

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»

Выполнил:
Болуров Ислам Расулович
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной техники и
автоматизированных систем», очная
форма обучения

(подпись)

Руководитель практики:
Воронкин Р. А.

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023

Исследование возможностей Git для работы с локальными репозиториями


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

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner *

 IslamBolurov ▾

Repository name *

/ Lab 1.2py

✓ Your new repository will be created as Lab-1.2py.

The repository name can only contain ASCII letters, digits, and the characters ., -, and _.

Great repository names are short and memorable. Need inspiration? How about **jubilant-system** ?

Description (optional)

Рисунок 1. Создание удалённого репозитория

```
IslamBolurov@Islam666 MINGW64 ~/Desktop/lab 222/Lab-1.2pyt (main)
$ git log
commit 52597bd4a404f0af4f0ead5b658661acf5e9bddf (HEAD -> main, origin/main, origin/HEAD)
Author: IslamBolurov <islambolurov9@gmail.com>
Date: Sat Feb 17 05:46:33 2024 +0300

    programm Hello world

commit c75c514d7cfa72380dca2a9fac61b3061465a9e6
Author: IslamBolurov <99473352+IslamBolurov@users.noreply.github.com>
Date: Sat Feb 17 05:42:46 2024 +0300

    Update README.md

commit 0deef57f81013620e285326323bb61c0b4b414a8
Author: IslamBolurov <99473352+IslamBolurov@users.noreply.github.com>
Date: Sat Feb 17 05:41:59 2024 +0300

    Initial commit

IslamBolurov@Islam666 MINGW64 ~/Desktop/lab 222/Lab-1.2pyt (main)
```

Рисунок 2. Выполнение команды git log

```

IslamBolurov@Islam666 MINGW64 ~/Desktop/lab 222/Lab-1.2pyt (main)
$ git log -p
commit 84b66fcfff5a96578e76120e130a9834794bd43f (HEAD -> main, tag: v2.0)
Author: IslamBolurov <islambolurov9@gmail.com>
Date: Sat Feb 17 11:28:49 2024 +0300

    age

diff --git a/programm/prog1.py b/programm/prog1.py
index f8a9514..b4d1a7b 100644
--- a/programm/prog1.py
+++ b/programm/prog1.py
@@ -1,2 +1,3 @@
    print("Hello World")
    print("Islam")
+print("20")
\ No newline at end of file

commit 9f81535169931b8a9d84d32fea97888237d288ac (tag: v1.0)
Author: IslamBolurov <islambolurov9@gmail.com>
Date: Sat Feb 17 08:08:52 2024 +0300

```

Рисунок3. Выполнение команды git log -p

```

IslamBolurov@Islam666 MINGW64 ~/Desktop/lab 222/Lab-1.2pyt (main)
$ git log --stat
commit 84b66fcfff5a96578e76120e130a9834794bd43f (HEAD -> main, tag: v2.0)
Author: IslamBolurov <islambolurov9@gmail.com>
Date: Sat Feb 17 11:28:49 2024 +0300

    age

    programm/prog1.py | 1 +
    1 file changed, 1 insertion(+)

commit 9f81535169931b8a9d84d32fea97888237d288ac (tag: v1.0)
Author: IslamBolurov <islambolurov9@gmail.com>
Date: Sat Feb 17 08:08:52 2024 +0300

    add name

    programm/prog1.py | 1 +
    1 file changed, 1 insertion(+)

commit 52597bd4a404f0af4f0ead5b658661acf5e9bddf (origin/main, origin/HEAD)
Author: IslamBolurov <islambolurov9@gmail.com>
Date: Sat Feb 17 05:46:33 2024 +0300

```

Рисунок4. Выполнение команды git log --stat

```

IslamBolurov@Islam666 MINGW64 ~/Desktop/lab 222/Lab-1.2pyt (main)
$ git log --pretty=oneline
9f81535169931b8a9d84d32fea97888237d288ac (HEAD -> main) add name
52597bd4a404f0af4f0ead5b658661acf5e9bddf (origin/main, origin/HEAD) programm Hel
lo World
c75c514d7cfa72380dca2a9fac61b3061465a9e6 Update README.md
0deef57f81013620e285326323bb61c0b4b414a8 Initial commit

```

Рисунок5. Выполнение команды git log --pretty=oneline

```

IslamBolurov@Islam666 MINGW64 ~/Desktop/lab 222/Lab-1.2pyt (main)
$ git log --pretty=format:"%h - an, %ar : %s"
9b99104 - an, 6 minutes ago : surname
84b66fc - an, 43 minutes ago : age
9f81535 - an, 4 hours ago : add name
52597bd - an, 6 hours ago : programm Hello World
c75c514 - an, 7 hours ago : Update README.md
0deef57 - an, 7 hours ago : Initial commit

```

Рисунок 6. Выполнение команды git log --pretty=format:"%h – an, %ar : %s"

```
IslamBolurov@Islam666 MINGW64 ~/Desktop/lab 222/Lab-1.2pyt (main)
$ git log --pretty=format:"%h %s" --graph
* 9b99104 surname
* 84b66fc age
* 9f81535 add name
* 52597bd programm Hello World
* c75c514 Update README.md
* 0deef57 Initial commit
```

Рисунок7. Выполнение команды git log --pretty=format:"%h %s" --graph

```
IslamBolurov@Islam666 MINGW64 ~/Desktop/lab 222/Lab-1.2pyt (main)
$ git log --since=2.weeks
commit 9b99104fff4084e4d3194316674f50a6e2974660 (HEAD -> main)
Author: IslamBolurov <islambolurov9@gmail.com>
Date: Sat Feb 17 12:06:39 2024 +0300

    surname

commit 84b66fcfff5a96578e76120e130a9834794bd43f
Author: IslamBolurov <islambolurov9@gmail.com>
Date: Sat Feb 17 11:28:49 2024 +0300

    age

commit 9f81535169931b8a9d84d32fea97888237d288ac
Author: IslamBolurov <islambolurov9@gmail.com>
Date: Sat Feb 17 08:08:52 2024 +0300

    add name

commit 52597bd4a404f0af4f0ead5b658661acf5e9bddf (origin/main, origin/HEAD)
Author: IslamBolurov <islambolurov9@gmail.com>
Date: Sat Feb 17 05:46:33 2024 +0300

    programm Hello World

commit c75c514d7cfa72380dca2a9fac61b3061465a9e6
Author: IslamBolurov <99473352+IslamBolurov@users.noreply.github.com>
Date: Sat Feb 17 05:42:46 2024 +0300

    Update README.md

commit 0deef57f81013620e285326323bb61c0b4b414a8
Author: IslamBolurov <99473352+IslamBolurov@users.noreply.github.com>
Date: Sat Feb 17 05:41:59 2024 +0300

    Initial commit
```

Рисунок8. Выполнение команды git log --since=2weeks

```
IslamBolurov@Islam666 MINGW64 ~/Desktop/lab 222/Lab-1.2pyt (main)
$ git remote -v
origin https://github.com/IslamBolurov/Lab-1.2pyt.git (fetch)
origin https://github.com/IslamBolurov/Lab-1.2pyt.git (push)
```

Рисунок9. Выполнение команды git remote -v

```
IslamBolurov@Islam666 MINGW64 ~/Desktop/lab 222/Lab-1.2pyt (main)
$ git tag v2.0 84b66fcfff5a96578e76120e130a9834794bd43f
```

Рисунок10. Создание тэга v2.0

```

IslamBolurov@Islam666 MINGW64 ~/Desktop/lab 222/Lab-1.2pyt (main)
$ git show v2.0
commit 84b66fcfff5a96578e76120e130a9834794bd43f (tag: v2.0)
Author: IslamBolurov <islambolurov9@gmail.com>
Date: Sat Feb 17 11:28:49 2024 +0300

    age

diff --git a/programm/prog1.py b/programm/prog1.py
index f8a9514..b4d1a7b 100644
--- a/programm/prog1.py
+++ b/programm/prog1.py
@@ -1,2 +1,3 @@
 print("Hello World")
 print("Islam")
+print("20")
\ No newline at end of file

```

Рисунок11. Выполнение команды git show v2.0

```

IslamBolurov@Islam666 MINGW64 ~/Desktop/lab 222/Lab-1.2pyt (main)
$ git tag v3.0 9b99104fff4084e4d3194316674f50a6e2974660

```

Рисунок12. Создание последнего тэга

```

IslamBolurov@Islam666 MINGW64 ~/Desktop/lab 222/Lab-1.2pyt (main)
$ git push origin --tags
Enumerating objects: 15, done.
Counting objects: 100% (15/15), done.
Delta compression using up to 4 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (12/12), 942 bytes | 314.00 KiB/s, done.
Total 12 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 1 local object.
To https://github.com/IslamBolurov/Lab-1.2pyt.git
 * [new tag]          v1.0 -> v1.0
 * [new tag]          v2.0 -> v2.0
 * [new tag]          v3.0 -> v3.0

```

Рисунок 13. Пуш всех тэгов на удалённый репозиторий

```

IslamBolurov@Islam666 MINGW64 ~/Desktop/lab 222/Lab-1.2pyt (main)
$ git log --graph
* commit 84b66fcfff5a96578e76120e130a9834794bd43f (HEAD -> main, tag: v2.0)
 | Author: IslamBolurov <islambolurov9@gmail.com>
 | Date: Sat Feb 17 11:28:49 2024 +0300
 |
 |     age
 |
 * commit 9f81535169931b8a9d84d32fea97888237d288ac (tag: v1.0)
 | Author: IslamBolurov <islambolurov9@gmail.com>
 | Date: Sat Feb 17 08:08:52 2024 +0300
 |
 |     add name
 |
 * commit 52597bd4a404f0af4f0ead5b658661acf5e9bddf (origin/main, origin/HEAD)
 | Author: IslamBolurov <islambolurov9@gmail.com>
 | Date: Sat Feb 17 05:46:33 2024 +0300
 |
 |     programm Hello World

```

Рисунок 14. Выполнение команды git log --graph

```

IslamBolurov@Islam666 MINGW64 ~/Desktop/lab 222/Lab-1.2pyt (main)
$ git log --pretty=oneline
9b99104fff4084e4d3194316674f50a6e2974660 (HEAD -> main, tag: v3.0) surname
84b66fcfff5a96578e76120e130a9834794bd43f (tag: v2.0) age
9f81535169931b8a9d84d32fea97888237d288ac (tag: v1.0) add name
52597bd4a404f0af4f0ead5b658661acf5e9bddf (origin/main, origin/HEAD) programm Hello Wor
d
c75c514d7cfa72380dca2a9fac61b3061465a9e6 Update README.md
0deef57f81013620e285326323bb61c0b4b414a8 Initial commit

```

Рисунок 15. Выполнение команды git log --pretty=oneline

```

IslamBolurov@Islam666 MINGW64 ~/Desktop/lab 222/Lab-1.2pyt (main)
$ git log --abbrev-commit
commit 9b99104 (HEAD -> main, tag: v3.0)
Author: IslamBolurov <islambolurov9@gmail.com>
Date: Sat Feb 17 12:06:39 2024 +0300

    surname

commit 84b66fc (tag: v2.0)
Author: IslamBolurov <islambolurov9@gmail.com>
Date: Sat Feb 17 11:28:49 2024 +0300

    age

commit 9f81535 (tag: v1.0)
Author: IslamBolurov <islambolurov9@gmail.com>
Date: Sat Feb 17 08:08:52 2024 +0300

    add name

commit 52597bd (origin/main, origin/HEAD)
Author: IslamBolurov <islambolurov9@gmail.com>
Date: Sat Feb 17 05:46:33 2024 +0300

    programm Hello world

```

Рисунок 16. Выполнение команды git log --abbrev-commit

```

IslamBolurov@Islam666 MINGW64 ~/Desktop/lab 222/Lab-1.2pyt (main)
$ git show head
commit 9b99104fff4084e4d3194316674f50a6e2974660 (HEAD -> main, tag: v3.0)
Author: IslamBolurov <islambolurov9@gmail.com>
Date: Sat Feb 17 12:06:39 2024 +0300

    surname

diff --git a/programm/prog1.py b/programm/prog1.py
index b4d1a7b..0836535 100644
--- a/programm/prog1.py
+++ b/programm/prog1.py
@@ -1,3 +1,4 @@
 print("Hello World")
 print("Islam")
-print("20")
\ No newline at end of file
+print("20")
+print("Bolurov")
\ No newline at end of file

```

Рисунок 17. Выполнение команды git show head

```

IslamBolurov@Islam666 MINGW64 ~/Desktop/lab 222/Lab-1.2pyt (main)
$ git show head~2
commit 9f81535169931b8a9d84d32fea97888237d288ac (tag: v1.0)
Author: IslamBolurov <islambolurov9@gmail.com>
Date: Sat Feb 17 08:08:52 2024 +0300

    add name

diff --git a/programm/prog1.py b/programm/prog1.py
index ad35e5a..f8a9514 100644
--- a/programm/prog1.py
+++ b/programm/prog1.py
@@ -1,2 @@
 print("Hello World")
+print("Islam")

```

Рисунок18. Выполнение команды git show head~2

```

IslamBolurov@Islam666 MINGW64 ~/Desktop/lab 222/Lab-1.2pyt (main)
$ git show 9b99104fff4084e4d3194316674f50a6e2974660
commit 9b99104fff4084e4d3194316674f50a6e2974660 (HEAD -> main, tag: v3.0)
Author: IslamBolurov <islambolurov9@gmail.com>
Date: Sat Feb 17 12:06:39 2024 +0300

    surname

diff --git a/programm/prog1.py b/programm/prog1.py
index b4d1a7b..0836535 100644
--- a/programm/prog1.py
+++ b/programm/prog1.py
@@ -1,3 +1,4 @@
 print("Hello World")
 print("Islam")
-print("20")
\ No newline at end of file
+print("20")
+print("Bolurov")
\ No newline at end of file

```

Рисунок19. Обращение к коммиту по хэш сумме

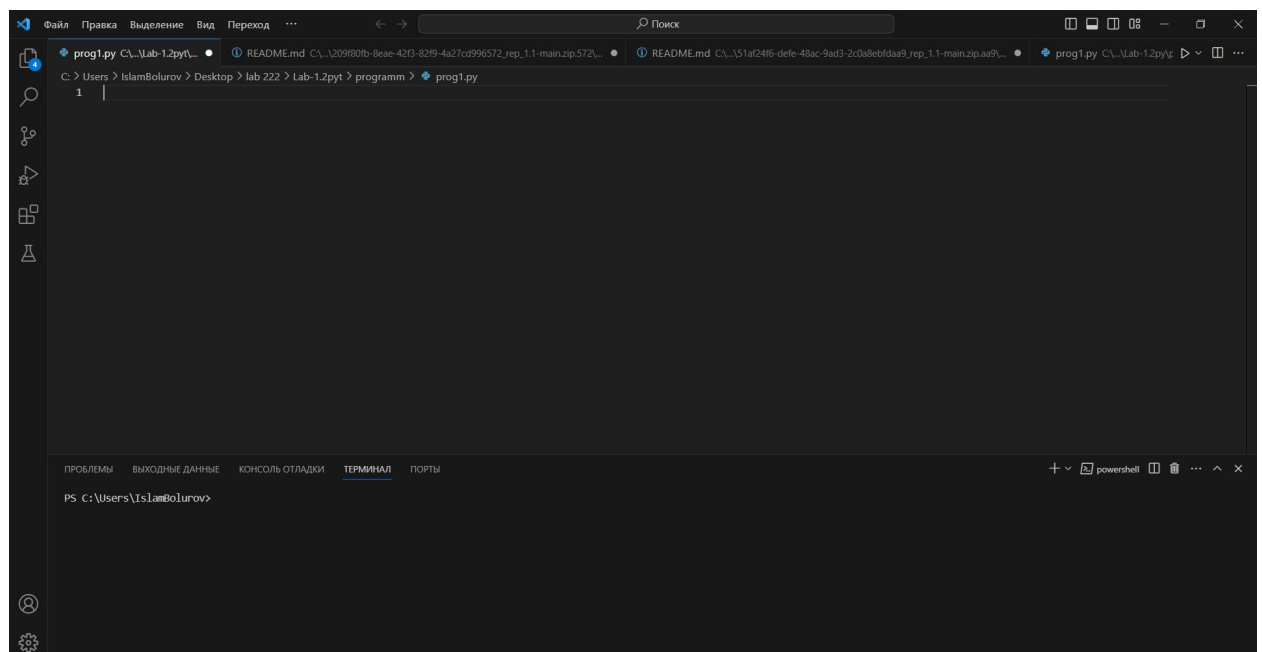


Рисунок20. Удаление кода

```
C: > Users > IslamBolurov > Desktop > lab 222 > Lab-1.2pyt > programm > prog1.py
1 print("Hello World")
2 print("Islam")
3 print("20")
4 print("Bolurov")
```

```
MINGW64:/c/Users/IslamBolurov/Desktop/lab 222/Lab-1.2pyt
IslamBolurov@Islam666 MINGW64 ~/Desktop/lab 222/Lab-1.2pyt (main)
$ git checkout -- programm
IslamBolurov@Islam666 MINGW64 ~/Desktop/lab 222/Lab-1.2pyt (main)
$ |
```

Рисунок21. Выполнение команды checkout – program

```
C: > Users > IslamBolurov > Desktop > lab 222 > Lab-1.2pyt > programm > prog1.py
1
```

```
MINGW64:/c/Users/IslamBolurov/Desktop/lab 222/Lab-1.2pyt
IslamBolurov@Islam666 MINGW64 ~/Desktop/lab 222/Lab-1.2pyt (main)
$ git checkout -- programm
IslamBolurov@Islam666 MINGW64 ~/Desktop/lab 222/Lab-1.2pyt (main)
$ |
```

Рисунок22. Удаление во второй раз

```
C: > Users > IslamBolurov > Desktop > lab 222 > Lab-1.2pyt > programm > prog1.py
1 print("Hello World")
2 print("Islam")
3 print("20")
```

```
MINGW64:/c/Users/IslamBolurov/Desktop/lab 222/Lab-1.2pyt
IslamBolurov@Islam666 MINGW64 ~/Desktop/lab 222/Lab-1.2pyt (main)
$ git reset --hard head~1
HEAD is now at 84b66fc age
IslamBolurov@Islam666 MINGW64 ~/Desktop/lab 222/Lab-1.2pyt (main)
$
```

Рисунок23. Выполнение команды git reset –hard head~1

Вывод по пунктам 10.1 – 10.4: содержимое файла в двух случаях никак не изменилось (то есть код успешно вернулся). Разница лишь в самих ситуациях. В первом случае я не производил коммит, а во втором коммит был произведён. Следовательно, когда изменения закоммичены, но необходим откат, то командой checkout проблему не решить: необходимо использовать git reset –hard head~n)

Ответы на контрольные вопросы:

1) Для просмотра истории коммитов в Git можно использовать команду "git log". Эта команда показывает список всех коммитов в репозитории, начиная с самого нового и заканчивая самым старым.

Некоторые дополнительные опции для просмотра истории коммитов:

- "--oneline": отображает каждый коммит в одной строке, содержащей хэш коммита и его сообщение.
- "--graph": отображает историю коммитов в виде графа, показывая связи между коммитами и ветками.
- "--author": фильтрует историю коммитов по автору.
- "--since" и "--until": фильтруют историю коммитов по дате, позволяя указать интервал времени.

Например, команда `git log --oneline --author=John --since=2021-01-01` покажет только те коммиты, которые были сделаны пользователем с именем "John" после 1 января 2021 года, в однострочном формате.

2) Команда `git log` принимает несколько опций для ограничения вывода — опций, с помощью которых можно увидеть определенное подмножество коммитов. В действительности можно использовать `-n`, где `n` — это любое натуральное число и представляет собой `n` последних коммитов. Есть опции для ограничения вывода по времени, такие как `--since` и `--until`. Опция `--author` дает возможность фильтровать по автору коммита, а опция `--grep` искать по ключевым словам в сообщении коммита.

3) Если вы хотите переделать коммит — внесите необходимые изменения, добавьте их в индекс и сделайте коммит ещё раз, указав параметр `--amend`.
Пример:

```
$ git commit -m 'Initial commit'
```

```
$ git add forgotten_file
```

```
$ git commit --amend
```

В итоге получится единый коммит — второй коммит заменит результаты первого.

4) Чтобы отменить индексацию файла в Git, можно использовать команду `git reset HEAD <имя_файла>`. Например, если вы хотите отменить индексацию файла `"file.txt"`, выполните следующую команду:

```
git reset HEAD file.txt
```

После этого файл будет снова находиться в состоянии "неотслеживаемых" изменений.

5) Для отмены изменений в файле можно использовать команду "git checkout -- <название файла>". Эта команда отменит все несохраненные изменения в указанном файле и восстановит его до последней версии в репозитории.

6) Удаленный репозиторий Git - это репозиторий, который расположен на удаленном сервере или хостинг-платформе, а не на локальном компьютере разработчика. Он используется для хранения и совместной работы над кодом с другими разработчиками. Удаленный репозиторий содержит все версии файлов и историю изменений, а также предоставляет возможность синхронизировать изменения между локальными и удаленными репозиториями.

7) Для просмотра удаленных репозиториях данного локального репозитория в Git можно использовать команду "git remote -v". Эта команда покажет список всех удаленных репозиториях, связанных с текущим локальным репозиторием, а также их URL-адреса.

8) Чтобы добавить удаленный репозиторий для текущего локального репозитория в Git, можно использовать команду "git remote add". Ниже приведен синтаксис этой команды:

```
git remote add <имя_удаленного_репозитория>  
<URL_адрес_удаленного_репозитория>
```

где:

- <имя_удаленного_репозитория> - имя, которое вы хотите присвоить удаленному репозиторию (обычно используется "origin");

- <URL_адрес_удаленного_репозитория> - URL-адрес удаленного репозитория, к которому вы хотите добавить доступ.

9) Чтобы отправить изменения на удаленный репозиторий, вы можете использовать команду "git push". Ниже приведен синтаксис этой команды:

```
git push <имя_удаленного_репозитория> <ветка>
```

где:

- <имя_удаленного_репозитория> - имя удаленного репозитория, на который вы хотите отправить изменения (например, "origin");

- <ветка> - имя ветки, содержащей изменения, которые вы хотите отправить.

Например, если вы хотите отправить изменения из текущей ветки на

удаленный репозиторий с именем "origin", вы можете выполнить следующую команду:

```
git push origin master
```

Чтобы получить изменения с удаленного репозитория, вы можете использовать команду "git pull". Ниже приведен синтаксис этой команды:

```
git pull <имя_удаленного_репозитория> <ветка>
```

где:

- <имя_удаленного_репозитория> - имя удаленного репозитория, с которого вы хотите получить изменения (например, "origin");

- <ветка> - имя ветки, содержащей изменения, которые вы хотите получить.

10) Если хотите получить побольше информации об одном из удалённых репозиториях, вы можете использовать команду `git remote show <remote>`. Она выдаёт URL удалённого репозитория, а также информацию об отслеживаемых ветках. Эта команда любезно сообщает вам, что если вы, находясь на ветке master, выполните `git pull`, ветка master с удалённого сервера будет автоматически влита в вашу сразу после получения всех необходимых данных. Она также выдаёт список всех полученных ею ссылок.

11) Git использует два основных типа тегов: легковесные и аннотированные. Легковесный тег — это что-то очень похожее на ветку, которая не

изменяется — просто указатель на определённый коммит. А вот аннотированные теги хранятся в базе данных Git как полноценные объекты. Они имеют контрольную сумму, содержат имя автора, его e-mail и дату создания, имеют комментарий и могут быть подписаны и проверены с помощью GNU Privacy Guard (GPG). Обычно рекомендуется создавать аннотированные теги, чтобы иметь всю перечисленную информацию; но если вы хотите сделать временную метку или по какой-то причине не хотите сохранять остальную информацию, то для этого годятся и легковесные.

12) Работа с тегами в Git довольно проста. Вот некоторые основные команды:

1. Создание тега: Чтобы создать тег, используйте команду `git tag` с указанием имени тега и коммита, к которому вы хотите привязать тег. Например:

```
git tag <имя тега> <идентификатор коммита>
```

2. Просмотр списка тегов: Чтобы просмотреть список доступных тегов, вы можете использовать команду `git tag` без аргументов: `git tag`

3. Просмотр информации о теге: Чтобы посмотреть дополнительную информацию о конкретном теге, вы можете использовать команду `git show` с указанием имени тега: `git show <имя тега>`

4. Удаление тега: Чтобы удалить тег, используйте команду `git tag` с опцией `-d` и указанием имени тега: `git tag -d <имя тега>`

5. Отправка тегов на удаленный репозиторий: По умолчанию, теги не отправляются на удаленный репозиторий при выполнении команды `git push`. Чтобы отправить один или все теги на удаленный репозиторий, используйте команду `git push` с опцией `--tags`: `git push origin --tags`

13) Флаг `--prune` в командах `git fetch` и `git push` используется для удаления удаленных веток и тегов, которые больше не существуют на удаленном репозитории.

Вот как работает флаг `--prune` в каждой из этих команд:

1. `git fetch --prune`: Команда `git fetch` используется для получения последних изменений из удаленного репозитория. При использовании флага `--prune`, Git будет проверять, какие ветки и теги удалены на удаленном репозитории, и затем удалит соответствующие ссылки на эти удаленные ветки и теги на вашем локальном репозитории. Это полезно, когда ветки или теги удалены на удаленном репозитории, но они все еще присутствуют в вашем локальном репозитории.

2. `git push --prune`: Команда `git push` используется для отправки изменений на удаленный репозиторий. При использовании флага `--prune`, Git будет проверять, какие ветки и теги удалены на вашем локальном репозитории, и

затем удалит соответствующие удаленные ветки и теги на удаленном репозитории. Это полезно, когда вы удалите ветку или тег на вашем локальном репозитории, но они все еще присутствуют на удаленном репозитории.

Использование флага `--prune` позволяет поддерживать синхронизацию между вашим локальным репозиторием и удаленным репозиторием, удаляя ссылки на удаленные ветки и теги.