**МИНИCTEPCTBO НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное образовательное учреждение высшего образования**

**«Северо-Кавказский федеральный университет» Кафедра инфокоммуникаций**

**Отчет по лабораторной работе №2**

**по дисциплине «Основы программной инженерии»**

|  |
| --- |
| Выполнил студент группы ПИЖ-б-о-20-1 |
| Ваньянц И.М. « » 20 г. |
| Подпись студента |
| Работа защищена « » \_20 г. |
| Проверил Воронкин Р.А.  (подпись) |

Ставрополь 2021

# ХОД РАБОТЫ

# 

# Рисунок 1 – просмотр истории коммитов

# 

# Рисунок 2 – просмотр истории коммитов репозитория simplegit-progit

# 

# Рисунок 3 – использование git log с параметрами –p и -2

# 

# Рисунок 4 – использование git log с параметром –stat

# 

# Рисунок 5 – использование git log с измененным форматом

# 

# Рисунок 6 – просмотр истории коммитов с опцией –graph

# 

# Рисунок 7 – пример отмены индексации файла

# 

# Рисунок 8 – пример отката изменений

# 

# Рисунок 9 – пример просмотра путей удаленного репозитория

# 

# Рисунок 10 – пример добавления удалённых репозиториев

# 

# Рисунок 11 – отправка изменений на удалённый репозиторий

# 

# Рисунок 12 – просмотр удалённого репозитория

# 

# Рисунок 13 – удаление и переименование удаленного репозитория

# 

# Рисунок 14 – просмотр данных тега с коммитом

# 

# Рисунок 15 – удаление тега

# 

Рисунок 16 – создание зеркала gitlab на github

# КОНТРОЛЬНЫЕ ВОПРОСЫ

# Git log показывает все ваши коммиты. –p или –patch показывают разницу в коммитах. –n где n – это целое число, позволяет ограничить количество выводимых записей. –stat показывает сокращенную статистику коммитов. –pretty позволяет изменять формат вывода коммитов, например git log –pretty = oneline будет выводить каждый коммит в одной строке. Опция format помогает вывести информацию о коммитах так, как вам нужно, при помощи опций: %H –хеш коммита, %h – сокращенный хеш коммита, %T – хеш дерева, %t – сокращенный хеш дерева, %P – хеш родителей, %p – сокращенный хеш родителей, %an – имя автора, %ae – электронная почта автора, %ad – дата автора, %ar – относительная дата автора, %cn – имя коммитера, %ce – электронная почта коммитера, %cd – дата коммитера, %cr – относительная дата коммитера, %s – содержание.

# Git log –since=n.weeks где n – это целое число, ограничивает вывод коммитов в пределах n месяцев, дней, лет, минут и т.д. Опция –s принимает аргумент в виде строки и показывает только те коммиты, в которых изменение в коде повлекло за собой добавление или удаление этой строки. Также, если указать путь с git log, то вывод ограничивается только теми коммитами, в которых были изменения этих файлов.

# Для того, чтобы сделать изменения в уже осуществленном коммите существует опция –amend – она отменяет последний осуществленный коммит, и вы можете внести изменения в него заново.

# Git add добавляет файлы в индекс коммита, git reset HEAD имя\_файла же отменяет данную индексацию, и вы можете, например, внести изменения по отдельности в каждый коммит.

# Чтобы отменить изменения в файле, нужно ввести git checkout – имя\_файла.

# Удалённые репозитории представляют собой версии проекта, сохраненные в интернете или ещё где-то в сети.

# Для того, чтобы просмотреть список настроенных удалённых репозиториев, можно запустить команду git remote – она выведет названия доступных удалённых репозиториев. Нужно ввести git clone для клонирования репозитория.

# Для того, чтобы добавить удалённый репозиторий и присвоить ему имя, достаточно просто ввести команду git remote add имя\_репозитория его\_url.

# Чтобы получить изменения из удалённого репозитория нужно ввести команду git fetch имя\_удалённого\_репозитория. Для отправки изменений на удалённый репозиторий достаточно ввести команду git push origin master – где origin и master это ветви.

# Для просмотра удалённого репозитория существует команда git remote show origin.

# Теги используются для отметки моментов выпуска версий, а также определённых моментов в истории как важных.

# Чтобы просмотреть список имеющихся тегов, достаточно ввести команду git tag и использовать опцию –l или –list для просмотра тегов определенной версии. Чтобы создать аннотированный тег, нужно ввести команду git tag –a имя\_тега –m “комментарий”, с помощью команды git show можно посмотреть данные тега вместе с коммитом. Чтобы отправить теги на удалённый сервер требуется ввести команду git push origin имя\_тега. Если вам надо отправить все теги за один раз, то целесообразно использовать команду git push origin –tags. Чтобы удалить тег нужно ввести команду git tag –d имя\_тега, но чтобы удалить теги из внешних репозиториев нужно использовать либо git push origin: путь\_к\_тегам, либо git push origin –delete имя\_тега. Чтобы выполнить переход на версии файлов, на которые указывает тег, то можно ввести команду git checkout <версия>.

# Git fetch –prune удаляет все ссылки на ветки, которые больше не существуют. Git push –prune удаляет ветвь, которая не имеет своей копии в локальном репозитории.