**Задание 2.1**

Текстовый отчет по работе с демонстрацией выполненных действий

и комментариями по выполнению

**Вариант 4.** Терминал (Terminal) или Командная строка (или Command Shell Git)

1. **git version -** показывает версию git



1. **git config –global user.name “имя”** - указывает имя пользователя



1. **git config –global user.email “почта”** - указывает почту пользователя



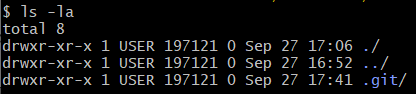
1. **cd + путь, в котором будет находиться репозиторий** - указывает папку в которой будет находиться репозиторий на диске



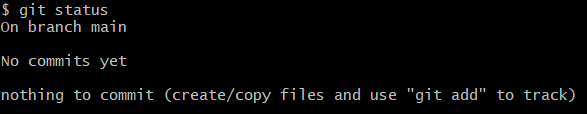
1. **git init -** инициализирует репозиторий



1. **ls -la -** проверка создания репозитория



1. **git status -** показывает статус репозитория



1. **touch + имя файла** - создает файл в репозитории

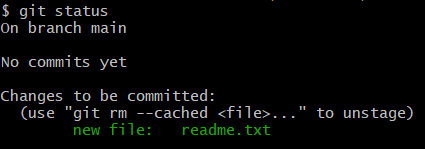


1. **nano + имя файла** - редактирование файла

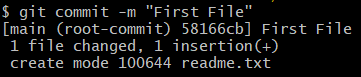




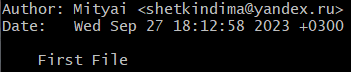
1. **git add + имя файла** - отслеживает файл



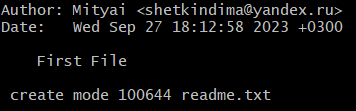
1. **git commit -m “First File”** - фиксирует статус файла



1. **git log** - показывает историю коммитов



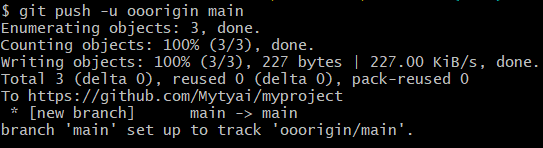
1. **git log –summary** - показывает более подробную историю коммитов



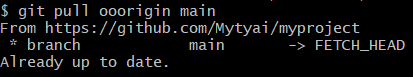
1. **git remote add + название удаленного репозитория + ссылка на репозиторий** - добавление удаленного репозитория



1. **git push -u + название удаленного репозитория + текущая ветка** - выгружает файлы в удаленный репозиторий



1. **git pull** **+ название удаленного репозитория + текущая ветка -** извлекает и интегрирует с репозиторием



1. **mkdir + название каталога** - создает каталог(папку)



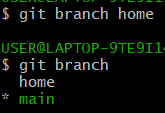
1. **git reset** - сбросить текущее отслеживание



1. **git checkout – ‘файл’** - возвращает файл к изначальному состоянию



1. **git branch + название ветки** - добавляет ветку



1. **git checkout + название ветки** - меняет ветку



1. **git rm + название файла** - удаляет отслеживаемого файла



1. **git branch -d + название ветки** - удаляет ветку

