## 0.1 Front matter

title: “Лабораторная работа № 3. Markdown”

subtitle: “Простейший вариант”

author: “Гусейнов Георгий Русланович”

## 0.2 Generic otions

lang: ru-RU

toc-title: “Содержание”

## 0.3 Bibliography

bibliography: bib/cite.bib

csl: pandoc/csl/gost-r-7-0-5-2008-numeric.csl

## 0.4 Pdf output format

toc: true # Table of contents

toc-depth: 2

lof: true # List of figures

lot: true # List of tables

fontsize: 12pt

linestretch: 1.5

papersize: a4

documentclass: scrreprt

## 0.5 I18n polyglossia

polyglossia-lang:

name: russian

options:

- spelling=modern  
  
- babelshorthands=true

polyglossia-otherlangs:

name: english

## 0.6 I18n babel

babel-lang: russian

babel-otherlangs: english

## 0.7 Fonts

mainfont: PT Serif

romanfont: PT Serif

sansfont: PT Sans

monofont: PT Mono

mainfontoptions: Ligatures=TeX

romanfontoptions: Ligatures=TeX

sansfontoptions: Ligatures=TeX,Scale=MatchLowercase

monofontoptions: Scale=MatchLowercase,Scale=0.9

## 0.8 Biblatex

biblatex: true

biblio-style: “gost-numeric”

biblatexoptions:

* parentracker=true
* backend=biber
* hyperref=auto
* language=auto
* autolang=other\*
* citestyle=gost-numeric

## 0.9 Pandoc-crossref LaTeX customization

figureTitle: “Рис.”

tableTitle: “Таблица”

listingTitle: “Листинг”

lofTitle: “Список иллюстраций”

lotTitle: “Список таблиц”

lolTitle: “Листинги”

## 0.10 Misc options

indent: true

header-includes:

# keep figures where there are in the text

* # keep figures where there are in the text

# 1 Цель работы

Научиться оформлять отчёты с помощью легковесного языка разметки Markdown.

# 2 Задание

* Сделайте отчёт по предыдущей лабораторной работе в формате Markdown.
* В качестве отчёта просьба предоставить отчёты в 3 форматах: pdf, docx и md

# 3 Выполнение лабораторной работы

### 3.0.1 Установил git

* Установим *git*:
* dnf install git

### 3.0.2 Установил gh

* Fedora:
* dnf install gh

## 3.1 Установил базовае настройка git

* Задал имя и email репозитория:
* git config --global user.name "Name Surname"  
  git config --global user.email "work@mail"
* Настроил utf-8 в выводе сообщений git:
* git config --global core.quotepath false
* Настроил верификацию и подписание коммитов git.
* Зададал имя начальной ветки (будем называть её master):
* git config --global init.defaultBranch master
* Параметр autocrlf:
* git config --global core.autocrlf input
* Параметр safecrlf:
* git config --global core.safecrlf warn
* image\_1](./image/Снимок экрана от 2023-02-25 14-36-13.png)

## 3.2 Создал ключи *pgp*

* Сгенерировал ключь
* gpg --full-generate-key
* Из предложенных опций выбирал:
  + тип *RSA and RSA*;
  + размер 4096;
  + выбрал срок действия; значение по умолчанию — 0 (срок действия не истекает никогда).
* GPG запросит личную информацию, которая сохранится в ключе:
* Ввел:
  + Имя (не менее 5 символов).
  + Адрес электронной почты.
    - При вводе email убедитесь, что он соответствует адресу, используемому на GitHub.
  + Комментарий. Можно ввести что угодно или нажать клавишу ввода, чтобы оставить это поле пустым.
  + image\_2](./image/Снимок экрана от 2023-02-25 14-37-34.png)

## 3.3 Настроил github

image\_3](./image/Снимок экрана от 2023-02-25 14-41-05.png)

## 3.4 Добавил PGP ключь в GitHub

* Выводим список ключей и копируем отпечаток приватного ключа:
* gpg --list-secret-keys --keyid-format LONG
* Отпечаток ключа — это последовательность байтов, используемая для идентификации более длинного, по сравнению с самим отпечатком ключа.
* Формат строки:
* sec Алгоритм/Отпечаток\_ключа Дата\_создания [Флаги] [Годен\_до]  
   ID\_ключа
* Cкопировал сгенерированный PGP ключ в буфер обмена:
* gpg --armor --export <PGP Fingerprint> | xclip -sel clip
* Перешел в настройки GitHub (<https://github.com/settings/keys>), нажал на кнопку *New GPG key* и вставил полученный ключ в поле ввода. image\_4](./image/Снимок экрана от 2023-02-25 14-53-58.png)

## 3.5 Настройка автоматических подписей коммитов git

* Используя введёный email, указал Git применять его при подписи коммитов:
* git config --global user.signingkey <PGP Fingerprint>  
  git config --global commit.gpgsign true  
  git config --global gpg.program $(which gpg2)
* image\_5](./image/Снимок экрана от 2023-02-25 17-25-16.png)

## 3.6 Настройка gh

* Для начала необходимо авторизоваться
* gh auth login
* Утилита задаст несколько наводящих вопросов.
* Авторизоваться можно через броузер.

# 4 Выводы

Научился настраивать и работать с git.