

# Лабораторная работа №3

Журавлев Георгий Иванович

## Цель работы

- Сделать отчет по предыдущей работе в формате Markdown.
- Предоставить в 3-х форматах: pdf, md and docx.

# Выполнение лабораторной работы

## 1. Заполняем отчет в формате Markdown

```
lang: ru-RU
title: "Лабораторная работа №2"
author: "Хуравлев Георгий Иванович"
...

# Цель работы
Получить опыт работы с Git. Создать аккаунт; подключить репозиторий к Github; Пройти первичную конфигурацию; провести конфигурацию git-flow.

# Задание
* Сделать отчет по предыдущей работе в формате Markdown.
* Предоставить в 3-х форматах: pdf, md and docx.
```

---

```
# Выполнение лабораторной работы

### 1. Создаем аккаунт github

![1scrsht](scrsht/1.jpg)
### 2. Генерируем ключ для настройки VCS.
![2scrsht](scrsht/2.jpg)
### 3. Добавляем сгенерированный ключ.
![3scrsht](scrsht/3.jpg)
### 4. Приступаем к созданию репозитория и файлов.
![4scrsht](scrsht/4.jpg)
### 5. Добавляем лицензию
![5scrsht](scrsht/5.jpg)
### 6. Добавляем игнорируемый файл ( с помощью vs code).
![6scrsht](scrsht/6.jpg)
### 7. Инициализируем git flow.
![7scrsht](scrsht/7.jpg)
### 8. Добавляем релиз и загружаем на Github.
![8scrsht](scrsht/8.jpg)
![9scrsht](scrsht/9.jpg)
![10scrsht](scrsht/10.jpg)
### 9. Получившиеся результаты(ветви + данные).
![11scrsht](scrsht/11.jpg)
```

```
# Вывод

Благодаря этой лабораторной работе, я научился: создавать репозитории, создавать файлы и проходить процесс подтверждения, загружать файлы в репозиторий git, делать релизы и создавать ветви.

# Контрольные вопросы

1. VCS – ПО для облегчения работы с изменяющейся информацией(хранит изменяющиеся версии; может быть изменена разными людьми, если проходит работа над совместным проектом). VCS предназначена для удобства использования в проектах, например один человек написал некоторый код, который хочет поместить в основную ветвь проекта, его коллеги могут проверить и сделать заметки в виде комментариев.

2. Хранилище – основное место хранения; Commit – подтверждение всех изменений и тп., для дальнейшей загрузки в репозиторий; History – история изменения; Рабочая копия – действительные файлы с файлами.

3. Централизованные системы контроля версий представляют собой приложения типа клиент-сервер:( 1 основной репозиторий)(SVN)
Распределенные системы контроля версий позволяют хранить репозиторий (его копии) у каждого разработчика, работающего с данной системой.( множество репозиториях, взаимодействующих с сервером)(Git)

4. Создание репозитория -> создание/добавление файлов в локальную файловую систему -> добавление в локальный репозиторий -> подтверждение добавления -> загрузка в онлайн репозиторий.

5. -
6. -

7. Git add – adds a file to the staging area; git commit- opens chosen git editor or git commit -m "message"(commitment with a message); git push – pushes files to repository; git pull – downloads/ changes files from repository; git branch – shows branches; git status – status of your process; git rm ""- removes a file/es; and so on.

8. Создание файла в репозитории( лока.); загрузка файлов на удалённый репозиторий(удал.)

9. Ветки – специальные разделения дерева, которые нужны для удобства использования git, например для дальнейшего merge'a.

10. "git editor".gitignore --git add .gitignore->git commit; Игнорирование нужно для исключения ненужных файлов/ файловых систем/ тп. из области работы.
```

## 2. Конвертируем отчет в форматы pdf and docx

```
g_zhuravlev@g ~:/github/qzhuravlev.pfur/2020-2021/05/labs/lab02  main  pandoc report.md -o report.docx
g_zhuravlev@g ~:/github/qzhuravlev.pfur/2020-2021/05/labs/lab02  main  pandoc report.md -o report.pdf
```

## Вывод

Благодаря этой лабораторной работе, я научился: работать с markdown, конвертировать файлы в другие форматы.