

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №2
дисциплины «Программирование на Python»

Выполнила:
Кубанова Ксения Олеговна
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника», очная
форма обучения

(подпись)

Руководитель практики:
Воронкин Р.А.

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Тема: Исследование возможностей Git для работы с локальными репозиториями

Цель: исследовать базовые возможности системы контроля версий Git для работы с локальными репозиториями.

Теоретические сведения

Ограничение вывода

В дополнение к опциям форматирования вывода, команда `git log` принимает несколько опций для ограничения вывода — опций, с помощью которых можно увидеть определенное подмножество коммитов. Вы уже видели одну из таких опций — это опция `-2`, которая показывает только последние два коммита. В действительности вы можете использовать `-n`, где `n` — это любое натуральное число и представляет собой `n` последних коммитов. На практике вы не будете часто использовать эту опцию, потому что Git по умолчанию использует постраничный вывод и вы будете видеть только одну страницу за раз.

Операции отмены

В любой момент вам может потребоваться что-либо отменить. Здесь мы рассмотрим несколько основных способов отмены сделанных изменений. Будьте осторожны, не все операции отмены в свою очередь можно отменить! Это одна из редких областей Git, где неверными действиями можно необратимо удалить результаты своей работы.

Отмена индексации файла

Следующие два раздела демонстрируют как работать с индексом и изменениями в рабочем каталоге. Радует, что команда, которой вы определяете состояние этих областей, также подсказывает вам как отменять изменения в них. Например, вы изменили два файла и хотите добавить их в разные коммиты, но случайно выполнили команду `git add *` и добавили в индекс оба. Как исключить из индекса один из них? Команда `git status` напомнит вам.

Отмена изменений в файле

Что делать, если вы поняли, что не хотите сохранять свои изменения файла CONTRIBUTING.md ? Как можно просто отменить изменения в нём — вернуть к тому состоянию, которое было в последнем коммите (или к начальному после клонирования, или еще как-то полученному)? Нам повезло, что `git status` подсказывает и это тоже.

Работа с удалёнными репозиториями

Для того, чтобы внести вклад в какой-либо Git-проект, вам необходимо уметь работать с удалёнными репозиториями. Удалённые репозитории представляют собой версии вашего проекта, сохранённые в интернете или ещё где-то в сети. У вас может быть несколько удалённых репозиториев, каждый из которых может быть доступен для чтения или для чтения-записи. Взаимодействие с другими пользователями предполагает управление удалёнными репозиториями, а также отправку и получение данных из них. Управление репозиториями включает в себя как умение добавлять новые, так и умение удалять устаревшие репозитории, а также умение управлять различными удалёнными ветками, объявлять их отслеживаемыми или нет и так далее. В данном разделе мы рассмотрим некоторые из этих навыков.

Просмотр удалённых репозиториев

Для того, чтобы просмотреть список настроенных удалённых репозиториев, вы можете запустить команду `git remote`. Она выведет названия доступных удалённых репозиториев. Если вы клонировали репозиторий, то увидите как минимум `origin` — имя по умолчанию, которое Git даёт серверу, с которого производилось клонирование

Добавление удалённых репозиториев

В предыдущих разделах мы уже упоминали и приводили примеры добавления удалённых репозиториев, сейчас рассмотрим эту операцию подробнее. Для того, чтобы добавить удалённый репозиторий и присвоить ему имя (`shortname`), просто выполните команду `git remote add`.

Получение изменений из удалённого репозитория — Fetch и Pull

Данная команда связывается с указанным удалённым проектом и забирает все те данные проекта, которых у вас ещё нет. После того как вы выполнили команду, у вас должны появиться ссылки на все ветки из этого удалённого проекта, которые вы можете просмотреть или слить в любой момент.

Отправка изменений в удаленный репозиторий (Push)

Когда вы хотите поделиться своими наработками, вам необходимо отправить их в удалённый репозиторий. Команда для этого действия простая: `git push`. Чтобы отправить вашу ветку `master` на сервер `origin` (повторимся, что клонирование обычно настраивает оба этих имени автоматически), вы можете выполнить следующую команду для отправки ваших коммитов.

Удаление и переименование удалённых репозиториев

Для переименования удалённого репозитория можно выполнить `git remote rename`. Например, если вы хотите переименовать `pb` в `raul`, вы можете это сделать при помощи `git remote rename`.

Работа с тегами

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.). Такие пометки в Git называются тегами. В этом разделе вы узнаете, как посмотреть имеющиеся теги, как создать новые или удалить существующие, а также какие типы тегов существуют в Git.

Порядок выполнения работы

1. Для начала требуется создать новый репозиторий, который будет соответствовать данной лабораторной.

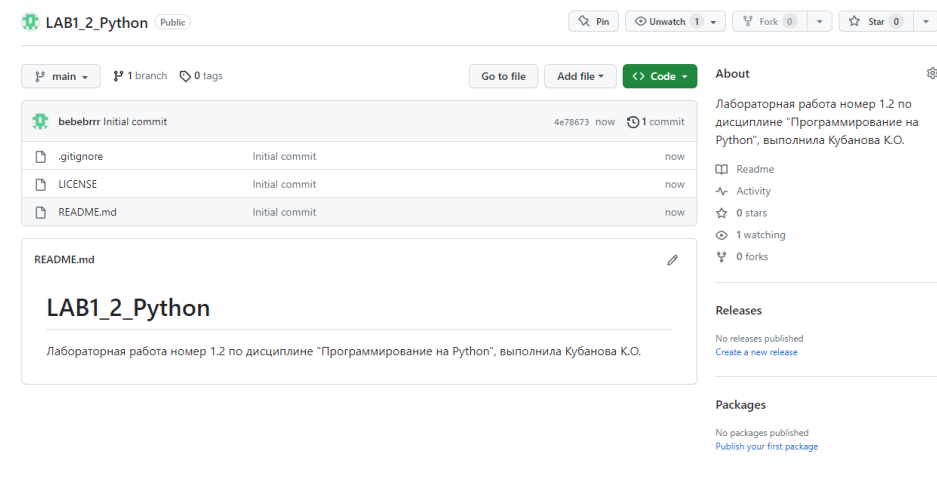


Рисунок 1. Новый репозиторий

2. Проработка примеров

2.1.

Для просмотра примеров лабораторной работы будет использоваться репозиторий с лабораторной номер 1.1.

```

MINGW64:/c:/Users/Student/Desktop/K/Программирование на py...
Student@PC-02-9_521 MINGW64 ~/Desktop/K/Программирование на pyth/1.2
$ git clone https://github.com/bebebrrr/Lab1.1_PYTH.git
Cloning into 'Lab1.1_PYTH'...
remote: Enumerating objects: 42, done.
remote: Counting objects: 100% (42/42), done.
remote: Compressing objects: 100% (25/25), done.
remote: Total 42 (delta 20), reused 35 (delta 16), pack-reused 0
Receiving objects: 100% (42/42), 619.61 KiB | 1.87 MiB/s, done.
Resolving deltas: 100% (20/20), done.

```

Рисунок 2. Клонирование репозитория

2.2.

Git log

```

Student@PC-02-9_521 MINGW64 ~/Desktop/K/Программирование на pyth/1.2/Lab1.
(main)
$ git log
commit 08f0ded2cfff1a02360e3223bb34b995c52e318 (HEAD -> main, origin/main
in/HEAD)
Author: bebebrrr <kubanova04@gmail.com>
Date: Wed Sep 13 09:48:29 2023 +0300

    Программа

commit 0ba0dd736c561e4412909292fcba8be5ef0728f1
Author: bebebrrr <kubanova04@gmail.com>
Date: Wed Sep 13 09:47:55 2023 +0300

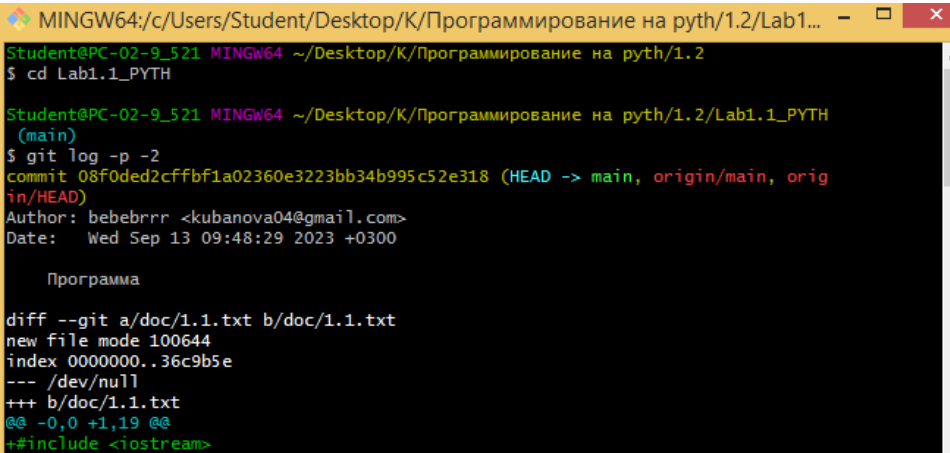
    Отчёт

```

Рисунок 3. Git log

2.3.

Git log -p -2

A screenshot of a Windows terminal window with a yellow title bar. The terminal shows the command 'git log -p -2' and its output. The output includes commit details for commit 08f0ded2cfff1a02360e3223bb34b995c52e318, the author 'bebebrrr', and a diff for a new file 'doc/1.1.txt' with 19 insertions. The diff shows the file content starting with '#include <iostream>'.

```
Student@PC-02-9_521 MINGW64 ~/Desktop/K/Программирование на pyth/1.2/Lab1...
$ cd Lab1.1_PYTH

Student@PC-02-9_521 MINGW64 ~/Desktop/K/Программирование на pyth/1.2/Lab1.1_PYTH
(main)
$ git log -p -2
commit 08f0ded2cfff1a02360e3223bb34b995c52e318 (HEAD -> main, origin/main, origin/HEAD)
Author: bebebrrr <kubanova04@gmail.com>
Date:   Wed Sep 13 09:48:29 2023 +0300

    Программа

diff --git a/doc/1.1.txt b/doc/1.1.txt
new file mode 100644
index 0000000..36c9b5e
--- /dev/null
+++ b/doc/1.1.txt
@@ -0,0 +1,19 @@
#include <iostream>
```

Рисунок 4. Git log -p -2

2.4.

Git log --stat

A screenshot of a Windows terminal window with a yellow title bar. The terminal shows the command 'git log --stat' and its output. The output includes commit details for commit 08f0ded2cfff1a02360e3223bb34b995c52e318, the author 'bebebrrr', and a summary of changes: '1 file changed, 19 insertions(+)' for 'doc/1.1.txt'. It also shows a second commit 0ba0dd736c561e4412909292fcb8be5ef0728f1 with '1 file changed, 0 insertions(+), 0 deletions(-)' for 'doc/1.1.pdf'.

```
Student@PC-02-9_521 MINGW64 ~/Desktop/K/Программирование на pyth/1.2/Lab1.1_PYTH
(main)
$ git log --stat
commit 08f0ded2cfff1a02360e3223bb34b995c52e318 (HEAD -> main, origin/main, origin/HEAD)
Author: bebebrrr <kubanova04@gmail.com>
Date:   Wed Sep 13 09:48:29 2023 +0300

    Программа

doc/1.1.txt | 19 +++++++++++++++++++++
1 file changed, 19 insertions(+)

commit 0ba0dd736c561e4412909292fcb8be5ef0728f1
Author: bebebrrr <kubanova04@gmail.com>
Date:   Wed Sep 13 09:47:55 2023 +0300

    Отчёт

doc/1.1.pdf | Bin 0 -> 698721 bytes
1 file changed, 0 insertions(+), 0 deletions(-)
```

Рисунок 5. Git log --stat

2.5.

Git log --pretty=oneline

```
MINGW64:/c/Users/Student/Desktop/K/Программирование на pyth...
Student@PC-02-9_521 MINGW64 ~/Desktop/K/Программирование на pyth/1.2/Lab1.1_PYTH
(main)
$ git log --pretty=oneline
08f0ded2cfff1a02360e3223bb34b995c52e318 (HEAD -> main, origin/main, origin/HEAD)
) Программа
0ba0dd736c561e4412909292fcb8be5ef0728f1 Отчёт
f41f2a03d133482bb4465f561787357a1a18a6ca Обновлённый
235a9ef645b680fa93af74a603f5a508c68af6ea Нахождение суммы введённых переменных
e2350323dcf3a3b8ae38abf6a2276110758a912c Ввод d
4d67f7e977e2486949a7742ee145d0b49caa227e Ввод c
ad6b77cbd10413d470d58da2e6689269687b4dfe Ввод b
b68eae51e305d6064df688abdd63beb1eb950a9 Ввод a
2251fcb990106b8dedfa0de10180fb34d7bd2f80 Инициализация переменных
e081db73bf9808d0df4efc80432b93ed2cb32e0 Приветствие и представление
f6e1679dbd6e6a562bdc680fb0ad0c25ea276116 Приветствие
7c6c75e4f744a7b38c00ca288069f15e4c4cdb6d Изменено
8cf73a80be1488005f85564dfec46e9be73e085 Initial commit
```

Рисунок 6. Git --pretty=oneline

2.6.

На данном этапе перехожу на другой репозиторий.

```
MINGW64:/c/Users/Student/Desktop/K/Программирование на pyth...
Student@PC-02-9_521 MINGW64 ~/Desktop/K/Программирование на pyth/1.2
$ git clone https://github.com/schacon/simplegit-progit
Cloning into 'simplegit-progit'...
remote: Enumerating objects: 13, done.
remote: Total 13 (delta 0), reused 0 (delta 0), pack-reused 13
Receiving objects: 100% (13/13), done.
Resolving deltas: 100% (3/3), done.
```

Рисунок 7. Переход на другой репозиторий

Git log --pretty=format: "%h - %an, %ar : %s"

```
Student@PC-02-9_521 MINGW64 ~/Desktop/K/Программирование на pyth/1.2/simplegit-progit (m
aster)
$ git log --pretty=format:"%h %s" --graph
* ca82a6d changed the verison number
* 085bb3b removed unnecessary test code
* a11bef0 first commit
```

Рисунок 8. Git log --pretty=format:"%h %s" --graph

2.7.

Git log --since = ...

```
Student@PC-02-9_521 MINGW64 ~/Desktop/K/Программирование на pyth/1.2/simplegit-progit (m
aster)
$ git log --since=15.years
commit ca82a6dff817ec66f44342007202690a93763949 (HEAD -> master, origin/master, origin/H
EAD)
Author: Scott Chacon <schacon@gmail.com>
Date: Mon Mar 17 21:52:11 2008 -0700

    changed the verison number

commit 085bb3bcb608e1e8451d4b2432f8ecbe6306e7e7
Author: Scott Chacon <schacon@gmail.com>
Date: Sat Mar 15 16:40:33 2008 -0700

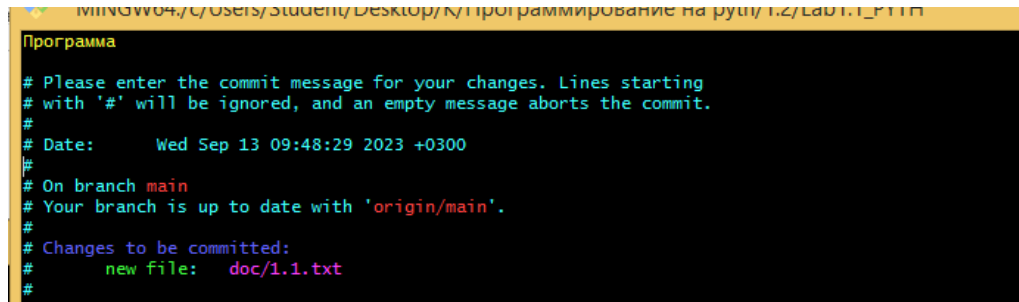
    removed unnecessary test code

Student@PC-02-9_521 MINGW64 ~/Desktop/K/Программирование на pyth/1.2/simplegit-progit (m
aster)
$
```

Рисунок 9. Git log --since=15.years

2.8.

Git commit --amend

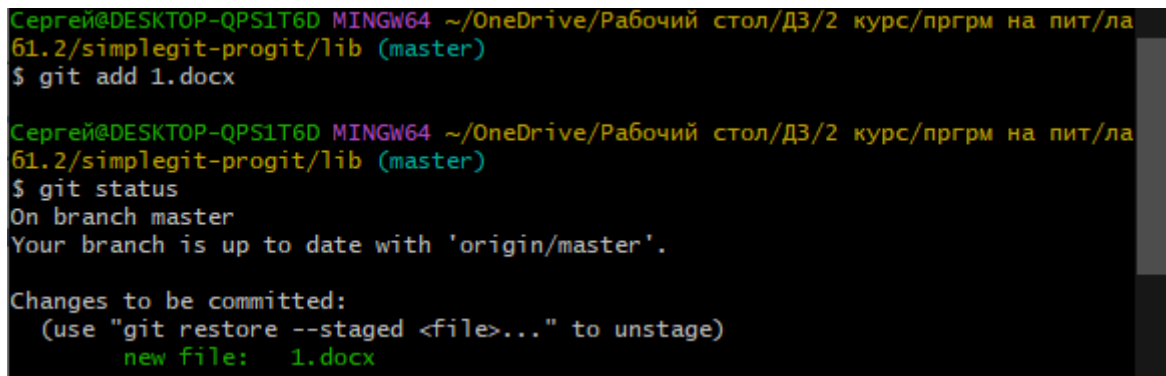


```
Программа
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# Date:      Wed Sep 13 09:48:29 2023 +0300
#
# On branch main
# Your branch is up to date with 'origin/main'.
#
# Changes to be committed:
#   new file:   doc/1.1.txt
#
```

Рисунок 10. Git commit --amend

2.9.

Git status



```
Сергей@DESKTOP-QPS1T6D MINGW64 ~/OneDrive/Рабочий стол/ДЗ/2 курс/пргрм на пит/лаб 61.2/simplegit-progit/lib (master)
$ git add 1.docx

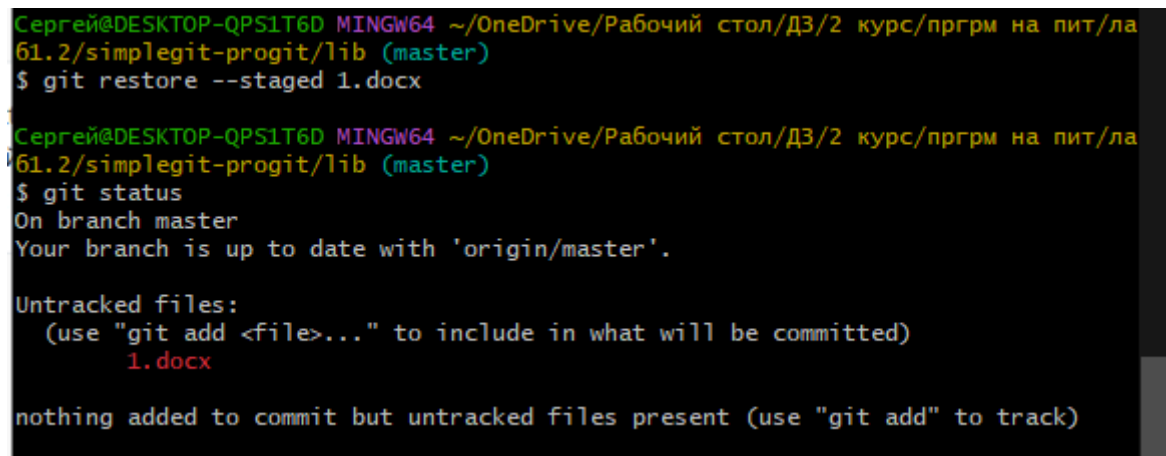
Сергей@DESKTOP-QPS1T6D MINGW64 ~/OneDrive/Рабочий стол/ДЗ/2 курс/пргрм на пит/лаб 61.2/simplegit-progit/lib (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   1.docx
```

Рисунок 1 - Git status

Компилятор сам указывает на то, какую команду следует использовать для отмены индексирования файла. В данном случае я использую команду `restore --staged`, и при запросе статуса видим, что индексация отменена.

Git restore --staged



```
Сергей@DESKTOP-QPS1T6D MINGW64 ~/OneDrive/Рабочий стол/ДЗ/2 курс/пргрм на пит/лаб 61.2/simplegit-progit/lib (master)
$ git restore --staged 1.docx

Сергей@DESKTOP-QPS1T6D MINGW64 ~/OneDrive/Рабочий стол/ДЗ/2 курс/пргрм на пит/лаб 61.2/simplegit-progit/lib (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    1.docx

nothing added to commit but untracked files present (use "git add" to track)
```

Рисунок 12 - Git status and git restore --staged

Удалённые репозитории

3.1

Git remote

```
Сергей@DESKTOP-QPS1T6D MINGW64 ~/OneDrive/Рабочий стол/ДЗ/2 курс/пргм на пит/ла
61.2
$ git clone https://github.com/schacon/ticgit
Cloning into 'ticgit'...
remote: Enumerating objects: 1857, done.
remote: Total 1857 (delta 0), reused 0 (delta 0), pack-reused 1857
Receiving objects: 100% (1857/1857), 334.06 KiB | 1.01 MiB/s, done.
Resolving deltas: 100% (837/837), done.

Сергей@DESKTOP-QPS1T6D MINGW64 ~/OneDrive/Рабочий стол/ДЗ/2 курс/пргм на пит/ла
61.2
$ cd ticgit

Сергей@DESKTOP-QPS1T6D MINGW64 ~/OneDrive/Рабочий стол/ДЗ/2 курс/пргм на пит/ла
61.2/ticgit (master)
$ git remote
origin
```

Рисунок 13 - Git remote

Git remote -v

```
$ git remote -v
origin https://github.com/schacon/ticgit (fetch)
origin https://github.com/schacon/ticgit (push)
```

Рисунок 14 - Git remote -v

3.2

```
Сергей@DESKTOP-QPS1T6D MINGW64 ~/OneDrive/Рабочий стол/ДЗ/2 курс/пргм на пит/ла
61.2/ticgit (master)
$ git remote add pb https://github.com/paulboone/ticgit

Сергей@DESKTOP-QPS1T6D MINGW64 ~/OneDrive/Рабочий стол/ДЗ/2 курс/пргм на пит/ла
61.2/ticgit (master)
$ git remote -v
origin https://github.com/schacon/ticgit (fetch)
origin https://github.com/schacon/ticgit (push)
pb https://github.com/paulboone/ticgit (fetch)
pb https://github.com/paulboone/ticgit (push)
```

Рисунок 14 - Git remote add pb

```
$ git fetch pb
remote: Enumerating objects: 43, done.
remote: Counting objects: 100% (22/22), done.
remote: Total 43 (delta 22), reused 22 (delta 22), pack-reused 21
Unpacking objects: 100% (43/43), 5.99 KiB | 15.00 KiB/s, done.
From https://github.com/paulboone/ticgit
* [new branch]      master    -> pb/master
* [new branch]      ticgit    -> pb/ticgit
```

Рисунок 15 - Git fetch pb

3.3

```
$ git push origin master
remote: Permission to schacon/ticgit.git denied to bebebrrr.
fatal: unable to access 'https://github.com/schacon/ticgit/': The requested URL
returned error: 403
```

Рисунок 16 - Git push origin master

3.4

```
Сергей@DESKTOP-QPS1T6D MINGW64 ~/OneDrive/Рабочий стол/ДЗ/2 курс/пргрм
61.2/ticgit (master)
$ git remote show origin
* remote origin
Fetch URL: https://github.com/schacon/ticgit
Push URL: https://github.com/schacon/ticgit
HEAD branch: master
Remote branches:
  master tracked
  ticgit tracked
Local branch configured for 'git pull':
  master merges with remote master
Local ref configured for 'git push':
  master pushes to master (up to date)
```

Рисунок 17 – git remote show origin

3.5

```
$ git remote rename pb ks
Renaming remote references: 100% (2/2), done.

Сергей@DESKTOP-QPS1T6D MINGW64 ~/OneDrive/Рабочий
61.2/ticgit (master)
$ git remote
ks
origin
```

Рисунок 18 – git remote rename

```
Сергей@DESKTOP-QPS1T6D MINGW64 ~/OneDrive/Рабочий
61.2/ticgit (master)
$ git remote remove ks

Сергей@DESKTOP-QPS1T6D MINGW64 ~/OneDrive/Рабочий
61.2/ticgit (master)
$ git remote
origin
```

Рисунок 19 – git remote remove

Работа с тэгами

4.1.

```
Сергей@DESKTOP-QPS1T6D MINGW64 ~/OneDrive/Рабочий
61.2/LAB1_2_Python/doc (main)
$ git tag -a v1.4 -m "my version 1.4"

Сергей@DESKTOP-QPS1T6D MINGW64 ~/OneDrive/Рабочий
61.2/LAB1_2_Python/doc (main)
$ git tag
v1.4
```

Рисунок 20 – создание и просмотр тэгов

4.2.

```
$ git show v1.4
tag v1.4
Tagger: bebebrrr <kubanova04@gmail.com>
Date:   Wed Sep 20 18:33:03 2023 +0300

my version 1.4

commit d0b88bd461dbdbb9bb7e073bddca52600fc17a0e (HEAD -> main, tag: v1.4, origin
/main, origin/HEAD)
Author: bebebrrr <kubanova04@gmail.com>
Date:   Mon Sep 18 17:43:06 2023 +0300

    Удалено

diff --git a/doc/1.docx b/doc/1.docx
deleted file mode 100644
index e69de29..0000000
<C:\Users\D899~1\AppData\Local\Temp/git-blob-a13588/1.docx> does not seem to be
a docx file!
```

Рисунок 21 – git show

4.3.

```
$ git push origin v1.4
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 161 bytes | 161.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/bebebrrr/LAB1_2_Python.git
 * [new tag]         v1.4 -> v1.4
```

Рисунок 22 – git push origin

4.4.

```
$ git tag -d v1.4
Deleted tag 'v1.4' (was 80c2d98)
```

Рисунок 23 – git tag -d

4.5.

```
$ git push origin --delete v1.4
To https://github.com/bebebrrr/LAB1_2_Python.git
- [deleted]         v1.4
```

Рисунок 24 – git push origin --delete

Выполнение основных заданий

5.1.

С помощью VS дополняю файл README.

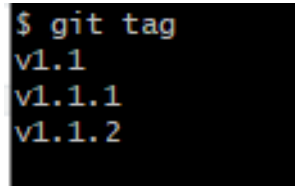
```
# LAB1_2_Python

<h1 align="center">Лабораторная работа номер 1.2 по дисциплине "Программирование на Python"
<h3 align="center">Выполнила Кубанова К.О., группа ИВТ-6-о-22-1
```

Рисунок 25 – дополнение README

5.2.

На рисунке 26 изображено количество тэгов в файле.

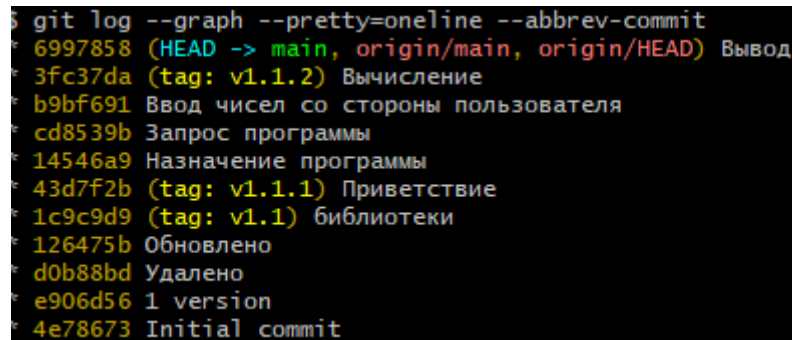


```
$ git tag
v1.1
v1.1.1
v1.1.2
```

Рисунок 26 – тэги в программе

5.3.

С помощью git log просматриваю коммиты.

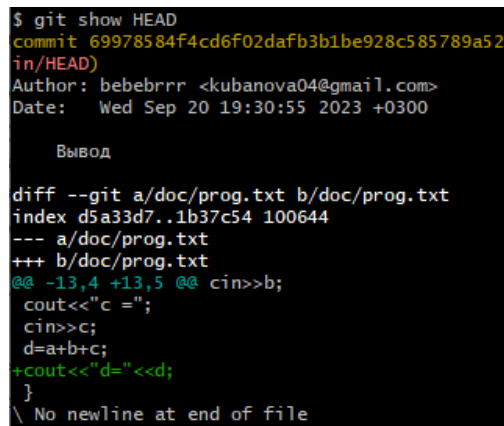


```
git log --graph --pretty=oneline --abbrev-commit
* 6997858 (HEAD -> main, origin/main, origin/HEAD) Вывод
* 3fc37da (tag: v1.1.2) Вычисление
* b9bf691 Ввод чисел со стороны пользователя
* cd8539b Запрос программы
* 14546a9 Назначение программы
* 43d7f2b (tag: v1.1.1) Приветствие
* 1c9c9d9 (tag: v1.1) библиотеки
* 126475b Обновлено
* d0b88bd Удалено
* e906d56 1 version
* 4e78673 Initial commit
```

Рисунок 26 – журнал

5.4.

Просмотр отдельных коммитов следующими командами, представленные на рисунках:



```
$ git show HEAD
commit 69978584f4cd6f02dafb3b1be928c585789a52
in/HEAD)
Author: bebebrrr <kubanova04@gmail.com>
Date: Wed Sep 20 19:30:55 2023 +0300

    Вывод

diff --git a/doc/prog.txt b/doc/prog.txt
index d5a33d7..1b37c54 100644
--- a/doc/prog.txt
+++ b/doc/prog.txt
@@ -13,4 +13,5 @@ cin>>b;
     cout<<"c =";
     cin>>c;
     d=a+b+c;
+cout<<"d="<<d;
 }
\ No newline at end of file
```

Рисунок 27 – git show HEAD

```
$ git show HEAD~1
commit 3fc37da05a0942a4ecf398e2ca17f11574c8656c (tag: v1.1.2)
Author: bebebrrr <kubanova04@gmail.com>
Date:   Wed Sep 20 19:28:29 2023 +0300

    Вычисление

diff --git a/doc/prog.txt b/doc/prog.txt
index 4fe2ca9..d5a33d7 100644
--- a/doc/prog.txt
+++ b/doc/prog.txt
@@ -5,10 +5,12 @@ int main()
    cout<<"Hello!";
    cout<<"This program is designed to add numbers";
    cout<<"Enter some numbers";
+int a, b, c, d;
    cout<<"a =";
    cin>>a;
    cout<<"b =";
    cin>>b;
```

Рисунок 28 – git show HEAD~1

```
$ git show 3fc37da
commit 3fc37da05a0942a4ecf398e2ca17f11574c8656c (tag: v1.1.2)
Author: bebebrrr <kubanova04@gmail.com>
Date:   Wed Sep 20 19:28:29 2023 +0300

    Вычисление

diff --git a/doc/prog.txt b/doc/prog.txt
index 4fe2ca9..d5a33d7 100644
--- a/doc/prog.txt
+++ b/doc/prog.txt
@@ -5,10 +5,12 @@ int main()
    cout<<"Hello!";
    cout<<"This program is designed to add numbers";
    cout<<"Enter some numbers";
+int a, b, c, d;
    cout<<"a =";
    cin>>a;
    cout<<"b =";
    cin>>b;
    cout<<"c =";
```

Рисунок 29 – git show <хэш>

5.5.

Далее сравниваю команды восстановления содержимого файла. Для этого сначала очищаю его от кода, а после использую следующую команду:

```
Сергей@DESKTOP-QPS1T6D MINGW64 ~/0r
61.2/LAB1_2_Python/doc (main)
$ git checkout -- prog.txt
```

Рисунок 30 – git checkout

Содержимое файла было восстановлено. Следующая команда:

```
61.2/LAB1_2_Python/doc (main)
$ git reset --hard HEAD~1
HEAD is now at 6997858 Вывод
```

Рисунок 31 – git reset hard HEAD~1

Содержимое так же было восстановлено. Обе эти команды выполняют аналогичные функции, однако если первая восстанавливает лишь по последнему коммиту, то вторая может восстановить ровно до того коммита, который будет указан.

Контрольные вопросы

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

С помощью команды `git log`.

Дополнительные опции:

<code>-(n)</code>	Показывает только последние n коммитов
<code>--since , -- after</code>	Показывает только те коммиты, которые были сделаны после указанной даты.
<code>--until , -- before</code>	Показывает только те коммиты, которые были сделаны до указанной даты
<code>--author</code>	Показывает только те коммиты, в которых запись <code>author</code> совпадает с указанной строкой.
<code>-- committer</code>	Показывает только те коммиты, в которых запись <code>committer</code> совпадает с указанной строкой.
<code>--grep</code>	Показывает только коммиты, сообщение которых содержит указанную строку.
<code>-S</code>	Показывает только коммиты, в которых изменение в коде повлекло за собой добавление или удаление указанной строки.

2. Как ограничить вывод при просмотре истории коммитов?

`Git log max count=..`

3. Как внести изменения в уже сделанный коммит?

`Git commit --amend`

4. Как отменить индексацию файла в Git?

`Git reset <file>`

5. Как отменить изменения в файле?

`Git checkout <file>`

6. Что такое удаленный репозиторий Git?

Репозиторий, расположенный на удалённом сервере и использующийся для совместной работы.

7. Как выполнить просмотр удаленных репозиториях данного локального репозитория?

Git remote

8. Как добавить удаленный репозиторий для данного локального репозитория?

Git remote add <name>

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

Git push, git pull

10. Как выполнить просмотр удаленного репозитория?

Git remote show <name>

11. Каково назначение тэгов Git?

Определённые метки, присваиваемые коммитам. Используются для обозначения важных моментов.

12. Как осуществляется работа с тэгами Git?

Git tag – для вывода списка тэгов

Git tag <tagname> - создание тэга

Git tag -d <tagname> - удаление тэга

Git push –tags – отправка тэгов на удалённый репозиторий

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

Используется для удаления ссылок на удалённые ветки и тэги, которые больше не существуют на удалённом репозитории.

Вывод: в ходе выполнения лабораторной работы были исследованы базовые возможности системы контроля версий Git для работы с локальными репозиториями.