

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

**Отчёт о лабораторной работе №2 по дисциплине основы программной
инженерии**

Выполнила:

Нестеренко Тамара Антоновна,
2 курс, группа ПИЖ-б-о-20-1,

Проверил:

Доцент кафедры инфокоммуникаций,
Воронкин Р.А.

Ставрополь, 2021 г.

ВЫПОЛНЕНИЕ

1. Работа с консолью Git
 - 1.1 Просмотр истории коммитов

```
Git CMD - "C:\Program Files\Git\cmd\git.exe" log
...skipping...
commit 92a5c6fab32d38de23105c939ff4ac206851677 (HEAD -> main, origin/main, origin/HEAD)
Author: tamaranesterenko <90053368+tamaranesterenko@users.noreply.github.com>
Date: Fri Sep 10 16:54:38 2021 +0300

    Update README.md

commit 0fd44f95533aae250e59dd6586ace9f1ab4b7f4f
Author: tamaranesterenko <90053368+tamaranesterenko@users.noreply.github.com>
Date: Thu Sep 9 22:09:08 2021 +0300

    Add files via upload

commit bdeb1d67ba7c7ea49750b8ccdd0dbda0d2c34dfd
Author: tamaranesterenko <90053368+tamaranesterenko@users.noreply.github.com>
Date: Thu Sep 9 12:05:12 2021 +0300

    Added the student's data

commit e653afa6940ba0e44b5c00ee4c511ea018d2abb2
Author: tamaranesterenko <t.a.nesterenko>
Date: Thu Sep 9 11:57:41 2021 +0300

    Changed file README.md

commit 05b9662f98914edcb378b93c6c25c1fcd1396c69
Author: tamaranesterenko <t.a.nesterenko>
Date: Thu Sep 9 10:31:00 2021 +0300

    Changed file Hello.txt

commit 5f2df9592b9d670a8f584e0e9093b240ca6537e8
Author: tamaranesterenko <t.a.nesterenko>
Date: Thu Sep 9 10:28:52 2021 +0300

    Changed file Hello.txt

commit 3adee91214a0707ea2aa8f6613cdd052025cfea5
Author: tamaranesterenko <t.a.nesterenko>
Date: Thu Sep 9 10:25:32 2021 +0300

    Rename Nice.txt

commit b413b3f4f0e6380859c6c627760385f8446c6054
Author: tamaranesterenko <t.a.nesterenko>
Date: Thu Sep 9 07:44:03 2021 +0300

    Added a comment to the file README.md

commit 4f2714e6d53f4f5e4d2f3a525f8b3392e0bbbbe0
Author: tamaranesterenko <t.a.nesterenko>
Date: Thu Sep 9 07:41:42 2021 +0300

    Added a comment to the file
```

Рисунок 1 – Результат выполнения команды «git log»

```
C:\Users\тома нестеренко>cd helloworld
C:\Users\тома нестеренко\helloworld>git log -p -2
commit 92a5c6fab32d38de23105c939ff4ac206851677 (HEAD -> main, origin/main, origin/HEAD)
Author: tamaranesterenko <90053368+tamaranesterenko@users.noreply.github.com>
Date: Fri Sep 10 16:54:38 2021 +0300

    Update README.md

diff --git a/README.md b/README.md
index 54e9612..a4946a2 100644
--- a/README.md
+++ b/README.md
@@ -1,5 +1,5 @@
-Работу выполнила Нестеренко Тамара, студентка 2 курса группы ПИЖ-6-о-20-1
+
+#include <iostream>
+#include <cstdlib>
+using namespace std;
@@ -9,4 +9,4 @@ int main()
     cout << "Hello, world! Have a nice day!" << endl; //Программа выведет "Привет, мир! Хорошего дня!"
     return 0;
 }

commit 0fd44f95533aae250e59dd6586ace9f1ab4b7f4f
Author: tamaranesterenko <90053368+tamaranesterenko@users.noreply.github.com>
Date: Thu Sep 9 22:09:08 2021 +0300

    Add files via upload
```

Рисунок 2 – Результат выполнения команды «git log -p -2»

```
Git CMD - "C:\Program Files\Git\cmd\git.exe" log --stat
Update README.md
README.md | 4 ++-
1 file changed, 2 insertions(+), 2 deletions(-)
commit 0fd44f95533aae250e59dd6586ace9f1ab4b7f4f
Author: tamaranesterenko <90053368+tamaranesterenko@users.noreply.github.com>
Date: Thu Sep 9 22:09:08 2021 +0300
    Add files via upload
DOC/LR1.pdf | Bin 0 -> 581003 bytes
1 file changed, 0 insertions(+), 0 deletions(-)
commit bdeb1d67ba7c7ea49750b8ccdd0dbda0d2c34dfd
Author: tamaranesterenko <90053368+tamaranesterenko@users.noreply.github.com>
Date: Thu Sep 9 12:05:12 2021 +0300
    Added the student's data
README.md | 2 +-
1 file changed, 2 insertions(+)
commit e653afa6940ba0e44b5c00ee4c511ea018d2abb2
Author: tamaranesterenko <t.a.nesterenko>
Date: Thu Sep 9 11:57:41 2021 +0300
    Changed file README.md
README.md | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)
commit 05b9662f98914edcb378b93c6c25c1fcd1396c69
Author: tamaranesterenko <t.a.nesterenko>
Date: Thu Sep 9 10:31:00 2021 +0300
    Changed file Hello.txt
Hello.txt | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)
commit 5f2df9592b9d670a8f584e0e9093b240ca6537e8
Author: tamaranesterenko <t.a.nesterenko>
Date: Thu Sep 9 10:28:52 2021 +0300
    Changed file Hello.txt
Hello.txt | 1 +
1 file changed, 1 insertion(+)
```

Рисунок 3 – Результат выполнения команды «git log --stat»

```
Git CMD
DOC/LR1.pdf | Bin 0 -> 581003 bytes
1 file changed, 0 insertions(+), 0 deletions(-)
commit bdeb1d67ba7c7ea49750b8ccdd0dbda0d2c34dfd
Author: tamaranesterenko <90053368+tamaranesterenko@users.noreply.github.com>
Date: Thu Sep 9 12:05:12 2021 +0300
    Added the student's data
README.md | 2 ++
1 file changed, 2 insertions(+)
commit e653afa6940ba0e44b5c00ee4c511ea018d2abb2
C:\Users\тома нестеренко\helloworld>git log --pretty=oneline
92a5c6fab32d38de23105c939ff4ac206851677 (HEAD -> main, origin/main, origin/HEAD) Update README.md
0fd44f95533aae250e59dd6586ace9f1ab4b7f4f Add files via upload
bdeb1d67ba7c7ea49750b8ccdd0dbda0d2c34dfd Added the student's data
e653afa6940ba0e44b5c00ee4c511ea018d2abb2 Changed file README.md
05b9662f98914edcb378b93c6c25c1fcd1396c69 Changed file Hello.txt
5f2df9592b9d670a8f584e0e9093b240ca6537e8 Changed file Hello.txt
3a4ee91214a0707ea2aa8f6613cdd052025cfea5 Rename Nice.txt
b413b3f4f0e6380859c6c627760385f8446c6054 Added a comment to the file README.md
4f2714e6d53f4f5e4d2f3a525f8b3392e0bbbbe0 Added a comment to the file
25bef9e587b08d688a626591074a41ee4a756fac Created file Hello.txt
e3fcc3151fb01e98bf53b2eb4581647a295ea13 Added the code to the file README.md
d62cf8258740210611b860f2d37364a1e14273ba File created Nice.txt
26bf14b41c86113e39e0f32827d10891dd28ebb3 Update README.md
e00312612032ab51fcc3d2efffcbb25b91749ba2 Update README.md
6f8569fb96cf357010ec781acb9639b7edc79bc Update README.md
3f5fb1d605ac86160d6badb1d374d2f699fef0b7 Initial commit
C:\Users\тома нестеренко\helloworld>
```

Рисунок 4 – Результат выполнения команды «git log --pretty=oneline»

```

C:\Users\тома нестеренко\helloworld>git log --pretty=format:"%h - %an, %ar : %s"
92a5c6f - tamaranesterenko, 45 minutes ago : Update README.md
0fd44f9 - tamaranesterenko, 20 hours ago : Add files via upload
bdeb1d6 - tamaranesterenko, 30 hours ago : Added the student's data
e653afa - tamaranesterenko, 30 hours ago : Changed file README.md
05b9662 - tamaranesterenko, 31 hours ago : Changed file Hello.txt
5f2df95 - tamaranesterenko, 31 hours ago : Changed file Hello.txt
3a4ee91 - tamaranesterenko, 31 hours ago : Rename Nice.txt
b413b3f - tamaranesterenko, 34 hours ago : Added a comment to the file README.md
4f2714e - tamaranesterenko, 34 hours ago : Added a comment to the file
25bef9e - tamaranesterenko, 34 hours ago : Created file Hello.txt
e3fcca3 - tamaranesterenko, 34 hours ago : Added the code to the file README.md
d62cf82 - tamaranesterenko, 5 days ago : File created Nice.txt
26bf14b - tamaranesterenko, 5 days ago : Update README.md
e003126 - tamaranesterenko, 5 days ago : Update README.md
6f8569f - tamaranesterenko, 7 days ago : Update README.md
3f5fb1d - tamaranesterenko, 7 days ago : Initial commit

```

Рисунок 5 – Результат выполнения команды «git log--pretty=format:"%h - %an, %ar : %s"»

```

C:\Users\тома нестеренко\helloworld>git log --pretty=format:"%h %s" --graph
* 92a5c6f Update README.md
* 0fd44f9 Add files via upload
* bdeb1d6 Added the student's data
* e653afa Changed file README.md
* 05b9662 Changed file Hello.txt
* 5f2df95 Changed file Hello.txt
* 3a4ee91 Rename Nice.txt
* b413b3f Added a comment to the file README.md
* 4f2714e Added a comment to the file
* 25bef9e Created file Hello.txt
* e3fcca3 Added the code to the file README.md
* d62cf82 File created Nice.txt
* 26bf14b Update README.md
* e003126 Update README.md
* 6f8569f Update README.md
* 3f5fb1d Initial commit

```

Рисунок 6 – Результат выполнения команды «git log --pretty=format:"%h %s" --graph»


```
C:\Users\тома нестеренко\helloworld>git remote -v
origin https://github.com/tamaranesterenko/helloworld.git (fetch)
origin https://github.com/tamaranesterenko/helloworld.git (push)
```

Рисунок 10 – Результат выполнения команды «git remote -v»

```
C:\Users\тома нестеренко\helloworld>git remote show origin
* remote origin
  Fetch URL: https://github.com/tamaranesterenko/helloworld.git
  Push URL: https://github.com/tamaranesterenko/helloworld.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 (up to date)

C:\Users\тома нестеренко\helloworld>
```

Рисунок 11 – Результат выполнения команды «git remote show origin»

1.2 Работа с тегами

```
C:\Users\тома нестеренко\helloworld>git tag -a v1.0 -m "My version 1.0"
C:\Users\тома нестеренко\helloworld>git tag
v1.0
```

Рисунок 12 – Пример добавления и вывод тега

```
Git CMD - "C:\Program Files\Git\cmd\git.exe" show v1.0
tag v1.0
Tagger: tamaranesterenko <t.a.nesterenko>
Date: Thu Sep 16 11:19:48 2021 +0300
My version 1.0
commit 92a5c6fab32d38de23105c939ff4ac206851677 (HEAD -> main, tag: v1.0, origin/main, origin/HEAD)
Author: tamaranesterenko <90053368+tamaranesterenko@users.noreply.github.com>
Date: Fri Sep 10 16:54:38 2021 +0300
    Update README.md
diff --git a/README.md b/README.md
index 54e9612..a4946a2 100644
--- a/README.md
+++ b/README.md
@@ -1,5 +1,5 @@
-Работу выполнила Нестеренко Тамара, студентка 2 курса группы ПИЖ-6-о-20-1
+
#include <iostream>
#include <cstdlib>
using namespace std;
@@ -9,4 +9,4 @@ int main()
cout << "Hello, world! Have a nice day!" << endl; //Программа выведет "Привет, мир! Хорошего дня!"
return 0;
}
```

Рисунок 13 – Результат выполнения команды «git show»


```

C:\Users\тома нестеренко\helloworld>git show e3fcca3
commit e3fcca3151fb01e98bf53b2eb4581647a295ea13
Author: tamaranesterenko <t.a.nesterenko>
Date: Thu Sep 9 07:34:36 2021 +0300

    Added the code to the file README.md

diff --git a/README.md b/README.md
index 43f2702..199e9c5 100644
--- a/README.md
+++ b/README.md
@@ -1,4 +1,10 @@
-# helloworld
-Всё хорошо
-Well
-Nice
+#include <iostream>
+#include <cstdlib>
+using namespace std;
+
+int main()
+{
+    cout << "Hello, world!" << endl;
+    return 0;
+}
+
C:\Users\тома нестеренко\helloworld>

```

Рисунок 16 – Результат выполнения команды «git show» коммит с указанным хэшем

```

C:\Users\тома нестеренко\helloworld>git checkout --README.md
error: unknown option 'README.md'
usage: git checkout [<options>] <branch>
       or: git checkout [<options>] [<branch>] -- <file>...

    -b <branch>          create and checkout a new branch
    -B <branch>          create/reset and checkout a branch
    -l                  create reflog for new branch
    --guess              second guess 'git checkout <no-such-branch>' (default)
    --overlay            use overlay mode (default)
    -q, --quiet          suppress progress reporting
    --recurse-submodules[=<checkout>] control recursive updating of submodules
    --progress           force progress reporting
    -m, --merge           perform a 3-way merge with the new branch
    --conflict <style>   conflict style (merge or diff3)
    -d, --detach          detach HEAD at named commit
    -t, --track           set upstream info for new branch
    -f, --force           force checkout (throw away local modifications)
    --orphan <new-branch> new unparented branch
    --overwrite-ignore   update ignored files (default)
    --ignore-other-worktrees do not check if another worktree is holding the given ref
    -2, --ours            checkout our version for unmerged files
    -3, --theirs          checkout their version for unmerged files
    -p, --patch           select hunks interactively
    --ignore-skip-worktree-bits do not limit pathspecs to sparse entries only
    --pathspec-from-file <file> read pathspec from file
    --pathspec-file-nul   with --pathspec-from-file, pathspec elements are separated with NUL character

```

Рисунок 17 – Пример удаления всех несохраненных изменений

```

C:\Users\тома нестеренко\helloworld>git reset --hard HEAD~1
HEAD is now at 0fd44f9 Add files via upload

```

Рисунок 18 – Пример откатки состояния хранилища к предыдущей версии


```
C:\Users\тома нестеренко\helloworld>git push origin v1.0
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/tamaranesterenko/helloworld.git
 * [new tag]          v1.0 -> v1.0
```

Рисунок 19 – Пример отправки тега на удалённый сервер

```
C:\Users\тома нестеренко\helloworld>git checkout v1.0
Note: switching to 'v1.0'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

    git switch -c <new-branch-name>

Or undo this operation with:

    git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at 92a5c6f Update README.md
```

Рисунок 20 – Результат выполнения команды «git checkout»

```
C:\Users\тома нестеренко\helloworld>git log --graph --pretty=oneline --abbrev-commit
* 0fd44f9 (HEAD) Add files via upload
* bdeb1d6 Added the student's data
* e653afa Changed file README.md
* 05b9662 Changed file Hello.txt
* 5f2df95 Changed file Hello.txt
* 3a4ee91 Rename Nice.txt
* b413b3f Added a comment to the file README.md
* 4f2714e Added a comment to the file
* 25bef9e Created file Hello.txt
* e3fcc33 Added the code to the file README.md
* d62cf82 File created Nice.txt
* 26bf14b Update README.md
* e003126 Update README.md
* 6f8569f Update README.md
* 3f5fb1d Initial commit
```

Рисунок 21 – Результат выполнения команды «git log --graph --pretty=oneline --abbrev-commit»

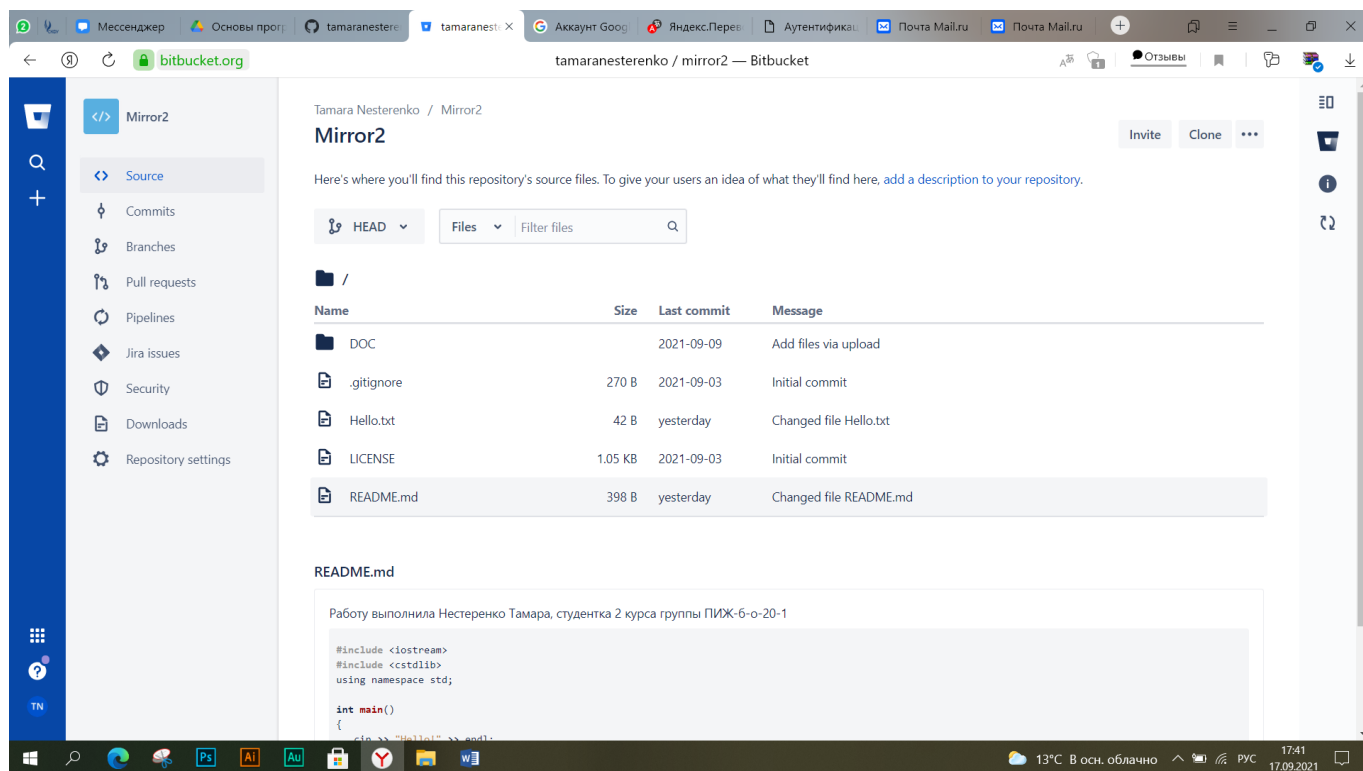


Рисунок 22 – Результат создания зеркала на сервисе bitbucket

Ссылка на репозиторий на bitbucket:

<https://click.mail.ru/redir?u=https%3A%2F%2Fbitbucket.org%2Ftamaranesterenko%2Fmirror2%2Fsrc%2FHEAD%2F&c=swm&r=http&o=mail&v=3&s=d92df1e067a4fe15>

2. 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`.

Опции: Одним из самых полезных аргументов является `-p` или `--patch`, который показывает разницу (выводит патч), внесённую в каждый коммит. Так же вы можете ограничить количество записей в выводе команды; Так же есть возможность использовать серию опций для обобщения. Например, если вы хотите увидеть сокращённую статистику для каждого коммита, вы можете использовать опцию `--stat`. Следующей действительно полезной опцией является `--pretty`. Эта опция 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. Полезные опции для `git log --pretty=format` отображает наиболее полезные опции для изменения формата. Опции

oneline и format являются особенно полезными с опцией --graph команды log . С этой опцией вы сможете увидеть небольшой граф в формате ASCII, который показывает текущую ветку и историю слияний.

2. Как ограничить вывод при просмотре истории коммитов?

Опция	Описание
<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 author совпадает с указанной строкой.
<code>--committer</code>	Показывает только те коммиты, в которых запись committer 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.

3. Как внести изменения в уже сделанный коммит? Если вы хотите переделать коммит - внесите необходимые изменения, добавьте их в индекс и сделайте коммит ещё раз, указав параметр --amend : Эта команда использует область подготовки (индекс) для внесения правок в коммит. Если вы ничего не меняли с момента последнего коммита (например, команда запущена сразу после предыдущего коммита), то снимок состояния останется в точности таким же, а всё что вы сможете изменить — это ваше сообщение к коммиту.

4. Как отменить индексацию файла в Git? Используйте git reset HEAD ... для исключения из индекса.

5. Как отменить изменения в файле? git checkout -- опасная команда. Все локальные изменения в файле пропадут — Git просто заменит его версией из последнего коммита.

6. Что такое удалённый репозиторий Git? Удалённые репозитории представляют собой версии вашего проекта, сохранённые в интернете или ещё где-то в сети.

7. Как выполнить просмотр удалённых репозиториях данного локального репозитория? Для того, чтобы просмотреть список настроенных удалённых репозиториях, вы можете запустить команду git remote . Она выведет названия доступных удалённых репозиториях.

8. Как добавить удалённый репозиторий для данного локального репозитория? Для того, чтобы добавить удалённый репозиторий и присвоить ему имя (shortname), 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 удалённых репозиториях, вы можете использовать команду `git remote show <remote>`.

11. Каково назначение тэгов Git? Как и большинство СКВ, Git имеет возможность пометать определённые моменты в истории как важные. Такие пометки в Git называются тэгами.

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): Данная команда перечисляет теги в алфавитном порядке; порядок их отображения не имеет существенного значения. Так же можно выполнить поиск тега по шаблону. Создание тегов 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. 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.

Создание аннотированного тега в Git. Самый простой способ — это указать - а 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.

Обмен тэгами. Процесс аналогичен отправке веток — достаточно выполнить команду `git push origin <tagname>`. Если у вас много тегов, и вам хотелось бы отправить все за один раз, то можно использовать опцию `--tags` для команды `git push` . В таком случае все ваши теги отправятся на удалённый сервер (если только их уже там нет).

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

Переход на тег. 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?

Для синхронизации текущей ветки с репозиторием используются команды `git fetch` и `git pull`.

`git fetch` — забирает изменения удаленной ветки из репозитория по умолчанию, основной ветки; той, которая была использована при клонировании репозитория. Изменения обновят удаленную ветку (remote tracking branch), после чего надо будет провести слияние с локальной веткой командой `git merge`.

`git prune` – команда является внутренней служебной программой, которая очищает недостижимые или "потерянные" объекты Git. Недостижимыми объектами являются те, которые недоступны по любым ссылкам. Любая фиксация, к которой невозможно получить доступ через ветвь или тег, считается недостижимой. `git prune` как правило, не выполняется напрямую. Команда Prune считается командой сборки мусора и является дочерней командой этой `git gc`команды.

`fetch` обновить только конкретную ветку, а не все.

Важный момент: когда кто-то удаляет ветку из origin, у вас локально всё равно остаётся запись о ней. Например, вы продолжаете видеть origin/deleted/branch, хотя её уже нет. Чтобы эти записи удалить используйте `git fetch origin --prune`.