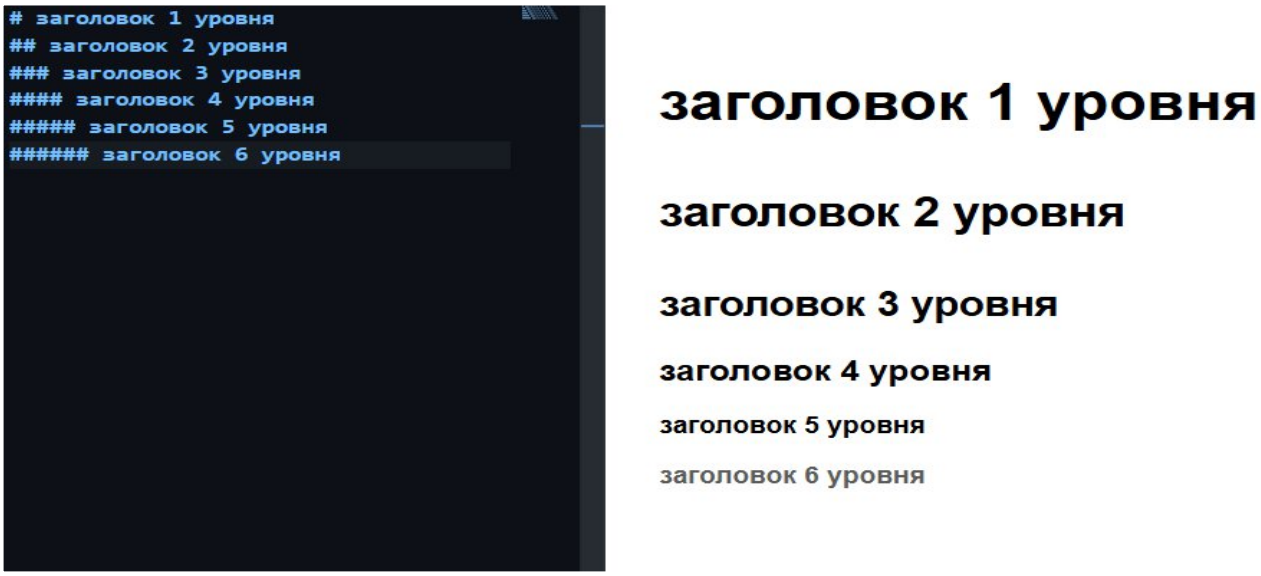


Рисунок 2.Создание заголовков в MD

2.1.3 Выделение текста

Если обернуть текст звёздочками (*) или нижними подчёркиваниями (_), то он станет жирным или курсивным. Оба символа работают одинаково, стиль выделения зависит только от их количества:

• одна пара * или _ сделает текст *курсивным*;
• две пары * или _ сделают текст **жирным**;
• три пары * или _ применят ***оба стиля***.

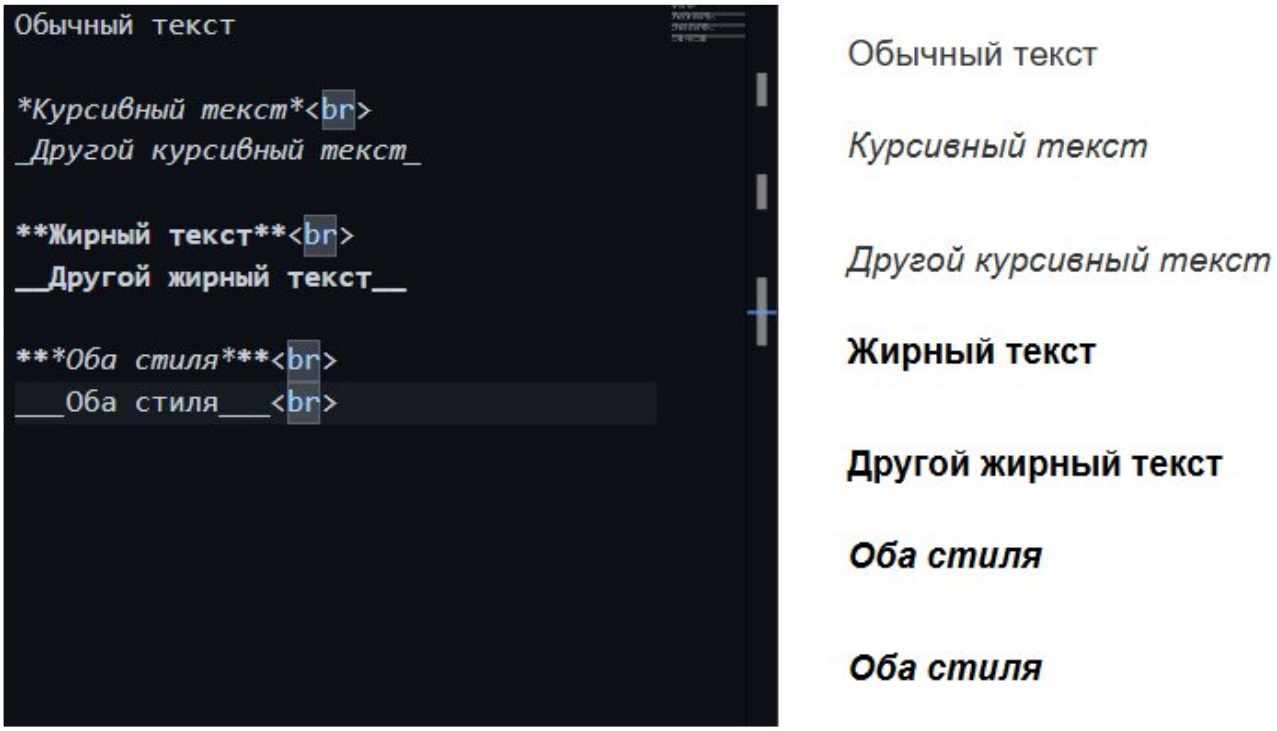


Рисунок 3.Выделение текста в MD

2.1.4 Ссылки

Для вставки ссылки в строчном стиле необходимо воспользоваться следующей конструкцией: [Текст ссылки](URL). Есть возможность добавить подсказку, для этого нужно после URL дописать текст в кавычках: [Текст ссылки](URL "Подсказка").

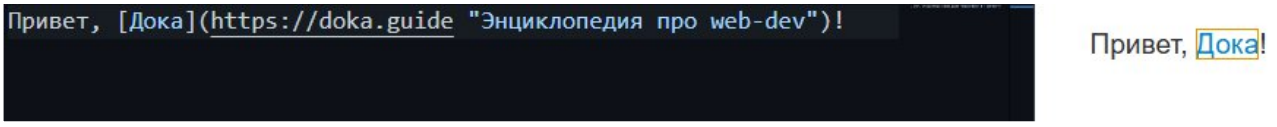


Рисунок 4.Ссылки в MD

2.1.5 Маркированные

Для создания маркированных (ненумерованных) списков перед каждым пунктом нужно поставить минус (-), плюс (+) или звёздочку (*). Маркер и текст пункта необходимо разделять пробелом.

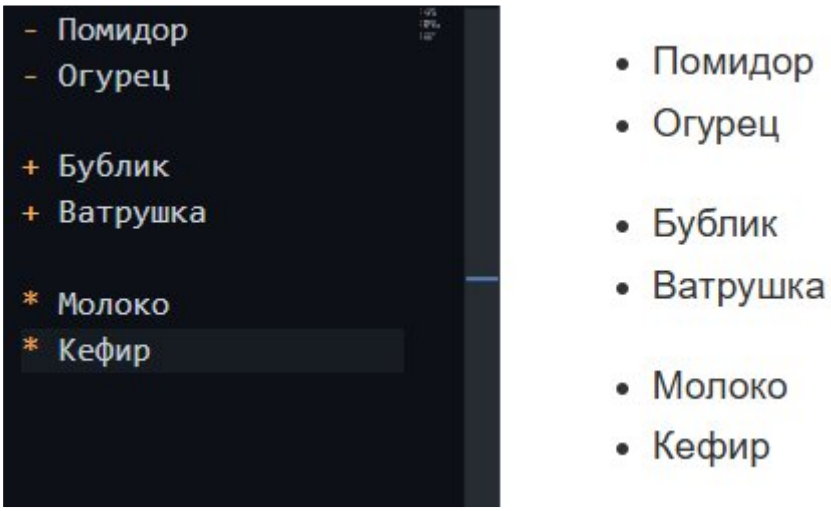


Рисунок 5. Маркировка в MD

Если в качестве маркеров использовать цифры с точкой на конце (1., 2. и т. д.), то мы получим упорядоченный (нумерованный) список.

2.2 Оформление формул в Markdown

Существует два варианта разделителя математических выражений, встроенных в текст. Можно либо окружить выражение символами доллара ($), либо запустить выражение $` и завершить его.`$ Последний синтаксис полезен, если написанное выражение содержит символы, перекрывающиеся синтаксисом markdown.

```
This sentence uses `$` delimiters to show math inline: $\sqrt{3x-1}+(1+x)^2$
```

This sentence uses `$` delimiters to show math inline: $\sqrt{3x-1}+(1+x)^2$

```
This sentence uses $\` and \`$ delimiters to show math inline: $`\sqrt{3x-1}+(1+x)^2`$
```

This sentence uses $` and `$ delimiters to show math inline: $\sqrt{3x-1}+(1+x)^2$

Рисунок 7. Маркировка формул в MD

Чтобы добавить математическое выражение в виде блока, запустите новую строку и разделите выражение двумя символами доллара $$.

```
**The Cauchy-Schwarz Inequality**\
$$\left( \sum_{k=1}^n a_k b_k \right)^2 \leq \left( \sum_{k=1}^n a_k^2 \right) \left(
\sum_{k=1}^n b_k^2 \right)$$
```

The Cauchy-Schwarz Inequality

$$\left(\sum_{k=1}^n a_k b_k \right)^2 \leq \left(\sum_{k=1}^n a_k^2 \right) \left(\sum_{k=1}^n b_k^2 \right)$$

Рисунок 8. Маркировка моделей в MD

Кроме того, для отображения математического выражения в виде блока вы можете использовать синтаксис блока кода ```math. В этом синтаксисе не нужно использовать разделители $$. Следующая отрисовка будет отображаться так же, как и выше:

```
**The Cauchy-Schwarz Inequality**

```math
\left( \sum_{k=1}^n a_k b_k \right)^2 \leq \left( \sum_{k=1}^n a_k^2 \right) \left(
\sum_{k=1}^n b_k^2 \right)
```
```

Рисунок 8. Альтернативная маркировка моделей в MD

2.3. Оформление изображений в Markdown

Существует два основных метода вставки изображений в Markdown: использование локальных изображений и внешних ссылок.

2.3.1 Локальные изображения

Для вставки изображений, хранящихся локально, важно правильно указать путь к файлу. Рекомендуется размещать изображения либо в той же директории, где находится Markdown-файл, либо на одном уровне с ним.Если изображение в той же директории, что и md-файл, достаточно указать его имя:

![Компьютер](computer.png)

• Текст в квадратных скобках — альтернативный текст (alt-текст). Он отображается, если изображение не загружается, и помогает программам для чтения с экрана описать изображение для пользователей с нарушениями зрения.
• Путь к изображению указывается в круглых скобках. Важно следить за корректностью указания пути, чтобы избежать проблем с отображением после загрузки на сервер.

2.3.2 Внешние изображение

Если вам нужно вставить изображение, размещенное в интернете, используйте URL:

![Описание изображения](https://site/foto)

Преимущества использования внешних изображений:

• Экономия места в репозитории.
• Простое управление содержимым, особенно если изображение часто обновляется. Недостатки:
• Зависимость от внешнего источника. Если изображение удалено или URL изменился, оно перестанет отображаться.

Выполнения лабораторной работы

В рамках основной части работы мне нужно было освоить базовые команды для работы с git, научиться компилировать шаблон отчета из формата .md в .pdf и .docx с помощью make, а также изучить структуру исходного файла отчета. Сначала я открыл терминал и перешел в рабочий каталог курса с помощью команды cd ~/work/study/2025-2026/"Архитектура компьютера"/arch-pc/. После этого я обновил свой локальный репозиторий, чтобы загрузить все последние изменения с сервера. Для этого я использовал команду git pull. Терминал сообщил, что все уже обновлено. Это значит, что я ничего не пропустил.

```
┌──(rzemcov㉿rzemcov)-[~]
└─$ cd ~/work/study/2025-2026/"Архитектура компьютера"/study_arch-pc/

┌──(rzemcov㉿rzemcov)-[~/…/study/2025-2026/Архитектура компьютера/study_arch-pc]
└─$ git pull
Already up to date.
```

Далее я перешел в каталог с шаблоном отчета для этой лабораторной работы: cd labs/lab03/report. Находясь в этой папке, я запустил компиляцию отчета командой make. Процесс прошел успешно, и в каталоге появились два новых файла: report.pdf и report.docx. Я открыл оба, чтобы убедиться, что они выглядят корректно. Мне понравилось, как одна простая команда создает документы сразу в двух форматах.

```
┌──(rzemcov㉿rzemcov)-[~/…/study_arch-pc/labs/lab03/report]
└─$ make
pandoc
  to: latex
  output-file: arch-pc--lab03--report.tex
  standalone: true
  self-contained: true
  pdf-engine: xelatex
  variables:
    graphics: true
    tables: true
  default-image-extension: pdf
  number-sections: true
  toc: true
  toc-depth: 2
  cite-method: biblatex

metadata
  documentclass: scrreprt
  classoption:
    - DIV=11
    - numbers=noendperiod
  papersize: a4
  header-includes:
    - '\KOMAoption{captions}{tableheading}'
    - |
      \usepackage{indentfirst}
      \usepackage{float}
      \floatplacement{figure}{H}
      \usepackage{libertine}
  block-headings: true
  lang: ru-RU
  toc-title: Содержание
  crossref:
    lof-title: Список иллюстраций
    lot-title: Список таблиц
    lol-title: Листинги
  bibliography:
    - bib/cite.bib
  csl: _resources/csl/gost-r-7-0-5-2008-numeric.csl
  colorlinks: false
  lof: true
  lot: true
```

После проверки скомпилированных файлов я удалил их командой make clean. Это полезная команда, чтобы быстро убрать "мусор" и оставить в папке только исходные файлы. Я проверил, что файлы report.pdf и report.docx действительно были удалены.

```
┌──(rzemcov㉿rzemcov)-[~/…/study_arch-pc/labs/lab03/report]
└─$ cd /home/rzemcov/work/study/2025-2026/"Архитектура компьютера"/study_arch-pc/labs/lab01/report/

┌──(rzemcov㉿rzemcov)-[~/…/study_arch-pc/labs/lab01/report]
└─$ git status
```

Я открыл файл report.md в текстовом редакторе gedit, чтобы посмотреть, как он устроен.

```
1 1.Банан
2 2.Курица
3 3.Арбуз
```

Когда отчет был готов, я вернулся в корневой каталог проекта. Чтобы сохранить все изменения на сервере, я последовательно выполнил три команды:

- git add . — подготовил все измененные файлы к отправке.
- git commit -am 'feat(main): add files lab-3' — зафиксировал изменения с комментарием.
- git push — отправил все на Github. Все прошло без ошибок, и теперь моя работа надежно сохранена.

Выполнения заданий для самостоятельной работы

В качестве самостоятельной работы мне нужно было оформить отчет по лабораторной работе №2 в формате Markdown. Затем скомпилировать его в pdf и docx и загрузить все три файла на Github.

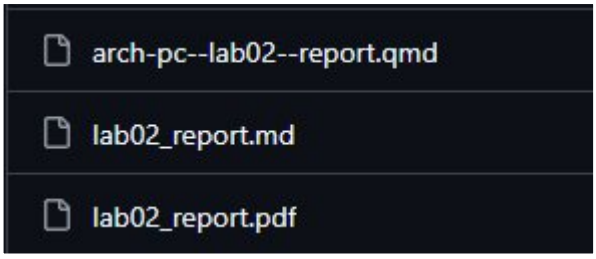
```
┌──(rzemcov㉿rzemcov)-[~]
└─$ cd ~/work/study/2025-2026/"Архитектура компьютера"/study_arch-pc/

┌──(rzemcov㉿rzemcov)-[~/…/study/2025-2026/Архитектура компьютера/study_arch-pc]
└─$ git add .
warning: in the working copy of 'labs/lab02/report/lab02_report.md', CRLF will be replaced by LF the next time Git
touches it

┌──(rzemcov㉿rzemcov)-[~/…/study/2025-2026/Архитектура компьютера/study_arch-pc]
└─$ git commit -am 'Добавление пдф,докс,мд'
[master 481c063] Добавление пдф,докс,мд
 5 files changed, 628 insertions(+)
 create mode 100644 labs/lab02/report/lab02_report.md
 create mode 100644 labs/lab03/report/arch-pc--lab03--report.tex
 create mode 100644 labs/lab03/report/report.docx
 create mode 100644 labs/lab03/report/report.md
 create mode 100644 labs/lab03/report/report.pdf

┌──(rzemcov㉿rzemcov)-[~/…/study/2025-2026/Архитектура компьютера/study_arch-pc]
└─$ git push
Enumerating objects: 16, done.
Counting objects: 100% (16/16), done.
Compressing objects: 100% (9/9), done.
Writing objects: 100% (10/10), 11.16 KiB | 11.16 MiB/s, done.
Total 10 (delta 4), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (4/4), completed with 4 local objects.
To github.com:zemsovroman-a11y/study_arch-pc.git
   3405d51..481c063  master → master
```

Все файлы (.md, .pdf, .docx) я добавил в репозиторий и отправил на Github, как и в основной части работы.



Вывод

В ходе выполнения данной лабораторной работы я достиг поставленной цели: освоил процедуру оформления отчетов с помощью языка разметки Markdown. Я научился использовать его базовый синтаксис для форматирования текста, создания списков и вставки изображений. Кроме того, я получил практический опыт использования утилиты make для автоматической компиляции документов в форматы .pdf и .docx, а также закрепил навыки работы с системой контроля версий git для сохранения и отправки своей работы в удаленный репозитории

Список литературы

Язык разметки MarkdownЗачем нужен ещё один язык разметки и как на нём писать.- https://doka.guide/tools/markdown/

Написание математических выражений. Используйте Markdown для отображения математических выражений на GitHub.- https://docs.github.com/ru/getstarted/writing-on-github/working-with-advanced-formatting/writing-mathematicalexpressions

Как добавлять изображения в Markdown - https://timeweb.cloud/tutorials/csshtml/kak-dobavlyat-izobrazheniya-v-markdown