

# Шаблон отчёта по 2 лабораторной работе

Яссин мохамад аламин

## Отчёт о выполнении лабораторной работы №2

### Цель работы

Изучить идеологию и применение средств контроля версий. Научиться использовать систему контроля версии в своих проектах

### Задание

Настроить git на своем компьютере, инициализировать локальный репозиторий в папке с лабораторными работами, добавить все нужные файлы в репозиторий, подключить удаленный репозиторий с github, связать локальный и удаленные репозитории

### Выполнение лабораторной работы

- На моем личном компьютере система Git и Gitflow уже их установил
- Создание ssh для подключения к репо без записи имени пользователя и пароля каждый раз ssh key
- добавление безопасного ключа на github git-ssh key
- создание нового репозитория New repo
- добавление лицензии Licence
- добавление файла git-ignore git-ignore
- закинуть файл в гитхаб uploading files
- используя git-flow using gitflow
- добавление файлов и завершение релиза Release
- отправка файлов в GitHub Github push
- Файлы на github Files

## Выводы

В этой лабораторной работе я научился использовать Git, и подключать удаленные репозитории, добавлять и удалять необходимые файлы. Я также научился использовать Git Flow, который значительно упрощает разработку проекта и навигацию между ветвями.

## Контрольные вопросы

1. Системы контроля версий -VCS- это программное обеспечение, которое используется для облегчения работы с изменяющейся информацией, обычно - в проектах. Чаще всего используется при разработке, когда над одним проектом работает большое количество людей. 2.
  - Репозиторий в системе контроля версий - Это удаленный репозиторий, в котором хранятся все файлы проекта
  - commit - фиксирует изменения перед загрузкой файлов в систему контроля версий
  - история хранит все изменения в проекте, и при необходимости позволяет откатиться в желаемое место
  - рабочая копия - - это копия проекта на компьютере разработчика. Если другой член команды изменил проект, вам необходимо скачать новую версию проекта на свой компьютер.
3. В отличие от классических, в распределённых системах контроля версий центральный репозиторий не является обязательным. Среди классических VCS наиболее известны CVS, Subversion, а среди распределённых — Git, Bazaar, Mercurial. Принципы их работы схожи, отличаются они в основном синтаксисом используемых в работе команд. В децентрализованных системах у каждого из участников проекта есть полная копия проекта на своем компьютере, что делает его менее зависимым от сервера (Git).
4. Для начала необходимо создать и подключить удаленный репозиторий. Затем, поскольку никто, кроме вас, не изменяет проект, по мере изменения проекта отправляйте изменения на сервер, и нет необходимости загружать изменения.
5. Участник проекта (пользователь) перед началом работы посредством определённых команд получает нужную ему версию файлов. После внесения изменений, пользователь размещает новую версию в хранилище. При этом предыдущие версии не удаляются из центрального хранилища и к ним можно вернуться в любой момент.
6. Упрощение обмена информацией, ускорение разработки, устранение ошибок и недочетов при разработке. 7.
  - git init - инициализирует локальный репозиторий
  - git add \* или add. - добавляет файлы в репозиторий
  - git commit - версия фиксации
  - git pull - загружает текущую версию проекта

- `git push` - отправляет измененный проект на сервер
  - `git checkout` - позволяет переключаться между ветками
  - `git status` - текущий статус проекта
  - `git branch` - просмотреть доступные ветки
  - `git remote add` - добавить удаленный репозиторий
8. `git push --all` (`push origin master/любой branch`)
  9. Ветви функций, также иногда называемые ветвями тем, используются для разработки новых функций, которые должны появиться в текущих или будущих выпусках.
  10. Существуют временные и системные файлы, которые загромождают проект и не нужны. путь к ним можно добавить в файл `.gitignore`, тогда они не будут добавлены в проект