**--global** – глобальная настройка

**Типы вывода**

**-l** – тип вывода список

**-m** – следующая строка – вводные данные

**--all** или **.** - все

**Commands**

**Mkdir** – создать папку

**Cd** – переход в другую папку ( change directory )

**Ls –** вывести список папок

Перейти в нужную папку написать название программы пробел название файла и одно откроет второе

**Rm …** - удалить … в выбранной папке

**git config --global user.name "Boris\_age"** – установка имени

**git config –list** – список настроек

**git init** – создать репозиторий

**git status** – вывод состояние объектов в выбранной папке, отслеживаемые или нет

**git add …** - добавить файл к списку для обновления

**git commit –m “…”** – сохранить обновления в выбранной папке с комментарием

**git push** – добавить в github

**git commit --amend –m “…”** – обновить прошлое сохранение обновления в выбранной папке с комментарием

**git reset …** - убрать файл из выбора для обновления ( кидает в не отслеживаемые )

**git checkout -- …** - отменить изменения в файле (если файл в stage области, тб выбран для сохранения, то его сначала нужно от туда исключить, … - название отдельного файла или точка для всех)

создать файл **.gitignore** и записать в него все название папок и файлов, которые не нужно включать в сохранения, можно использовать регулярные выражения

**git commit --amend --no-edit** – все текущие выбранные файлы будут запомнены, как исключения для сохранений, они перестану отображаться в списке доступных

**git checkout hashed-id** – вернуться к прошлому коммиту

**git log** – все изменения

**git branch …** - создать ветку под именем …

**git checkout …** - переключиться на ветку …

**git branch –va** – вывести все ветки

**git stash** – сохранить в репозиторий ( при переключении на другу ветку, тебе может не хотеться сохранять проект, тогда его можно закинуть в буфер этой командой )

**git stash list** – список буферных данных

**git stash pop** – освобождает буфер, не важно в какую ветку, просто переписывает сохраненные данные

**git merge …** - выполнить слияние веток ( правильно нужно вызываться от главного корня проекта )

**git merge –abort** – отменить слияние

**git remote add имя ссылка** – соединение с удаленным репозиторием

**git remote –v** – проверка наличия соединения

**git remote remove имя** - удалить удаленный репозиторий

**git clone ссылка** - скопировать файлы из удаленного репозитория в выбранную папку

**git fetch** … - обновить данные из онлайн репозитория + **git pull** – вставить их