

# ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ

## № 3

Markdown

Коняева Марина Александровна

# Содержание

Цель работы	3
Теоретическое введение	4
Выполнение лабораторной работы	5
Выводы	8
Контрольные вопросы	9

## Цель работы

Научиться оформлять отчёты с помощью легковесного языка разметки Markdown.

# Теоретическое введение

Markdown — язык текстовой разметки, созданный писателем и блогером Джоном Грубером. Он предназначен для создания красиво оформленных текстов в обычных файлах формата ТХТ. Вам не нужны громоздкие процессоры вроде Word или Pages, чтобы создавать документы с жирным или курсивным начертанием, цитатами, ссылками и даже таблицами. Достаточно запомнить простые правила Markdown, и можно писать хоть в «Блокноте». Хотя специализированные Markdown-редакторы, конечно, намного удобнее.

# Выполнение лабораторной работы

## 1. Запишем отчет по лабораторной работе 2 в markdown file

```
tableTitle: "Таблица"
listingTitle: "Листинг"
lolTitle: "Листинги"
## Misc options
indent: true
header-includes:
- \usepackage[indentfirst]
- \usepackage{float} # keep figures where there are in the text
- \floatplacement{figure}{H} # keep figures where there are in the text
---

# Цель работы

Изучить идеологию и применение средств контроля версий, освоить умения по работе с git.

# Теоретическое введение

Markdown – язык текстовой разметки, созданный писателем и блогером Джоном Грубером. Он предназначен для создания красиво оформленных текстов в обычных файлах формата TXT. Вам не нужны громоздкие процессоры вроде Word или Pages, чтобы создавать документы с жирным или курсивным начертанием, цитатами, ссылками и даже таблицами. Достаточно запомнить простые правила Markdown, и можно писать хоть в «Блокноте». Хотя специализированные Markdown-редакторы, конечно, намного удобнее.

# Выполнение лабораторной работы

1. Заведом учетную запись на сайте github.com, свой аккаунт я создала ранее, введя необходимые данные

2. Скачивание git-flow и выдача прав на выполнение

![[скачивание git-flow]](image/70e7a070e787d1817d18370e8070e78e70e78a1.png){ #fig:001 width=100% }
*Изображение 2.1 Скачивание git-flow*

![[выдача прав на выполнение]](image/70e7a070e787d1817d18370e8070e78e70e78a2.png){ #fig:001 width=100% }
*Изображение 2.2 выдача прав на выполнение*

3. Установка git-flow

![[установка git-flow]](image/70e7a070e787d1817d18370e8070e78e70e78a3.png){ #fig:001 width=100% }
```

Изображение 1.1 отчет в markdown

## 2. Затем сконвертируем файл в форматах pdf и docx

## Выполнение лабораторной работы

1. Заведём учетную запись на сайте github.com, свой аккаунт я создала ранее, введи необходимые данные
2. Скачивание git-flow и выдача прав на выполнение

```
makonyaeva@makonyaeva tmp1$ curl --fail -O https://raw.githubusercontent.com/gitflow-developer/gitflow-installer.sh
gitflow-installer.sh
gitflow-installer.sh 1
makonyaeva@makonyaeva tmp1$ cat gitflow-installer.sh
#!/bin/sh
set -e
# git-flow no-make installer
# Installing git-flow to /usr/local/bin
# Cloning repo from GitHub to gitflow
# Клонирование в «gitflow»
remote: Enumerating objects: 4270, done.
remote: Total 4270 (delta 0), reused 0 (delta 0), pack-reused 4270
```

Изображение 2.1 Скачивание git-flow

```
[makonyaeva@makonyaeva tmp]$ chmod +x gitflow-installer.sh
[makonyaeva@makonyaeva tmp]$
```

Изображение 2.2 выдача прав на выполнение

3. Установка git-flow

```
[makonyaeva@makonyaeva tmp]$ sudo ./gitflow-installer.sh install stable
```

Изображение 3.1 Установка git-flow

```
[sudo] пароль для makonyaeva:
## git-flow no-make installer ##
Installing git-flow to /usr/local/bin
Cloning repo from GitHub to gitflow
Клонирование в «gitflow»
remote: Enumerating objects: 4270, done.
remote: Total 4270 (delta 0), reused 0 (delta 0), pack-reused 4270
```

Изображение 2.1 сконвертируем файл в форматах

## Выполнение лабораторной работы

1. Заведём учетную запись на сайте [github.com](https://github.com), свой аккаунт я создала ранее, введя необходимые данные
2. Скачивание [git-flow](#) и выдача прав на выполнение

```
makonyaeva@makonyaeva tmp]$ curl -o- -L https://raw.githubusercontent.com/gitflow-developer/gitflow/master/gitflow-installer.sh
gitflow-installer.sh
gitflow-installer.sh 1s
makonyaeva@makonyaeva tmp]$
```

Изображение 2.1 Скачивание [git-flow](#)

```
[makonyaeva@makonyaeva tmp]$ chmod +x gitflow-installer.sh
[makonyaeva@makonyaeva tmp]$
```

Изображение 2.2 выдача прав на выполнение

3. Установка [git-flow](#)

```
[makonyaeva@makonyaeva tmp]$ sudo ./gitflow-installer.sh install stable
```

Изображение 3.1 Установка [git-flow](#)

```
[sudo] пароль для makonyaeva:
### git-flow no-make installer ###
Installing git-flow to /usr/local/bin
Cloning repo from GitHub to gitflow
Клонирование в «gitflow»...
remote: Enumerating objects: 4270, done.
remote: Total 4270 (delta 0), reused 0 (delta 0), pack-reused 4270
Получение объектов: 100% (4270/4270), 1.74 Миб | 3.06 Миб/с, готово.
Определение изменений: 100% (2533/2533), готово.
```

Изображение 3.2 Установка [git-flow](#)

4. Установка пакета [git](#)

```
makonyaeva@makonyaeva:~$ sudo apt-get install git
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
git is already the default version (1:2.34.3-1ubuntu1) on this system.
The following packages were automatically installed and are no longer required:
  libcurl4-openssl-dev libexpat1-dev libffi-dev libgdbm-dev libgdbm6
  libldap2-dev libncurses6 libperl5.34 libpython3.10-dev libpython3.10
  libpython3.10-stdlib libpython3.11-dev libpython3.11-stdlib libpython3
  libpython3.11-stdlib libpython3.11-stdlib libpython3.11-stdlib libpython3.11-stdlib
  python3-pip python3-setuptools python3-wheel
Use 'dpkg --get-selections --purge' to remove these packages.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

Изображение 4.1 Установка [git](#)

5. Базовая настройка [git](#), ветки, [autocrlf](#) и [safecr](#)

```
[makonyaeva@makonyaeva tmp]$ git config --global user.name "makonyaeva"
[makonyaeva@makonyaeva tmp]$ git config --global user.email "makonyaeva@yandex.ru"
[makonyaeva@makonyaeva tmp]$ git config --global core.autocrlf input
[makonyaeva@makonyaeva tmp]$ git config --global core.safecrlf true
```

Изображение 5.1 Базовая настройка [git](#)

6. Создание [ssh](#) ключей с ;

Изображение 2.2 сконвертируем файл в форматах

3. Проверим читабельное оформление отчета в разных форматах

# Выводы

В ходе данной лабораторной работы научились оформлять отчеты с помощью легковесного языка разметки Markdown.



# Контрольные вопросы

Контрольные вопросы отсутствуют в лабораторной работе 3