

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ
ФЕДЕРАЦИИ**
**Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Кафедра инфокоммуникаций

**«Исследование возможностей Git для работы с
локальными репозиториями»**

Отчет по лабораторной работе № 1.2

по дисциплине «Основы программной инженерии»

Выполнил студент группы ПИЖ-б-о-21-1

Трушева В. О. «20» сентября 2022г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверила Воронкин Р.А. _____
(подпись)

Ставрополь 2022

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

Методика и порядок выполнения работы

1. Изучить теоретический материал работы.
2. Создать общедоступный репозиторий на GitHub, в котором будет использована лицензия

MIT и выбранный Вами язык программирования (выбор языка программирования будет доступен после установки флажка Add .gitignore).

Owner * Repository name *

pynikcmd / 2_LR ✓

Great repository names are short and memorable. Need inspiration? How about [vigilant-journey](#)?

Description (optional)

☒ Public
Anyone on the internet can see this repository. You choose who can commit.

☐ Private
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

☒ Add a README file
This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore
Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: C++ ▼

Choose a license
A license tells others what they can and can't do with your code. [Learn more.](#)

License: MIT License ▼

This will set main as the default branch. Change the default name in your [settings](#).

You are creating a public repository in your personal account.

Create repository

Рисунок 1 – Создание нового репозитория

3. Проработайте примеры лабораторной работы. Отрадите вывод на консоли при выполнении команд `git` в отчете для лабораторной работе.

При выполнении команды `git log` в консоли выводится список коммитов с их SHA-1 контрольными суммами, именем и электронной почтой автора, датой создания и сообщением коммита.

```

d:\fgit\1_LR>git log
commit 9962db5f1776d6034412b5c954ef527dec59fe3d (HEAD -> main, origin/main, origin/HEAD) Author: pynikcmd <ver.r762@gmail.com>
Date: Tue Sep 27 01:32:57 2022 +0300

    Добавление строки Завершение

commit a7b07cc8de0794bb13d8e85a7d8b1295a4bc3748
Author: pynikcmd <ver.r762@gmail.com>
Date: Tue Sep 27 01:30:57 2022 +0300

    Добавлена строки года

commit fa5b5d8dc1ae0e4246cb5d4eaa397068d688eb61
Author: pynikcmd <ver.r762@gmail.com>
Date: Tue Sep 27 01:29:00 2022 +0300

    Добавление строки Выполнила студентка

commit 1de76df91e691d8483034ede89070c6e752ea50d
Author: pynikcmd <ver.r762@gmail.com>
Date: Tue Sep 27 01:27:06 2022 +0300

    Добавление строки Лабораторная работа №1

commit 98785108cdb9cb57bfde1d7a21f41e875e01f2e5
Author: pynikcmd <ver.r762@gmail.com>
Date: Tue Sep 27 01:23:48 2022 +0300

    Добавление строки с вычислением 5 + 5
:
commit 9962db5f1776d6034412b5c954ef527dec59fe3d (HEAD -> main, origin/main, origin/HEAD)
Author: pynikcmd <ver.r762@gmail.com>
Date: Tue Sep 27 01:32:57 2022 +0300

    Добавление строки Завершение

commit a7b07cc8de0794bb13d8e85a7d8b1295a4bc3748
Author: pynikcmd <ver.r762@gmail.com>
Date: Tue Sep 27 01:30:57 2022 +0300

    Добавлена строки года

commit fa5b5d8dc1ae0e4246cb5d4eaa397068d688eb61
Author: pynikcmd <ver.r762@gmail.com>
Date: Tue Sep 27 01:29:00 2022 +0300

    Добавление строки Выполнила студентка

```

Рисунок 2 – Результат команды git log

Команда *git -p* показывает разницу, внесенную в каждый коммит. А параметр -2 говорит о том, что будет выведено последние 2 записи.

```

d:\fgit\1_LR>git log -p -2
commit 9962db5f1776d6034412b5c954ef527dec59fe3d (HEAD -> main, origin/main, origin/HEAD)
Author: pynikcmd <ver.r762@gmail.com>
Date: Tue Sep 27 01:32:57 2022 +0300

    Добавление строки Завершение

diff --git "a/\320\236\320\237\320\230\320\236\320\237\320\230\320\236\320\237\320\230.cpp" "b/\320\236\320\237\320\230\320\236\320\237\320\230.cpp"
index 50128c4..5ebc74c 100644
--- "a/\320\236\320\237\320\230\320\236\320\237\320\230\320\236\320\237\320\230.cpp"
+++ "b/\320\236\320\237\320\230\320\236\320\237\320\230\320\236\320\237\320\230.cpp"
@@ -12,5 +12,5 @@ int main()
     cout << "город Ставрополь\n";
     cout << "2022 год";
     cout << "5 + 5 = " << 5 + 5 << "\n";
-
+     cout << "Завершение";
 }

commit a7b07cc8de0794bb13d8e85a7d8b1295a4bc3748
Author: pynikcmd <ver.r762@gmail.com>
Date: Tue Sep 27 01:30:57 2022 +0300

    Добавлена строки года

diff --git "a/\320\236\320\237\320\230\320\236\320\237\320\230\320\236\320\237\320\230.cpp" "b/\320\236\320\237\320\230\320\236\320\237\320\230.cpp"
index 9fbd888..50128c4 100644
--- "a/\320\236\320\237\320\230\320\236\320\237\320\230\320\236\320\237\320\230.cpp"
:~...skipping...
commit 9962db5f1776d6034412b5c954ef527dec59fe3d (HEAD -> main, origin/main, origin/HEAD)
Author: pynikcmd <ver.r762@gmail.com>
Date: Tue Sep 27 01:32:57 2022 +0300

    Добавление строки Завершение

```

Рисунок 3 – Результат команды `git log -p -2`

При использовании команды `git --stat`, на экран выводится под каждым коммитом список и количество измененных файлов, сколько в каждом из файлов было добавлено и удалено.

```

d:\fgit\1_LR>git log --stat
commit 9962db5f1776d6034412b5c954ef527dec59fe3d (HEAD -> main, origin/main, origin/HEAD)
Author: pynikcmd <ver.r762@gmail.com>
Date: Tue Sep 27 01:32:57 2022 +0300

    Добавление строки Завершение

.../\320\236\320\237\320\230\/\320\236\320\237\320\230.cpp" | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

commit a7b07cc8de0794bb13d8e85a7d8b1295a4bc3748
Author: pynikcmd <ver.r762@gmail.com>
Date: Tue Sep 27 01:30:57 2022 +0300

    Добавлена строки года

.../\320\236\320\237\320\230\/\320\236\320\237\320\230.cpp" | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

commit fa5b5d8dc1ae0e4246cb5d4eaa397068d688eb61
Author: pynikcmd <ver.r762@gmail.com>
Date: Tue Sep 27 01:29:00 2022 +0300

    Добавление строки Выполнила студентка

.../\320\236\320\237\320\230\/\320\236\320\237\320\230.cpp" | 2 ++
1 file changed, 2 insertions(+)

commit 1de76df91e691d8483034ede89070c6e752ea50d
Author: pynikcmd <ver.r762@gmail.com>
Date: Tue Sep 27 01:27:06 2022 +0300

    Добавление строки Лабораторная работа №1

.../\320\236\320\237\320\230\/\320\236\320\237\320\230.cpp" | 4 +++-
1 file changed, 3 insertions(+), 1 deletion(-)

commit 98785108cdb9cb57bfde1d7a21f41e875e01f2e5
Author: pynikcmd <ver.r762@gmail.com>
Date: Tue Sep 27 01:23:48 2022 +0300

    Добавление строки с вычислением 5 + 5

.../\320\236\320\237\320\230\/\320\236\320\237\320\230.cpp" | 1 +
1 file changed, 1 insertion(+)

commit 63a8942702e41905c57573552c57289a74e7f814
Author: pynikcmd <ver.r762@gmail.com>
Date: Tue Sep 27 01:22:24 2022 +0300

```

Рисунок 4 – Результат команды git --stat

Опция *pretty* меняет формат вывода *git log*. Опция *online* вывод каждый коммит в одну строку.

```
d:\fgit\1_LR>git log --pretty=oneline
ce94d842fce6eb3a8a67ca98ce4b09096f177df4 (HEAD -> main, origin/main, origin/HEAD) Add files via upload
ea37cc2f34427953594c1dc748c09034795a171b Delete 1_LP_ОПИ.docx
2a5c77225e821c903b32cc0d5782398af1d859cf Add files via upload
9962db5f1776d6034412b5c954ef527dec59fe3d Добавление строки Завершение
a7b07cc8de0794bb13d8e85a7d8b1295a4bc3748 Добавлена строки года
fa5b5d8dc1ae0e4246cb5d4eaa397068d688eb61 Добавление строки Выполнила студентка
1de76df91e691d8483034ede89070c6e752ea50d Добавление строки Лабораторная работа №1
98785108cdb9cb57bfde1d7a21f41e875e01f2e5 Добавление строки с вычислением 5 + 5
63a8942702e41905c57573552c57289a74e7f814 Добавление строки город Ставрополь
707d0358cf85d1f373df65314dba20f5d37a44f2 Добавление строки с группой в программу c++
07cdb6e0803aea7b14bed29bfaa0b27d79a029a2 Добавление в программу c++ фамилия и имя
b3634a3b296bb65b9cfd499180b48f031e319acb Update README.md
b6e3ce8b6ae483aa7ab723a907c7f8b977ee75b9 Merge branch 'main' of https://github.com/pynikcmd/1_LR
c4645da9032113683a71693c53243b144eb0f728 дополнен .gitignore для c++
681138e076211fcfa18ab9557c62e21c9df37946 Create README.md
55a233e21f61229b239eed9906c54f08dc59cee6 Initial commit
```

Рисунок 5 – Результат команды git log --pretty=oneline

Команды git --pretty=format:"%h - %an, %cd : %cn" выводит коммиты чпичок в виде: сокращенный хеш коммита - имя автора, дата коммитера : имя коммитера.

```
d:\fgit\1_LR>git log --pretty=format:"%h - %an, %cd : %cn"
ce94d84 - pynikcmd, Wed Oct 5 23:47:26 2022 +0300 : GitHub
ea37cc2 - pynikcmd, Wed Oct 5 23:43:45 2022 +0300 : GitHub
2a5c772 - pynikcmd, Wed Oct 5 23:42:38 2022 +0300 : GitHub
9962db5 - pynikcmd, Tue Sep 27 01:32:57 2022 +0300 : pynikcmd
a7b07cc - pynikcmd, Tue Sep 27 01:30:57 2022 +0300 : pynikcmd
fa5b5d8 - pynikcmd, Tue Sep 27 01:29:00 2022 +0300 : pynikcmd
1de76df - pynikcmd, Tue Sep 27 01:27:06 2022 +0300 : pynikcmd
9878510 - pynikcmd, Tue Sep 27 01:23:48 2022 +0300 : pynikcmd
63a8942 - pynikcmd, Tue Sep 27 01:22:24 2022 +0300 : pynikcmd
707d035 - pynikcmd, Mon Sep 26 23:16:58 2022 +0300 : pynikcmd
07cdb6e - pynikcmd, Mon Sep 26 23:10:28 2022 +0300 : pynikcmd
b3634a3 - pynikcmd, Mon Sep 26 15:21:28 2022 +0300 : GitHub
b6e3ce8 - pynikcmd, Mon Sep 26 15:16:50 2022 +0300 : pynikcmd
c4645da - pynikcmd, Mon Sep 26 15:13:12 2022 +0300 : pynikcmd
681138e - pynikcmd, Mon Sep 26 01:25:37 2022 +0300 : GitHub
55a233e - pynikcmd, Sat Sep 24 00:44:21 2022 +0300 : GitHub
```

Рисунок 6 – Результат команды git --pretty=format:"%h - %an, %cd : %cn"

Опция граф выводит показывает небольшой граф в формате ASCII, который показывает текущую ветку и историю слияний.

```
d:\fgit\1_LR>git log --pretty=format:"%h - %s" --graph
* ce94d84 - Add files via upload
* ea37cc2 - Delete 1_LP_ОПИ.docx
* 2a5c772 - Add files via upload
* 9962db5 - Добавление строки Завершение
* a7b07cc - Добавлена строки года
* fa5b5d8 - Добавление строки Выполнила студентка
* 1de76df - Добавление строки Лабораторная работа №1
* 9878510 - Добавление строки с вычислением 5 + 5
* 63a8942 - Добавление строки город Ставрополь
* 707d035 - Добавление строки с группой в программу с++
* 07cddb6e - Добавление в программу с++ фамилия и имя
* b3634a3 - Update README.md
* b6e3ce8 - Merge branch 'main' of https://github.com/pynikcmd/1_LR
\
* 681138e - Create README.md
* | c4645da - дополнен .gitignore для с++
|/
* 55a233e - Initial commit
```

Рисунок 7 – Результат команды `git --pretty=format:"%h - %s" --graph`

Для изменения коммита существует команда `git commit --amend`

```
commit e3ced51eb1dd6075b3c47054bdf9a6622a21f3a (HEAD -> main)
Author: pynikcmd <ver.r762@gmail.com>
Date: Sat Oct 15 10:42:52 2022 +0300

    "add strochka skfu(2)"
```

Рисунок 8 – Изменение коммита

Чтобы исключить какой-то файл из индексирования, необходимо ввести команду `git reset HEAD <название файла>`.


```

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   1.txt
        new file:   2.txt

d:\fgit\2_LR>git reset HEAD 2.txt

d:\fgit\2_LR>git status
On branch main
Your branch and 'origin/main' have diverged,
and have 1 and 1 different commits each, respectively.
  (use "git pull" to merge the remote branch into yours)

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

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

```

Рисунок 9 – Результат работы программы git reset HEAD 2.txt

Команда `git remote` выведет названия доступных удалённых репозиториях.

```

d:\fgit\2_LR>git remote
origin

```

Рисунок 10 – Результат команды git remote

Команда `git remote` с ключом `-v` адреса для чтения и записи, привязанные к репозиторию.

```

d:\fgit\2_LR>git remote -v
origin https://github.com/pynikcmd/2_LR.git (fetch)
origin https://github.com/pynikcmd/2_LR.git (push)

```

Рисунок 11 – Результат команды git remote -v

Если хотите получить побольше информации об одном из удалённых репозиториях, вы можете использовать команду `git remote show`.

```
d:\fgit\2_LR>git remote show origin
* remote origin
Fetch URL: https://github.com/pynikcmd/2_LR.git
Push URL: https://github.com/pynikcmd/2_LR.git
HEAD branch: main
Remote branch:
  main tracked
Local branch configured for 'git pull':
  main merges with remote main
Local ref configured for 'git push':
  main pushes to main (local out of date)
```

Рисунок 12 – Результат команды git remote show origin

```
d:\fgit\2_LR>git remote -v
origin https://github.com/pynikcmd/2_LR.git (fetch)
origin https://github.com/pynikcmd/2_LR.git (push)

d:\fgit\2_LR>git remote add new_rep https://github.com/pynikcmd/2_LR

d:\fgit\2_LR>git remote -v
new_rep https://github.com/pynikcmd/2_LR (fetch)
new_rep https://github.com/pynikcmd/2_LR (push)
origin https://github.com/pynikcmd/2_LR.git (fetch)
origin https://github.com/pynikcmd/2_LR.git (push)
```

Рисунок 13 – Результат команды git remote add

Для переименования удалённого репозитория можно выполнить git remote rename.

```
d:\fgit\2_LR>git remote rename new_rep rep2

d:\fgit\2_LR>git remote
origin
rep2
```

Рисунок 14 – Результат команды git remote rename

Если по какой-то причине вы хотите удалить удаленный репозиторий — вы сменили сервер или больше не используете определённое зеркало, или кто-то перестал вносить изменения — вы можете использовать git remote remove.

```
d:\fgit\2_LR>git remote remove rep2  
  
d:\fgit\2_LR>git remote  
origin
```

Рисунок 15 – Результат команды Git remove

4. Выполните клонирование созданного репозитория на рабочий компьютер.

```
d:\fgit>git clone https://github.com/pynikcmd/2_LR.git  
Cloning into '2_LR'...  
remote: Enumerating objects: 49, done.  
remote: Counting objects: 100% (49/49), done.  
remote: Compressing objects: 100% (43/43), done.  
remote: Total 49 (delta 7), reused 38 (delta 5), pack-reused 0  
Receiving objects: 100% (49/49), 11.66 KiB | 5.83 MiB/s, done.  
Resolving deltas: 100% (7/7), done.
```

Рисунок 16 – Копирование репозитория на компьютер

5. Дополните файл .gitignore необходимыми правилами для выбранного языка программирования и интегрированной среды разработки.


 .gitignore Дополнен необходимыми правилами для языка C++

Рисунок 17 – Файл .gitignore

6. Добавьте в файл README.md информацию о дисциплине, группе и ФИО студента, выполняющего лабораторную работу.

README.md

2_LR

Основы программной инженерии ПИЖ-6-о-21-1 Трушева Вероника Олеговна

Рисунок 18 – Дополнен файл README.md

7. Напишите небольшую программу на выбранном Вами языке программирования. Фиксируйте изменения при написании программы в локальном репозитории. Должно быть сделано не менее 7 коммитов, отмеченных не менее 3 тэгами.

```
d:\fgit\2_LR>git tag -a v1.2 -m "my version LB 1.2"
d:\fgit\2_LR>git tag
v1.2
```

Рисунок 19 – Создание первого тэга

```
d:\fgit\2_LR>git tag
v1.2
v1.2.2
v1.2.3
v1.2.4
v1.2.5
v1.2.6
v1.2.7
```

Рисунок 20 – Список тэгов

8. Просмотреть историю (журнал) хранилища командой `git log`. Например, помощью команды `git log --graph --pretty=oneline --abbrev-commit/` Добавить скриншот консоли с выводом в отчет по лабораторной работе.

```
d:\fgit\2_LR>git log --graph --pretty=oneline --abbrev-commit
* f33821f (HEAD -> main, tag: v1.2.7, origin/main, origin/HEAD) Добавлена строка СКФУ
* 6d2f6f4 (tag: v1.2.6) Добавлена строка город Ставрополь
* 5588602 (tag: v1.2.5) Добавлена новая строка
* 3d786cb (tag: v1.2.4) Добавлена строка Трушева Вероника Олеговна
* 90dfb0d (tag: v1.2.3) Добавлена тсрочка Основы программной инженерии
* 9b7cb01 (tag: v1.2.2) Добавлена новая строка
* 5898b04 (tag: v1.2) Добавлен проект VS на языке C++
* c8e7526 Дополнен необходимыми правилами для языка C++
* d886e78 Добавлены ФИО, группа и информация о дисциплине
* 059dd2c Initial commit
```

Рисунок 21 – История хранилища

9. Просмотреть содержимое коммитов командой `git show`, где:

HEAD : последний коммит;

HEAD~1 : предпоследний коммит (и т. д.);

b34a0e : коммит с указанным хэшем.

Отобразите результаты работы этих команд в отчете.

```
d:\fgit\2_LR>git show HEAD
commit f33821fc8fac8a77ead48ee60cc401258f720afb (HEAD -> main, tag: v1.2.7, origin/main, origin/HEAD)
Author: pynikcmd <ver.r762@gmail.com>
Date: Sat Oct 15 10:42:52 2022 +0300

    Добавлена строка СКФУ

diff --git a/2_LR/2_LR/2_LR.cpp b/2_LR/2_LR/2_LR.cpp
index f90c060..5aaa7a9 100644
--- a/2_LR/2_LR/2_LR.cpp
+++ b/2_LR/2_LR/2_LR.cpp
@@ -10,4 +10,5 @@ int main()
     cout << "Трушева Вероника Олеговна";
     cout << "15.10.2022";
     cout << "город Ставрополь";
+    cout << "СКФУ";
 }
\ No newline at end of file
```

Рисунок 22 – Результат работы команды git show HEAD

```
d:\fgit\2_LR>git show HEAD~3
commit 3d786cb46e4214be857cea978d7b3a2eed50fd7c (tag: v1.2.4)
Author: pynikcmd <ver.r762@gmail.com>
Date: Sat Oct 15 10:33:18 2022 +0300

    Добавлена строка Трушева Вероника Олеговна

diff --git a/2_LR/2_LR/2_LR.cpp b/2_LR/2_LR/2_LR.cpp
index e07bbc1..d3e81d8 100644
--- a/2_LR/2_LR/2_LR.cpp
+++ b/2_LR/2_LR/2_LR.cpp
@@ -7,6 +7,6 @@ int main()

     cout << "Лабораторная работа №2";
     cout << "Основы программной инженерии";
-
+    cout << "Трушева Вероника Олеговна";
 }
\ No newline at end of file
```

Рисунок 23 – Результат работы команды git show HEAD~1

```

d:\fgit\2_LR>git show 6d2f6f
commit 6d2f6f4f48be4d7e45745509f7355fd0920cb316 (tag: v1.2.6)
Author: pynikcmd <ver.r762@gmail.com>
Date: Sat Oct 15 10:39:41 2022 +0300

    Добавлена строка город Ставрополь

diff --git a/2_LR/2_LR/2_LR.cpp b/2_LR/2_LR/2_LR.cpp
index 2959181..f90c060 100644
--- a/2_LR/2_LR/2_LR.cpp
+++ b/2_LR/2_LR/2_LR.cpp
@@ -9,4 +9,5 @@ int main()
     cout << "Основы программной инженерии";
     cout << "Трушева Вероника Олеговна";
     cout << "15.10.2022";
+    cout << "город Ставрополь";
 }
\ No newline at end of file

```

Рисунок 24 – Результат работы команды git show 6d2f6f

10. Освойте возможность отката к заданной версии.

```

d:\fgit\2_LR\2_LR>git checkout -- 2_LR.cpp
error: pathspec '2_LR.cpp' did not match any file(s) known to git

d:\fgit\2_LR\2_LR>cd 2_LR

d:\fgit\2_LR\2_LR\2_LR>git checkout -- 2_LR.cpp

d:\fgit\2_LR\2_LR\2_LR>git add .

d:\fgit\2_LR\2_LR\2_LR>git commit -m "Удаление кода из проекта"
[main 6d66bd6] Удаление кода из проекта
1 file changed, 1 insertion(+), 14 deletions(-)

d:\fgit\2_LR\2_LR\2_LR>git reset --hard HEAD~1
HEAD is now at f33821f Добавлена строка СКФУ

```

Рисунок 25 – Результат возможности отката версий

Если удалить код из проекта и применить команду git checkout -- <имя файла>, то код вновь вернется. А если код снова удалить и применить команду commit то, чтобы вернуться к предыдущей версии необходимо ввести команду git reset --hard HEAD~1, и вернется предыдущая версия, то есть крайний коммит перед удалением кода из проекта.

11. Зафиксируйте сделанные изменения.

12. Добавьте отчет по лабораторной работе в формате PDF в папку doc репозитория. Зафиксируйте изменения.

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

Контрольные вопросы.

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

Одним из самых полезных аргументов является `-p` или `--patch`, который показывает разницу (выводит патч), внесенную в каждый коммит. Так же вы можете ограничить количество записей в выводе команды; используйте параметр `-2` для вывода только двух записей.

`--stat` – сокращенная статистика для каждого коммита;

`--pretty` – меняет формат вывода;

`oneline` – выводит каждый коммит в одну строку;

`format` – позволяет указать формат для вывода информации, такие как:

Опция	Описания вывода
<code>%H</code>	Хеш коммита
<code>%h</code>	Сокращенный хеш коммита
<code>%T</code>	Хеш дерева
<code>%t</code>	Сокращенный хеш дерева
<code>%P</code>	Хеш родителей
<code>%p</code>	Сокращенный хеш родителей
<code>%an</code>	Имя автора
<code>%ae</code>	Электронная почта автора
<code>%ad</code>	Дата автора (формат даты можно задать опцией <code>--date=option</code>)
<code>%ar</code>	Относительная дата автора
<code>%cn</code>	Имя коммитера
<code>%ce</code>	Электронная почта коммитера
<code>%cd</code>	Дата коммитера
<code>%cr</code>	Относительная дата коммитера
<code>%s</code>	Содержание

2. Как ограничить вывод при просмотре истории коммитов?

Используя параметр `-n`, где *n* - число коммитов, которые необходимо посмотреть.

3. Как внести изменения в уже сделанный коммит?

Для этого надо внести необходимые изменения, добавить их в индекс и сделать коммит ещё раз, указав параметр `--amend`: `git commit --amend`.

4. Как отменить индексацию файла в Git?

Командой `git reset HEAD <файл>`.

5. Как отменить изменения в файле?

Командой *git checkout -- <файл>*.

6. Что такое удаленный репозиторий Git?

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

7. Как выполнить просмотр удаленных репозиториях данного локального репозитория?

Команда `git remote` выведет названия доступных удалённых репозиториях.

8. Как добавить удаленный репозиторий для данного локального репозитория?

Для того, чтобы добавить удалённый репозиторий и присвоить ему имя (short name), просто выполните команду `git remote add`.

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

Когда вы хотите поделиться своими наработками, вам необходимо отправить их в удалённый репозиторий. Команда для этого действия простая: `git push`.

10. Как выполнить просмотр удаленного репозитория?

Если хотите получить побольше информации об одном из удаленных репозиториях, вы можете использовать команду `git remote show`.

11. Каково назначение тэгов Git?

Как правило, эта функциональность используется для отметки моментов выпуска версий (v1.0, и т. п.). Такие пометки в Git называются тегами.

12. Как осуществляется работа с тэгами Git?

Просмотреть список имеющихся тегов в Git можно очень просто. Достаточно набрать команду `git tag` (параметры `-l` и `--list` опциональны).

Создание аннотированного тега в Git выполняется легко. Самый простой способ — это указать `-a` при выполнении команды `tag`. Опция `-m` задаёт сообщение, которое будет храниться вместе с тегом. Если не указать сообщение, то Git запустит редактор, чтобы вы смогли его ввести.

С помощью команды `git show` вы можете посмотреть данные тега вместе с коммитом.

После создания теги нужно отправлять явно на удалённый сервер. Процесс аналогичен отправке веток — достаточно выполнить команду `git push origin`.

Для удаления тега в локальном репозитории достаточно выполнить команду `git tag -d`.

Существует два способа изъятия тега из внешнего репозитория. Первый способ — это выполнить команду `git push <remote> :refs/tags/<tagname>`.

Второй способ убрать тег из внешнего репозитория более интуитивный `git push origin --delete`.

Если вы хотите получить версии файлов, на которые указывает тег, то вы можете сделать `git checkout` для тега.

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

Если вы хотите получить все данные и одновременно удалить удаленные данные, добавьте флаг `--prune` в команде `git fetch`. Удалите удаленные ветки, у которых нет локального аналога. Например, удаленная ветка `tmp` будет удалена, если локальная ветка с таким же именем больше не существует. Это также учитывает `refspecs`, например `git push --prune remote refs/heads/*:refs/tmp/*`, убедитесь, что удаленный объект `refs/tmp/foo` будет удален, если `refs/heads/foo` он не существует.