

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
ИНСТИТУТ ЦИФРОВОГО РАЗВИТИЯ**

**Отчет о лабораторной работе №2 по дисциплине:
«Основы программной инженерии»**

Выполнил:
Гребе Владислав
Александрович,
2 курс, группа ПИЖ-б-о-20-1,

Проверил:
Доцент кафедры
прикладной математики и
компьютерной безопасности,
Воронкин Р.А.

Отчет защищен с оценкой_____Дата защиты_____

Ставрополь, 2021 г.

ВЫПОЛНЕНИЕ:

```
C:\Users\gravity_cat\LR2>git commit -m "commit 1"
[main 5f77176] commit 1
1 file changed, 16 insertions(+)
create mode 100644 main.py
C:\Users\gravity_cat\LR2>
```

Рисунок 2.1 - Первый коммит

```
Microsoft Windows [Version 6.1.7601]
(c) Корпорация Майкрософт (Microsoft Corp.), 2009. Все права защищены.

C:\Users\gravity_cat>cd LR2

C:\Users\gravity_cat\LR2>git log
commit 5f771762bfa520b1b9448560a11c268013779526 (HEAD -> main)
Author: GravityC4T <GravityC4T>
Date: Mon Mar 21 21:08:06 2022 +0200

    commit 1

commit ccb6b104c8ef68bca37b761ca120779e8c10197f (origin/main, origin/HEAD)
Author: GravityC4T <90048571+GravityC4T@users.noreply.github.com>
Date: Mon Mar 21 21:02:26 2022 +0200

    Initial commit

C:\Users\gravity_cat\LR2>
```

Рисунок 2.2 - История хранилища (команда "git log")

```
C:\Users\gravity_cat\LR2>git checkout -- main.py

C:\Users\gravity_cat\LR2>git reset --hard HEAD~1
HEAD is now at ccb6b10 Initial commit
```

Рисунок 2.3 – Команды checkout и reset

ОТВЕТЫ НА ВОПРОСЫ:

1. Как выполнить историю коммитов в Git? Какие существуют дополнительные опции для просмотра истории коммитов?

Одним из основных и наиболее мощных инструментов для этого является команда `git log`. По умолчанию (без аргументов) `git log` перечисляет коммиты, сделанные в репозитории в обратном к хронологическому порядку — последние коммиты находятся вверху. Данная команда перечисляет коммиты с их SHA-1 контрольными суммами, именем и электронной почтой автора, датой создания и сообщением коммита. Команда `git log` имеет очень большое количество опций для поиска коммитов по разным критериям. Одним из самых полезных аргументов является `-p` или `--patch`, который показывает разницу (выводит патч), внесенную в каждый коммит. Если вы хотите увидеть сокращенную статистику для каждого коммита, вы можете использовать опцию `--stat`. Опция `--pretty` меняет формат вывода. Существует несколько встроенных вариантов отображения. Опция `oneline` выводит каждый коммит в одну строку, что может быть очень удобным если вы

просматриваете большое количество коммитов. К тому же, опции `short`, `full` и `fuller` делают вывод приблизительно в том же формате, но с меньшим или большим количеством информации соответственно. Наиболее интересной опцией является `format`, которая позволяет указать формат для вывода информации. Особенно это может быть полезным, когда вы хотите сгенерировать вывод для автоматического анализа — так как вы указываете формат явно, он не будет изменен даже после обновления Git.

2. Как ограничить вывод при просмотре истории коммитов?

Вы можете ограничить количество записей в выводе команды; используйте параметр `-2` для вывода только двух записей, опции для ограничения вывода по времени: `--since` и `--until`, следующим действительно полезным фильтром является опция `-S`, которая принимает аргумент в виде строки и показывает только те коммиты, в которых изменение в коде повлекло за собой добавление или удаление этой строки. Последней полезной опцией, которую принимает команда `git log` как фильтр, является путь. Если вы укажете каталог или имя файла, вы ограничите вывод только теми коммитами, в которых были изменения этих файлов.

3. Как внести изменения в уже сделанный коммит?

`git commit --amend`.

4. Как отменить индексацию файла в Git?

`git reset HEAD`.

5. Как отменить изменения в файле?

`git checkout --`.

6. Что такое удаленный репозиторий Git?

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

7. Как выполнить просмотр удаленных репозиториях данного локального репозитория?

`git remote` `git remote -v`

8. Как добавить удаленный репозиторий для данного локального репозитория?

`Git remote add`

9. Как выполнить отправку/получение изменений с удаленного репозитория?

`git fetch [remote-name]` `git pull`

10. Как выполнить просмотр удаленного репозитория?

`git remote show`

11. Каково назначение тэгов Git?

Как и большинство СКВ, Git имеет возможность пометить определённые моменты в истории как важные. Как правило, эта функциональность используется для отметки моментов выпуска версий (v1.0, и т. п.).

12. Как осуществляется работа с тэгами Git?

Просмотр списка тэгов: `git tag` (`git tag -l`) Пример создания аннотированного тэга: `git tag -a v1.4 -m "my version 1.4"` С помощью команды `git show` вы можете посмотреть данные тэга вместе с коммитом.

13. Самостоятельно изучите назначение флага `--prune` в командах `git fetch` и `git push`. Каково назначение этого флага?

Данный флаг помогает удалять старые удалённые ветки.