

**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
Степанов Д.А. .«14» сентября 2022г.

Подпись студента _____

Работа защищена « » _____ 20__г.

Проверила Воронкин Р.А. _____
(подпись)

Ставрополь 2022

№3 Проработайте примеры лабораторной работы. Отрадите вывод на консоли при выполнении команд git в отчете для лабораторной работе.

```

dogim@DESKTOP-E8SKKTS MINGW64 ~/OneDrive/Рабочий стол/lab1/software_engineering_lab2 (main)
$ git log
commit b9e63346c92afc6eacb06c3c64c017ff89844f57 (HEAD -> main, origin/main, origin/HEAD)
Author: Dima <65311242+Lunyawawa@users.noreply.github.com>
Date:   Fri Oct 21 18:17:40 2022 +0300

    Update README.md

commit fc9137f481fe5ed81c4980ab28f3bdc038b05867
Author: Dima <65311242+Lunyawawa@users.noreply.github.com>
Date:   Fri Oct 21 18:17:11 2022 +0300

    Initial commit

dogim@DESKTOP-E8SKKTS MINGW64 ~/OneDrive/Рабочий стол/lab1/software_engineering_lab2 (main)
$ |

```

Рисунок 1 – git log

```

dogim@DESKTOP-E8SKKTS MINGW64 ~/OneDrive/Рабочий стол/lab1/software_engineering_lab2 (main)
$ git log --stat
commit b9e63346c92afc6eacb06c3c64c017ff89844f57 (HEAD -> main, origin/main, origin/HEAD)
Author: Dima <65311242+Lunyawawa@users.noreply.github.com>
Date:   Fri Oct 21 18:17:40 2022 +0300

    Update README.md

    README.md | 3 ++-
    1 file changed, 2 insertions(+), 1 deletion(-)

commit fc9137f481fe5ed81c4980ab28f3bdc038b05867
Author: Dima <65311242+Lunyawawa@users.noreply.github.com>
Date:   Fri Oct 21 18:17:11 2022 +0300

    Initial commit

    .gitignore | 129 +++++
    LICENSE    | 21 +
    README.md  | 1 +
    3 files changed, 151 insertions(+)

dogim@DESKTOP-E8SKKTS MINGW64 ~/OneDrive/Рабочий стол/lab1/software_engineering_lab2 (main)
$ |

```

Рисунок 2 – git log --stat

```

dogim@DESKTOP-E8SKKTS MINGW64 ~/OneDrive/Рабочий стол/lab1/software_engineering_lab2 (main)
$ git log --pretty=oneline
b9e63346c92afc6eacb06c3c64c017ff89844f57 (HEAD -> main, origin/main, origin/HEAD) Update README.md
fc9137f481fe5ed81c4980ab28f3bdc038b05867 Initial commit

dogim@DESKTOP-E8SKKTS MINGW64 ~/OneDrive/Рабочий стол/lab1/software_engineering_lab2 (main)
$ |

```

Рисунок 3 – git log --pretty=oneline

```

dogim@DESKTOP-E8SKKTS MINGW64 ~/OneDrive/Рабочий стол/lab1/software_engineering_lab2 (main)
$ git log --pretty=format:"%h - %an, %ar : %s"
b9e6334 - Dima, 15 minutes ago : Update README.md
fc9137f - Dima, 15 minutes ago : Initial commit

dogim@DESKTOP-E8SKKTS MINGW64 ~/OneDrive/Рабочий стол/lab1/software_engineering_lab2 (main)
$ |

```

Рисунок 4 – git log --pretty=format:"%h - %an, %ar : %s"

```

dogim@DESKTOP-E8SKKTS MINGW64 ~/OneDrive/Рабочий стол/lab1/software_engineering_lab2 (main)
$ git log --pretty=format:"%h %s" --graph
* b9e6334 Update README.md
* fc9137f Initial commit

dogim@DESKTOP-E8SKKTS MINGW64 ~/OneDrive/Рабочий стол/lab1/software_engineering_lab2 (main)
$ |

```

Рисунок 5 – git log --pretty=format:"%h %s" --graph

```

dogim@DESKTOP-E8SKKTS MINGW64 ~/OneDrive/Рабочий стол/lab1/software_engineering_lab2 (main)
$ git log --since=2.weeks
commit b9e63346c92afc6eacb06c3c64c017ff89844f57 (HEAD -> main, origin/main, origin/HEAD)
Author: Dima <65311242+Lunyawawa@users.noreply.github.com>
Date:   Fri Oct 21 18:17:40 2022 +0300

    Update README.md

commit fc9137f481fe5ed81c4980ab28f3bdc038b05867
Author: Dima <65311242+Lunyawawa@users.noreply.github.com>
Date:   Fri Oct 21 18:17:11 2022 +0300

    Initial commit

dogim@DESKTOP-E8SKKTS MINGW64 ~/OneDrive/Рабочий стол/lab1/software_engineering_lab2 (main)
$

```

Рисунок 6 – git log –since=2.weeks

```

dogim@DESKTOP-E8SKKTS MINGW64 ~/OneDrive/Рабочий стол/lab1/software_engineering_lab2 (main)
$ git log -S Степанов
commit b9e63346c92afc6eacb06c3c64c017ff89844f57 (HEAD -> main, origin/main, origin/HEAD)
Author: Dima <65311242+Lunyawawa@users.noreply.github.com>
Date:   Fri Oct 21 18:17:40 2022 +0300

    Update README.md

```

Рисунок 7 – git log -S Степанов

```

dogim@DESKTOP-E8SKKTS MINGW64 ~/OneDrive/Рабочий стол/lab1/software_engineering_lab2 (main)
$ git log -- README.md
commit b9e63346c92afc6eacb06c3c64c017ff89844f57 (HEAD -> main, origin/main, origin/HEAD)
Author: Dima <65311242+Lunyawawa@users.noreply.github.com>
Date:   Fri Oct 21 18:17:40 2022 +0300

    Update README.md

commit fc9137f481fe5ed81c4980ab28f3bdc038b05867
Author: Dima <65311242+Lunyawawa@users.noreply.github.com>
Date:   Fri Oct 21 18:17:11 2022 +0300

    Initial commit

dogim@DESKTOP-E8SKKTS MINGW64 ~/OneDrive/Рабочий стол/lab1/software_engineering_lab2 (main)
$

```

Рисунок 8 – git log – README.md

```

$ git reset HEAD main.py
Unstaged changes after reset:
M       main.py

dogim@DESKTOP-E8SKKTS MINGW64 ~/OneDrive/Рабочий стол/lab1/software_engineering_lab2 (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   .gitignore

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   main.py

```

Рисунок 9 – git reset HEAD <file>

```

$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   main.py

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   .gitignore

dogim@DESKTOP-E8SKKTS MINGW64 ~/OneDrive/Рабочий стол/lab1/software_engineering_lab2 (main)
$ git checkout -- .gitignore

dogim@DESKTOP-E8SKKTS MINGW64 ~/OneDrive/Рабочий стол/lab1/software_engineering_lab2 (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   main.py

```

Рисунок 10 – git checkout – <file>

```

dogim@DESKTOP-E8SKKTS MINGW64 ~/OneDrive/Рабочий стол/lab1/software_engineering_lab2 (main)
$ git remote
origin

dogim@DESKTOP-E8SKKTS MINGW64 ~/OneDrive/Рабочий стол/lab1/software_engineering_lab2 (main)
$ git remote -v
origin https://github.com/Lunyawa/software_engineering_lab2.git (fetch)
origin https://github.com/Lunyawa/software_engineering_lab2.git (push)

dogim@DESKTOP-E8SKKTS MINGW64 ~/OneDrive/Рабочий стол/lab1/software_engineering_lab2 (main)
$

```

Рисунок 11 – git remote

```

dogim@DESKTOP-E8SKKTS MINGW64 ~/OneDrive/Рабочий стол/lab1/software_engineering_lab2 (main)
$ git remote add pb https://github.com/paulboone/ticgit

dogim@DESKTOP-E8SKKTS MINGW64 ~/OneDrive/Рабочий стол/lab1/software_engineering_lab2 (main)
$ git remote -v
origin https://github.com/Lunyawa/software_engineering_lab2.git (fetch)
origin https://github.com/Lunyawa/software_engineering_lab2.git (push)
pb      https://github.com/paulboone/ticgit (fetch)
pb      https://github.com/paulboone/ticgit (push)

dogim@DESKTOP-E8SKKTS MINGW64 ~/OneDrive/Рабочий стол/lab1/software_engineering_lab2 (main)
$

```

Рисунок 12 – git remote add pb <url> и git remote -v

```

dogim@DESKTOP-E8SKKTS MINGW64 ~/OneDrive/Рабочий стол/lab1/software_engineering_lab2 (main)
$ git fetch pb
remote: Enumerating objects: 634, done.
remote: Total 634 (delta 0), reused 0 (delta 0), pack-reused 634
Receiving objects: 100% (634/634), 88.93 KiB | 1.48 MiB/s, done.
Resolving deltas: 100% (261/261), done.
From https://github.com/paulboone/ticgit
* [new branch]      master      -> pb/master
* [new branch]      ticgit      -> pb/ticgit

dogim@DESKTOP-E8SKKTS MINGW64 ~/OneDrive/Рабочий стол/lab1/software_engineering_lab2 (main)
$

```

Рисунок 13 – git fetch <shortname>

```

dogim@DESKTOP-E8SKKTS MINGW64 ~/OneDrive/Рабочий стол/lab1/software_engineering_lab2 (main)
$ git remote show origin
* remote origin
  Fetch URL: https://github.com/Lunyawa/software_engineering_lab2.git
  Push URL: https://github.com/Lunyawa/software_engineering_lab2.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)

```

Рисунок 14 – git remote show origin

```

dogim@DESKTOP-E8SKKTS MINGW64 ~/OneDrive/Рабочий стол/lab1/software_engineering_lab2 (main)
$ git remote rename pb paul
Renaming remote references: 100% (2/2), done.

dogim@DESKTOP-E8SKKTS MINGW64 ~/OneDrive/Рабочий стол/lab1/software_engineering_lab2 (main)
$ git remote
origin
paul

```

Рисунок 15 – git remote rename

```

dogim@DESKTOP-E8SKKTS MINGW64 ~/OneDrive/Рабочий стол/lab1/software_engineering_lab2 (main)
$ git tag -a v1.4 -m "my version 1.4"

dogim@DESKTOP-E8SKKTS MINGW64 ~/OneDrive/Рабочий стол/lab1/software_engineering_lab2 (main)
$ git tag
v1.4

```

Рисунок 16 – git tag -a <name> -m <message> и git tag

```

dogim@DESKTOP-E8SKKTS MINGW64 ~/OneDrive/Рабочий стол/lab1/software_engineering_lab2 (main)
$ git show v1.4
tag v1.4
Tagger: dima <dogimal@mail.ru>
Date: Fri Oct 21 20:34:16 2022 +0300

my version 1.4

commit 2b377ce8c57bb3bcb0c06517c177a922d92a4cef (HEAD -> main, tag: v1.4, origin/main, origin/HEAD)
Author: dima <dogimal@mail.ru>
Date: Fri Oct 21 19:21:25 2022 +0300

    add ignores

diff --git a/.gitignore b/.gitignore
index e4e0ac0..98bf328 100644
--- a/.gitignore
+++ b/.gitignore
@@ -128,4 +128,5 @@ dmpy.json
 # Pyre type checker
 .pyre/

-.idea
\ No newline at end of file
+.idea
+venv
\ No newline at end of file

```

Рисунок 17 – git show <tag_name>

```

dogim@DESKTOP-E8SKKTS MINGW64 ~/OneDrive/Рабочий стол/lab1/software_engineering_lab2 (main)
$ git push origin v1.4
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 156 bytes | 156.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Lunyawa/software_engineering_lab2.git
 * [new tag]          v1.4 -> v1.4

dogim@DESKTOP-E8SKKTS MINGW64 ~/OneDrive/Рабочий стол/lab1/software_engineering_lab2 (main)
$ git push origin --tags
Everything up-to-date

```

Рисунок 18 – git push origin <tag_name>

```

dogim@DESKTOP-E8SKKTS MINGW64 ~/OneDrive/Рабочий стол/lab1/software_engineering_lab2 (main)
$ git tag -d v1.4
Deleted tag 'v1.4' (was f5d444b)

dogim@DESKTOP-E8SKKTS MINGW64 ~/OneDrive/Рабочий стол/lab1/software_engineering_lab2 (main)
$ git push origin :refs/tags/v1.4
To https://github.com/Lunyawa/software_engineering_lab2.git
 - [deleted]          v1.4

```

Рисунок 19 – git tag -d <tag_name>

Выполнение работы:

№8 Просмотреть историю (журнал) хранилища командой git log . Например, с помощью команды git log --graph --pretty=oneline --abbrev-commit . Добавить скриншот консоли с выводом в отчет по лабораторной работе.

```

dogim@DESKTOP-E8SKKTS MINGW64 ~/OneDrive/Рабочий стол/lab1/software_engineering_lab2 (main)
$ git log --graph --pretty=oneline --abbrev-commit
* e9ded24 (HEAD -> main, tag: v3, origin/main, origin/HEAD) afixed version
* 73bd215 easy calculator
* 914c746 (tag: v2) add minus
* 00a84ee add error
* 372d7b5 modified output
* 9d59708 (tag: v1) a + b calculator
* 2b377ce add ignores
* 3a11ffe Hello world
* 3390c9f main
* b9e6334 Update README.md
* fc9137f Initial commit

```

Рисунок 20 - git log --graph --pretty=oneline --abbrev-commit

№9 Просмотреть содержимое коммитов командой git show <ref>

```

$ git show HEAD
commit e9ded24ad7b400d8e8fa053fe0aa6e4bd7fd0627 (HEAD -> main, tag: v3, origin/main, origin/HEAD)
Author: dima <dogimal@mail.ru>
Date:   Fri Oct 21 21:11:03 2022 +0300

    afixed version

diff --git a/main.py b/main.py
index 5c54eec..48af998 100644
--- a/main.py
+++ b/main.py
@@ -6,13 +6,14 @@ def calc(znak, a, b):
     elif (znak == '*'):
         return a * b
     elif (znak == '/'):
-         if (znak == '/'):
-             choice = input("Вам нужно целочисленное деление? (напишите + или -) ")
-             if (choice == '+'):
-                 return a // b
-             else:
-                 return a / b
+         choice = input("Вам нужно целочисленное деление? (напишите + или -) ")
+         if (choice == '+'):
+             return a // b
+         else:
+             return a / b
+     return "ОШИБКА"
+     choice = input("Вам нужно целочисленное деление? (напишите + или -) ")
+     if (b == 0):
+         return "Ошибка на 0 делить нельзя!"
+     if (choice == '+'):
+         return a // b
+     else:
+         return a / b
+     return "ОШИБКА вы ввели не знак"

```

Рисунок 21 – git show HEAD

```

dogim@DESKTOP-E8SKKTS MINGW64 ~/OneDrive/Рабочий стол/lab1/software_engineering_lab2 (main)
$ git show HEAD~1
commit 73bd215f85f51b61ccea385c59b46b929f86edc6
Author: dima <dogimal@mail.ru>
Date:   Fri Oct 21 21:08:14 2022 +0300

    easy calculator

diff --git a/main.py b/main.py
index 6526e20..5c54eec 100644
--- a/main.py
+++ b/main.py
@@ -3,9 +3,21 @@ def calc(znak, a, b):
     return a + b
     elif (znak == '-'):
         return a - b
+     elif (znak == '*'):
+         return a * b
+     elif (znak == '/'):
+         if (znak == '/'):
+             choice = input("Вам нужно целочисленное деление? (напишите + или -) ")
+             if (choice == '+'):
+                 return a // b
+             else:
+                 return a / b
+     return "ОШИБКА"

znak = input("Введите знак -> ")
a = int(input("Введите первое число = "))
b = int(input("Введите второе число = "))

print(calc(znak, a, b))
\ No newline at end of file

```

Рисунок 22 – git show HEAD~1

```

dogim@DESKTOP-E8SKKTS MINGW64 ~/OneDrive/Рабочий стол/lab1/software_engineering_lab2 (main)
$ git show 73bd215
commit 73bd215f85f51b61ccea385c59b46b929f86edc6
Author: dima <dogimal@mail.ru>
Date:   Fri Oct 21 21:08:14 2022 +0300

    easy calculator

diff --git a/main.py b/main.py
index 6526e20..5c54eec 100644
--- a/main.py
+++ b/main.py
@@ -3,9 +3,21 @@ def calc(znak, a, b):
     return a + b
     elif (znak == '-'):
         return a - b
+ elif (znak == '*'):
+     return a * b
+ elif (znak == '/'):
+     if (znak == '/'):
+         choice = input("Вам нужно целочисленное деление? (напишите + или -)" )
+         if (choice == '+'):
+             return a // b
+         else:
+             return a / b
     return "ОШИБКА"

znak = input("Введите знак -> ")
a = int(input("Введите первое число = "))
b = int(input("Введите второе число = "))
+
+
+
print(calc(znak, a, b))
\ No newline at end of file

```

Рисунок 23 – git show <hash>

№10 Освойте возможность отката к заданной версии.

```

dogim@DESKTOP-E8SKKTS MINGW64 ~/OneDrive/Рабочий стол/lab1/software_engineering_lab2 (main)
$ git checkout -- main.py

dogim@DESKTOP-E8SKKTS MINGW64 ~/OneDrive/Рабочий стол/lab1/software_engineering_lab2 (main)
$ git add .

dogim@DESKTOP-E8SKKTS MINGW64 ~/OneDrive/Рабочий стол/lab1/software_engineering_lab2 (main)
$ git commit -m "deleted"
[main 2880a83] deleted
1 file changed, 24 deletions(-)

```

Рисунок 24 – Удаление всех несохраненных изменений в файле и коммит этого файла


```

dogim@DESKTOP-E8SKKTS MINGW64 ~/OneDrive/Рабочий стол/lab1/software_engineering_lab2 (main)
$ git reset --hard HEAD~1
HEAD is now at e9ded24 afixed version

dogim@DESKTOP-E8SKKTS MINGW64 ~/OneDrive/Рабочий стол/lab1/software_engineering_lab2 (main)
$ git show HEAD
commit e9ded24ad7b400d8e8fa053fe0aa6e4bd7fd0627 (HEAD -> main, tag: v3, origin/main, origin/HEAD)
Author: dima <dogimal@mail.ru>
Date:   Fri Oct 21 21:11:03 2022 +0300

    afixed version

diff --git a/main.py b/main.py
index 5c54eec..48af998 100644
--- a/main.py
+++ b/main.py
@@ -6,13 +6,14 @@ def calc(znak, a, b):
     elif (znak == '*'):
         return a * b
     elif (znak == '/'):
-        if (znak == '/'):
-            choice = input("Вам нужно целочисленное деление? (напишите + или -) ")
-            if (choice == '+'):
-                return a // b
-            else:
-                return a / b
-        return "ОШИБКА"
+        choice = input("Вам нужно целочисленное деление? (напишите + или -) ")
+        if (b == 0):
+            return "Ошибка на 0 делить нельзя!"
+        if (choice == '+'):
+            return a // b
+        else:
+            return a / b
+    return "ОШИБКА вы ввели не знак"

znak = input("Введите знак -> ")
a = int(input("Введите первое число = "))

```

Рисунок 25 – Откат к предыдущей версии



Рисунок 26 – Все теги

afixed version	Lunyawa committed 1 hour ago	e9ded24	<>
easy calculator	Lunyawa committed 1 hour ago	73bd215	<>
add minus	Lunyawa committed 1 hour ago	914c746	<>
add error	Lunyawa committed 1 hour ago	00a84ee	<>
modified output	Lunyawa committed 1 hour ago	372d7b5	<>
a + b calculator	Lunyawa committed 1 hour ago	9d59708	<>
add ignores	Lunyawa committed 3 hours ago	2b377ce	<>
Hello world	Lunyawa committed 3 hours ago	3a11ffe	<>
main	Lunyawa committed 3 hours ago	3390c9f	<>
Update README.md	Lunyawa committed 4 hours ago	Verified b9e6334	<>
Initial commit	Lunyawa committed 4 hours ago	Verified fc9137f	<>

Рисунок 27 – Все коммиты

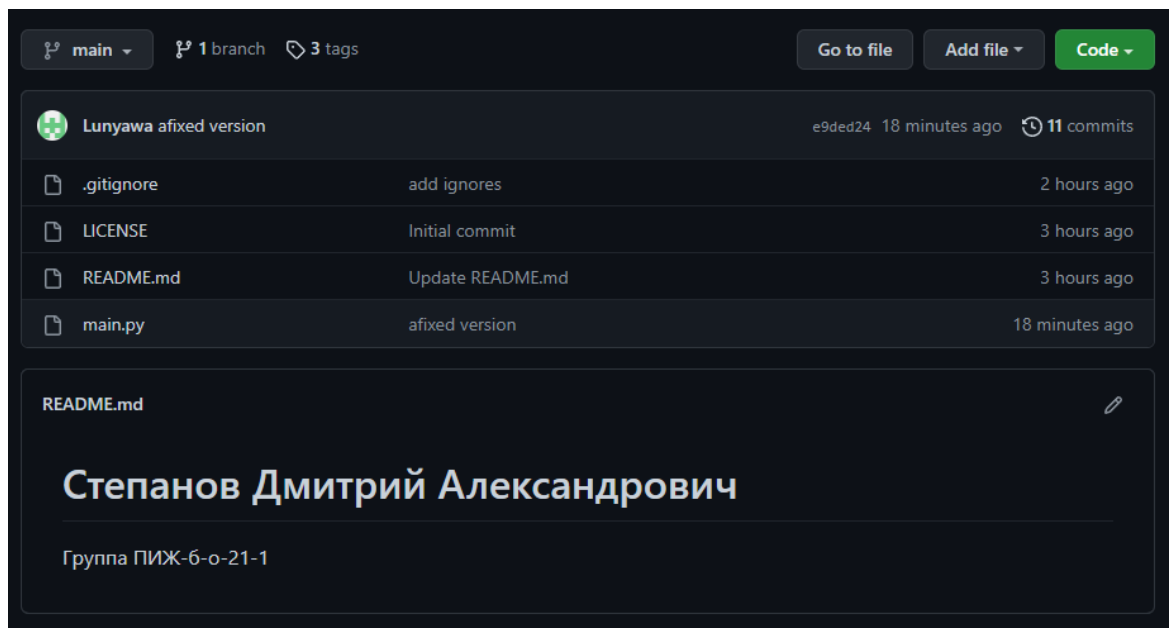


Рисунок 28 – Репозиторий

1. Вопросы

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), внесенную 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 сокращенную 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 формат

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 <code>n</code> 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.

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.

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

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 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?

Для того, чтобы просмотреть список настроенных удалённых репозиториях, вы можете запустить команду `git remote`. Она 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 удалённых репозиториях, вы можете использовать команду `git remote show<remote>`.

11. Каково назначение тэгов Git?

Как и большинство СКВ, Git имеет возможность пометить определённые моменты в истории как важные. 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): 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 тегами. Процесс аналогичен отправке веток — достаточно выполнить команду `git push origin <tag name>`. 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 локальной 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?

`--prune`

Если вы хотите получить все данные и одновременно удалить удалённые данные, добавьте флаг `--prune` в команду `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 `gitpush -- prune remote refs/heads/*:refs/tmp/*`, убедитесь, что удаленный объект refs/tmp/foo будет удален, если refs/heads/foo он не существует.