

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ
ФЕДЕРАЦИИ**

**Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРОКАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Кафедра инфокоммуникаций
Институт цифрового развития
ОТЧЁТ**

по лабораторной работе №1.2

Дисциплина: «Основы кроссплатформенного программирования»

Тема: «Исследование возможностей Git для работы с локальным
репозиторием»

Выполнил: студент 1 курса

группы ИВТ-б-о-21-1

Толубаев Рамиль Ахметович

Ставрополь 2022

Ход работы:

1) Создал новый репозиторий со всеми дополнениями:

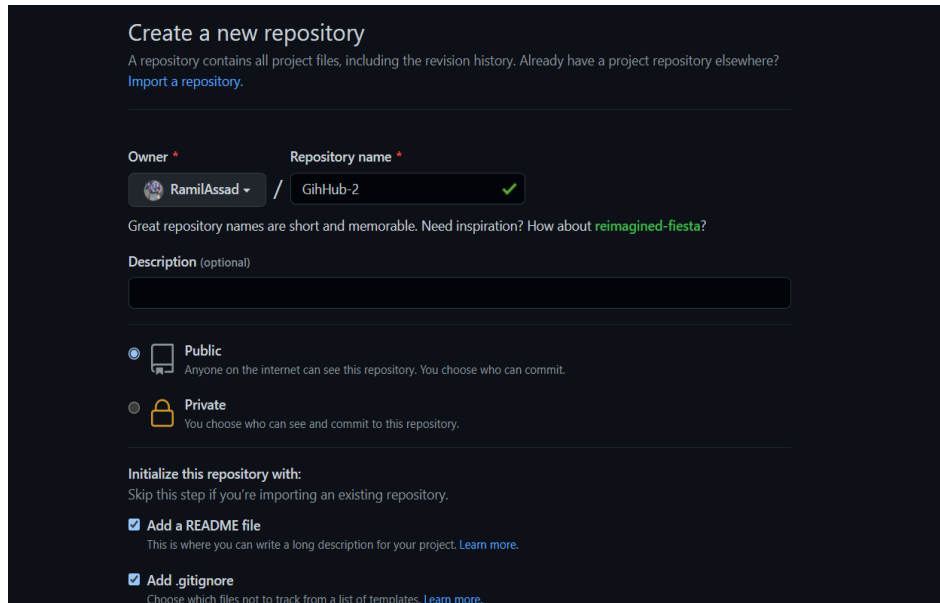


Рисунок 1. Новый репозиторий

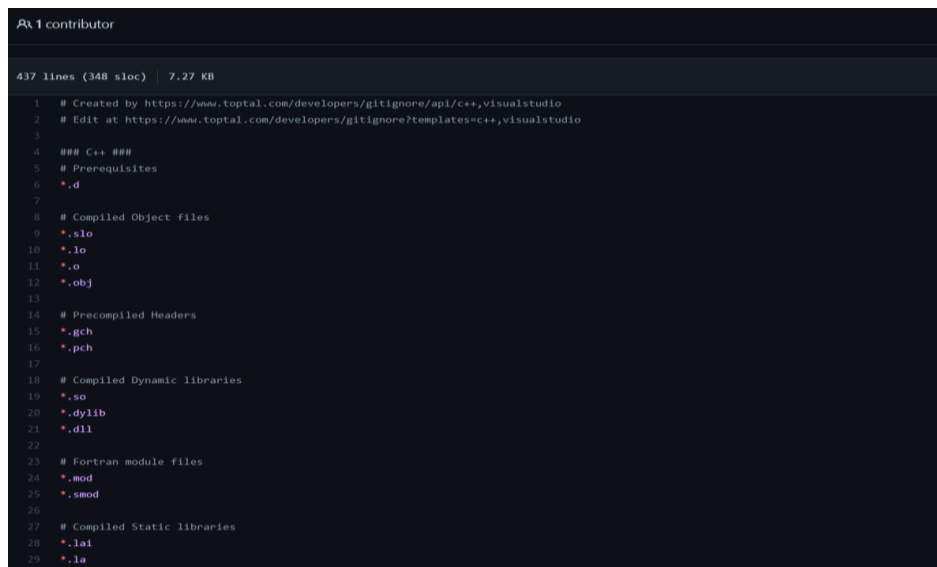


Рисунок 2. Gitignore

2) Клонировал репозитории на рабочий компьютер

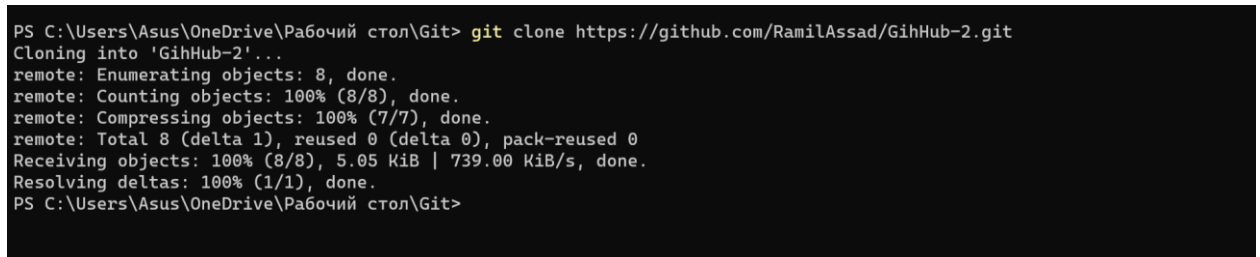


Рисунок 3. Работа в консоли

3) Изменил информацию в README, сделал коммит и запустил на удаленный репозитории

```
Windows PowerShell
(C) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

Установите последнюю версию PowerShell для новых функций и улучшения! https://aka.ms/PSWindows

PS C:\Users\Asus\OneDrive\Рабочий стол\Git\GihHub-2> git status
On branch main
Your branch is up to date with 'origin/main'.

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:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
PS C:\Users\Asus\OneDrive\Рабочий стол\Git\GihHub-2> git add .
PS C:\Users\Asus\OneDrive\Рабочий стол\Git\GihHub-2> 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:   README.md

PS C:\Users\Asus\OneDrive\Рабочий стол\Git\GihHub-2>
```

Рисунок 4. Изменение в README

```
Windows PowerShell
(C) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

Установите последнюю версию PowerShell для новых функций и улучшения! https://aka.ms/PSWindows

PS C:\Users\Asus\OneDrive\Рабочий стол\Git\GihHub-2> git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
PS C:\Users\Asus\OneDrive\Рабочий стол\Git\GihHub-2> git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 365 bytes | 121.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/RamilAssad/GihHub-2.git
   9fdd686..362ee95  main -> main
PS C:\Users\Asus\OneDrive\Рабочий стол\Git\GihHub-2>
```

Рисунок 5. Пуш изменения

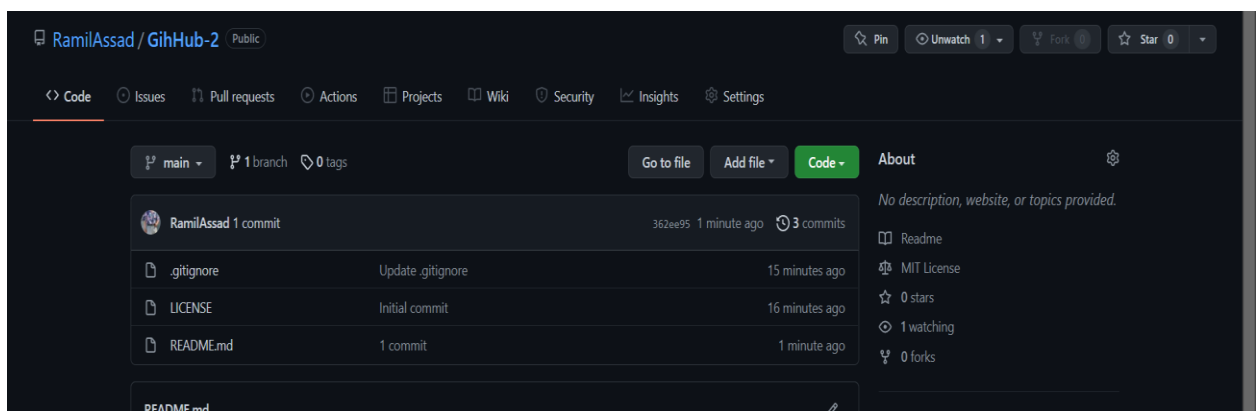


Рисунок 6. Удаленный репозитории

4)

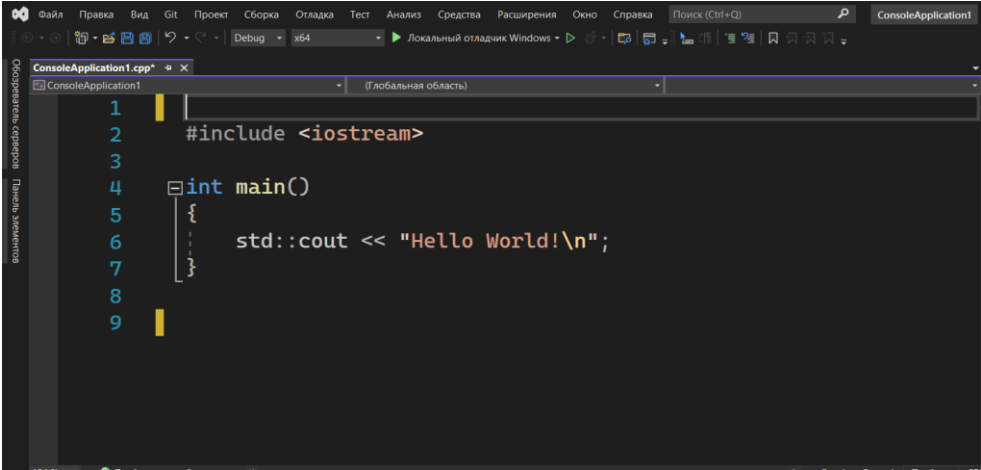


Рисунок 7. Программа на С++

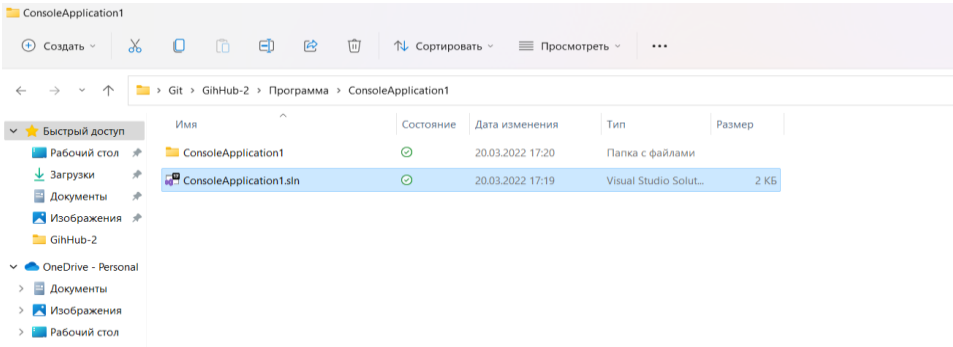


Рисунок 8. Папка с программой

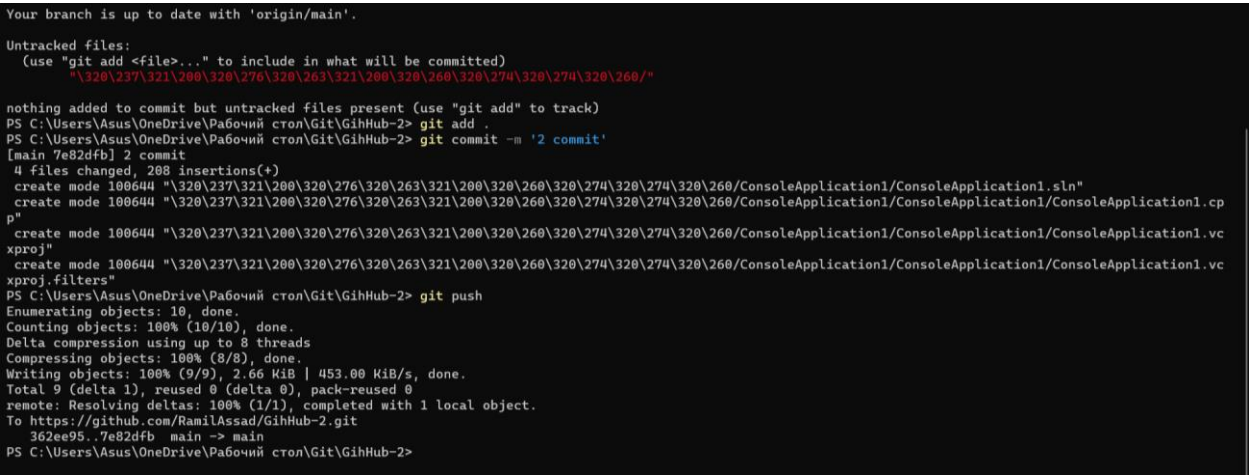


Рисунок 9. Работа в консоли

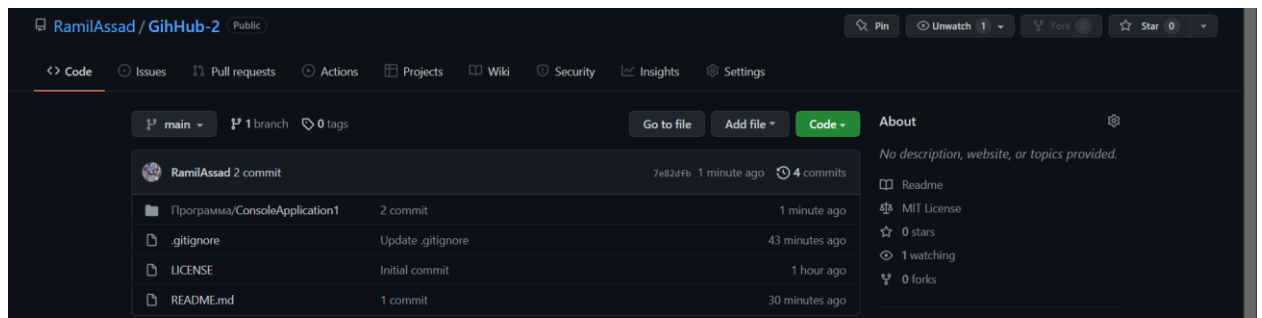


Рисунок 10. GitHub

5) Сделал коммиты в процессе изменения программы, отметил их тегами и запустил

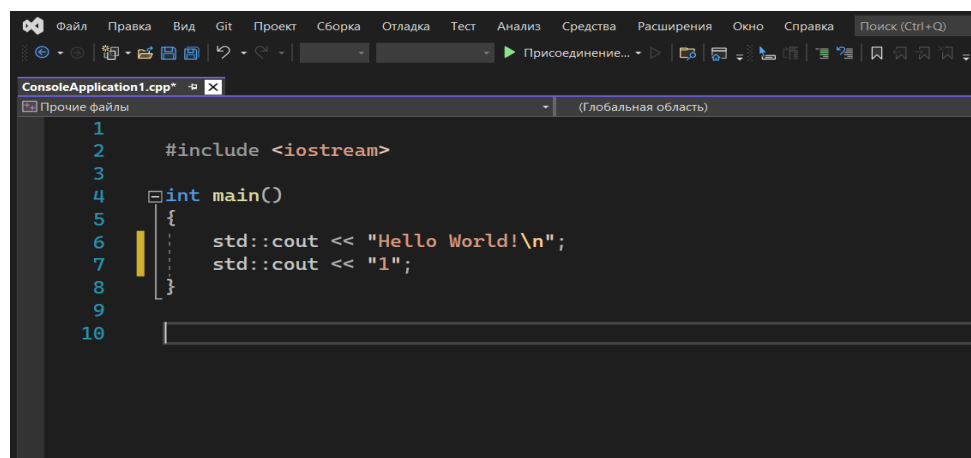


Рисунок 11. Изменение в программе

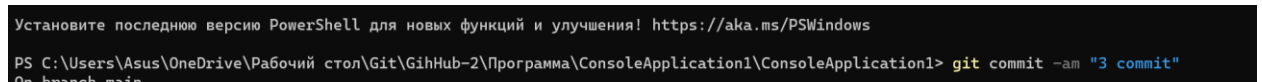


Рисунок 12. Коммит изменения

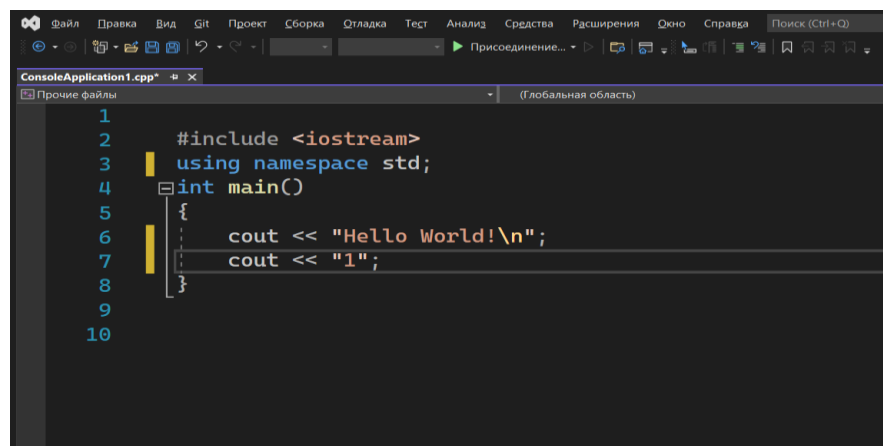


Рисунок 13. Изменение в программе

```
Windows PowerShell
(C) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

Установите последнюю версию PowerShell для новых функций и улучшения! https://aka.ms/PSWindows

PS C:\Users\Asus\OneDrive\Рабочий стол\Git\GihHub-2\Программа\ConsoleApplication1\ConsoleApplication1> git commit -am " 4 commit"
[main 261844f] 4 commit
1 file changed, 3 insertions(+), 3 deletions(-)
PS C:\Users\Asus\OneDrive\Рабочий стол\Git\GihHub-2\Программа\ConsoleApplication1\ConsoleApplication1>
```

Рисунок 14. Коммит изменения

```
File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q)
ConsoleApplication1.cpp
1
2     #include <iostream>
3     using namespace std;
4     int main()
5     {
6         cout << "Hello World!\n";
7         cout << "1";
8         int a, b;
9         a = 12;
10        b = 13;
11    }
```

Рисунок 15. Изменение в программе

```
Windows PowerShell
(C) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

Установите последнюю версию PowerShell для новых функций и улучшения! https://aka.ms/PSWindows

PS C:\Users\Asus\OneDrive\Рабочий стол\Git\GihHub-2\Программа\ConsoleApplication1\ConsoleApplication1> git commit -am " 5 commit"
[main 043d1f0] 5 commit
1 file changed, 3 insertions(+)
PS C:\Users\Asus\OneDrive\Рабочий стол\Git\GihHub-2\Программа\ConsoleApplication1\ConsoleApplication1>
```

Рисунок 16. Коммит изменения

```
File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q)
ConsoleApplication1.cpp
1
2     #include <iostream>
3     using namespace std;
4     int main()
5     {
6         cout << "Hello World!\n";
7         cout << "1";
8         int a, b;
9         a = 12;
10        b = 13;
11        cout << "a = " << a;
12    }
13
14
```

Рисунок 15. Изменение в программе

```
Windows PowerShell
(C) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

Установите последнюю версию PowerShell для новых функций и улучшения! https://aka.ms/PSWindows

PS C:\Users\Asus\OneDrive\Рабочий стол\Git\GihHub-2\Программа\ConsoleApplication1\ConsoleApplication1> git commit -am "6 commit"
[main b2c2404] 6 commit
1 file changed, 1 insertion(+)
PS C:\Users\Asus\OneDrive\Рабочий стол\Git\GihHub-2\Программа\ConsoleApplication1\ConsoleApplication1>
```

Рисунок 16. Коммит изменения

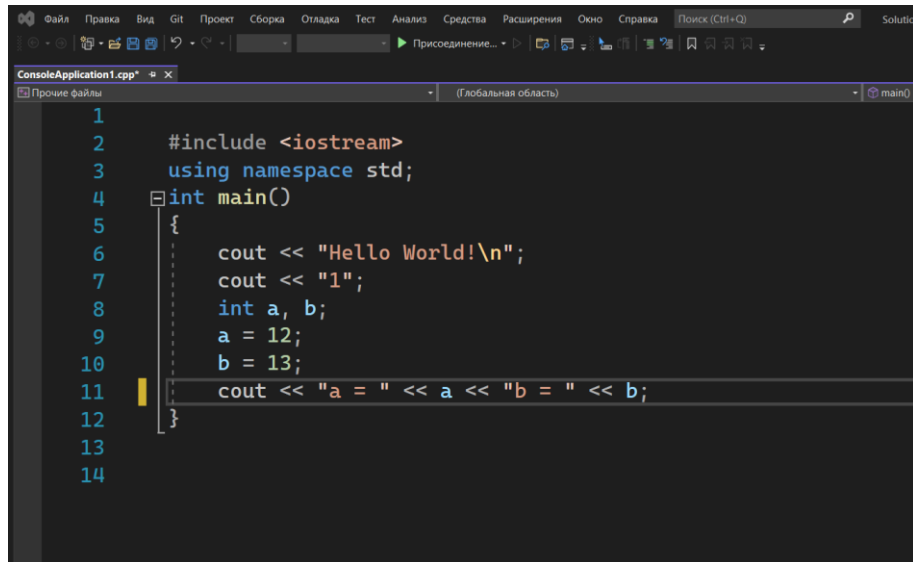


Рисунок 17. Изменение в программе

```
Windows PowerShell
(C) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

Установите последнюю версию PowerShell для новых функций и улучшения! https://aka.ms/PSWindows

PS C:\Users\Asus\OneDrive\Рабочий стол\Git\GihHub-2\Программа\ConsoleApplication1\ConsoleApplication1> git commit -am "7commit"
[main 8e1af8f] 7commit
1 file changed, 1 insertion(+), 1 deletion(-)
PS C:\Users\Asus\OneDrive\Рабочий стол\Git\GihHub-2\Программа\ConsoleApplication1\ConsoleApplication1>
```

Рисунок 18. Коммит изменения

```
Windows PowerShell
(C) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

Установите последнюю версию PowerShell для новых функций и улучшения! https://aka.ms/PSWindows

PS C:\Users\Asus\OneDrive\Рабочий стол\Git\GihHub-2\Программа\ConsoleApplication1\ConsoleApplication1> git push
Enumerating objects: 34, done.
Counting objects: 100% (34/34), done.
Delta compression using up to 8 threads
Compressing objects: 100% (25/25), done.
Writing objects: 100% (30/30), 2.20 KiB | 374.00 KiB/s, done.
Total 30 (delta 17), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (17/17), completed with 2 local objects.
To https://github.com/RamilAssad/GihHub-2.git
7e82dfb..8e1af8f main -> main
PS C:\Users\Asus\OneDrive\Рабочий стол\Git\GihHub-2\Программа\ConsoleApplication1\ConsoleApplication1>
```

Рисунок 19. Пуш изменений

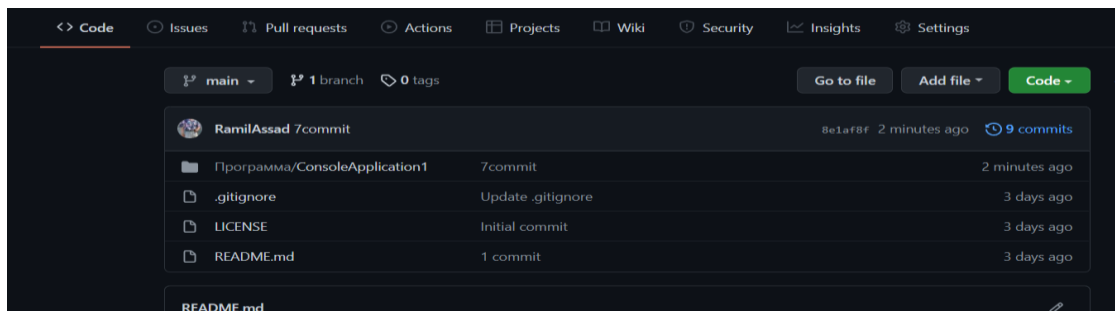


Рисунок 20. Изменение на GitHub

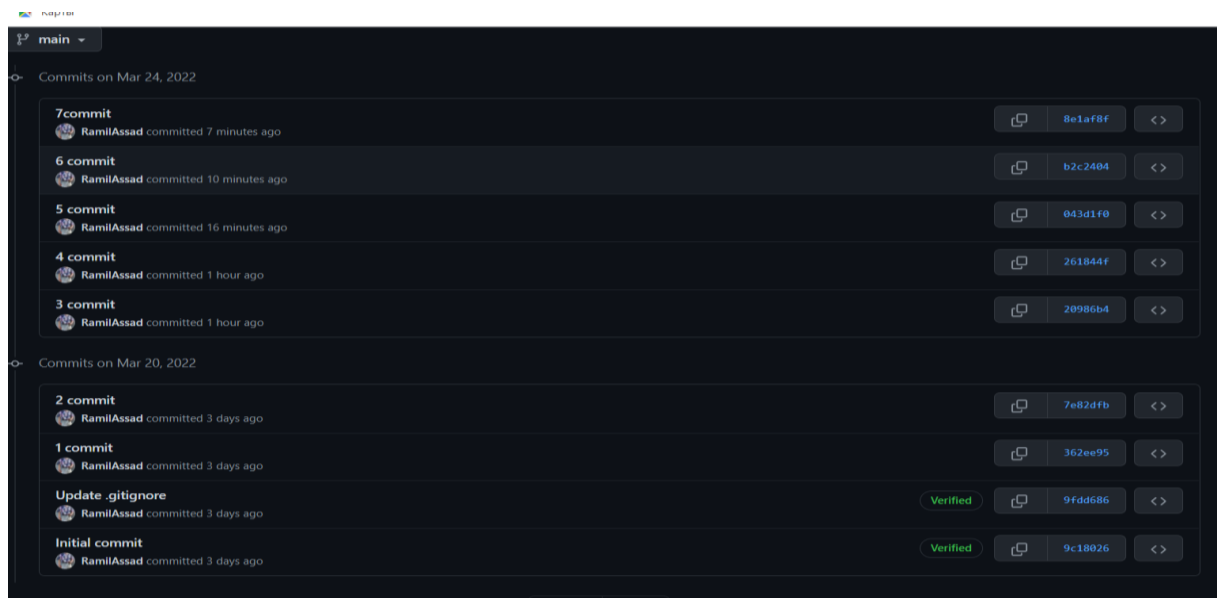


Рисунок 21. История коммитов

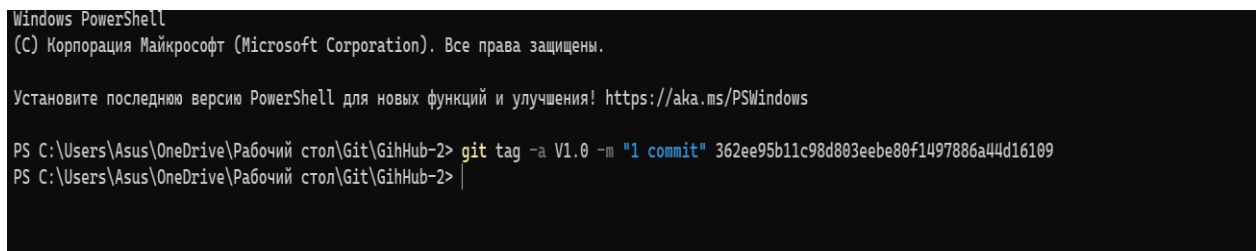


Рисунок 22. Присваивание тег к коммиту

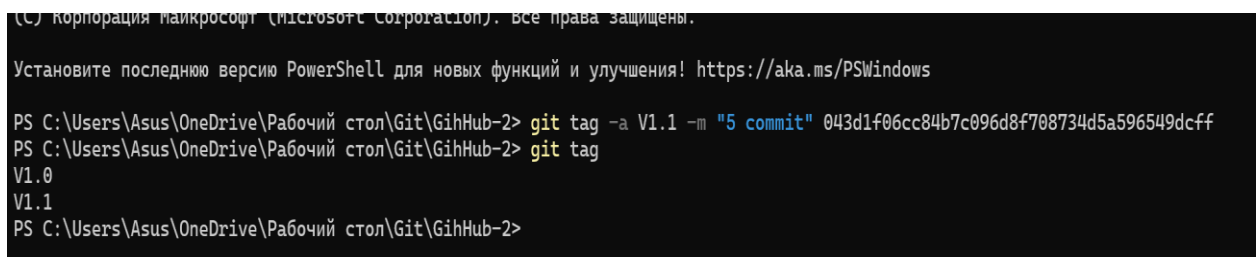


Рисунок 23. Присваивание тег к коммиту


```
Windows PowerShell
(C) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

Установите последнюю версию PowerShell для новых функций и улучшения! https://aka.ms/PSWindows

PS C:\Users\Asus\OneDrive\Рабочий стол\Git\GihHub-2> git tag -a V2.0 -m "7commit" 8e1af8f204d72808d122e8e2e1860652db87979e
PS C:\Users\Asus\OneDrive\Рабочий стол\Git\GihHub-2> git tag
V1.0
V1.1
V2.0
PS C:\Users\Asus\OneDrive\Рабочий стол\Git\GihHub-2> |
```

Рисунок 24. Присваивание тег к коммиту

```
Windows PowerShell
(C) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

Установите последнюю версию PowerShell для новых функций и улучшения! https://aka.ms/PSWindows

PS C:\Users\Asus\OneDrive\Рабочий стол\Git\GihHub-2> git push origin --tags
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 419 bytes | 419.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/RamilAssad/GihHub-2.git
 * [new tag]          V1.0 -> V1.0
 * [new tag]          V1.1 -> V1.1
 * [new tag]          V2.0 -> V2.0
PS C:\Users\Asus\OneDrive\Рабочий стол\Git\GihHub-2>
```

Рисунок 25. Пуш тегов

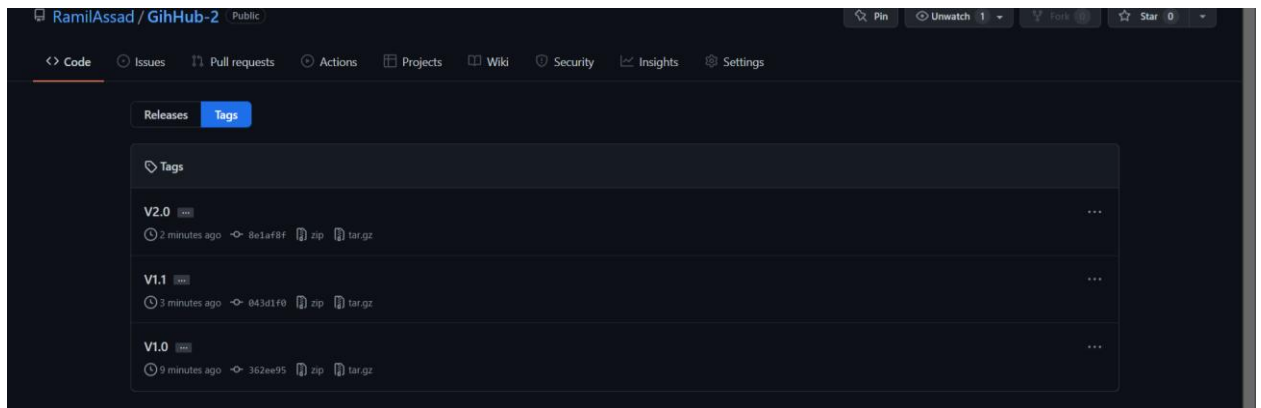


Рисунок 26. История тегов на удаленном сервере

```
Установите последнюю версию PowerShell для новых функций и улучшения! https://aka.ms/PSWindows

PS C:\Users\Asus\OneDrive\Рабочий стол\Git\GihHub-2> git log --graph --pretty=oneline --abbrev-commit
* 8e1af8f (HEAD -> main, tag: V2.0, origin/main, origin/HEAD) 7commit
* b2c2404 6 commit
* 043d1f0 (tag: V1.1) 5 commit
* 261844f 4 commit
* 20986b4 3 commit
* 7e82dfb 2 commit
* 362ee95 (tag: V1.0) 1 commit
* 9fdd686 Update .gitignore
* 9c18026 Initial commit
PS C:\Users\Asus\OneDrive\Рабочий стол\Git\GihHub-2>
```

Рисунок 27. История коммитов с помощью git log

```

PS C:\Users\Asus\OneDrive\Рабочий стол\Git\GihHub-2> git show HEAD
commit 8e1af8f204d72808d122e8e2e1860652db87979e (HEAD -> main, tag: V2.0, origin/main, origin/HEAD)
Author: RamilAssad <ramil_tolubaev@mail.ru>
Date: Thu Mar 24 03:26:55 2022 +0300

    7commit

diff --git "a/\320\237\321\200\320\276\320\263\321\200\320\260\320\274\320\274\320\260\ConsoleApplication1\ConsoleApplication1\ConsoleApplication1.cpp" "b/\320\237\321\200\320\276\320\263\321\200\320\260\320\274\320\274\320\260\ConsoleApplication1\ConsoleApplication1\ConsoleApplication1.cpp"
index 29b01fb..0a2e9ec 100644
--- "a/\320\237\321\200\320\276\320\263\321\200\320\260\320\274\320\274\320\260\ConsoleApplication1\ConsoleApplication1\ConsoleApplication1.cpp"
+++ "b/\320\237\321\200\320\276\320\263\321\200\320\260\320\274\320\274\320\260\ConsoleApplication1\ConsoleApplication1\ConsoleApplication1.cpp"
@@ -8,6 +8,6 @@ int main()
     int a, b;
     a = 12;
     b = 13;
-    cout << "a = " << a;
+    cout << "a = " << a << "b = " << b;
}

```

Рисунок 28. Команда git show HEAD

```

PS C:\Users\Asus\OneDrive\Рабочий стол\Git\GihHub-2> git show HEAD~
commit b2c2404112a555ed1ed8b806f7b93295e4676fd
Author: RamilAssad <ramil_tolubaev@mail.ru>
Date: Thu Mar 24 03:24:12 2022 +0300

    6 commit

diff --git "a/\320\237\321\200\320\276\320\263\321\200\320\260\320\274\320\274\320\260\ConsoleApplication1\ConsoleApplication1\ConsoleApplication1.cpp" "b/\320\237\321\200\320\276\320\263\321\200\320\260\320\274\320\274\320\260\ConsoleApplication1\ConsoleApplication1\ConsoleApplication1.cpp"
index f7dd5fb..29b01fb 100644
--- "a/\320\237\321\200\320\276\320\263\321\200\320\260\320\274\320\274\320\260\ConsoleApplication1\ConsoleApplication1\ConsoleApplication1.cpp"
+++ "b/\320\237\321\200\320\276\320\263\321\200\320\260\320\274\320\274\320\260\ConsoleApplication1\ConsoleApplication1\ConsoleApplication1.cpp"
@@ -8,5 +8,6 @@ int main()
     int a, b;
     a = 12;
     b = 13;
+    cout << "a = " << a;
}

```

Рисунок 29. Команда git show HEAD~

```

(C) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

Установите последнюю версию PowerShell для новых функций и улучшения! https://aka.ms/PSWindows

PS C:\Users\Asus\OneDrive\Рабочий стол\Git\GihHub-2> git show 043d1f06cc84b7c096d8f708734d5a596549dcff
commit 043d1f06cc84b7c096d8f708734d5a596549dcff (tag: V1.1)
Author: RamilAssad <ramil_tolubaev@mail.ru>
Date: Thu Mar 24 03:18:31 2022 +0300

    5 commit

diff --git "a/\320\237\321\200\320\276\320\263\321\200\320\260\320\274\320\274\320\260\ConsoleApplication1\ConsoleApplication1\ConsoleApplication1.cpp" "b/\320\237\321\200\320\276\320\263\321\200\320\260\320\274\320\274\320\260\ConsoleApplication1\ConsoleApplication1\ConsoleApplication1.cpp"
index a9f56e0..f7dd5fb 100644
--- "a/\320\237\321\200\320\276\320\263\321\200\320\260\320\274\320\274\320\260\ConsoleApplication1\ConsoleApplication1\ConsoleApplication1.cpp"
+++ "b/\320\237\321\200\320\276\320\263\321\200\320\260\320\274\320\274\320\260\ConsoleApplication1\ConsoleApplication1\ConsoleApplication1.cpp"
@@ -5,5 +5,8 @@ int main()
{
    cout << "Hello World!\n";
    cout << "1";
+   int a, b;
+   a = 12;
+   b = 13;
}

```

Рисунок 30. Команда git show 043d1f0

б) Откат к заданной версии

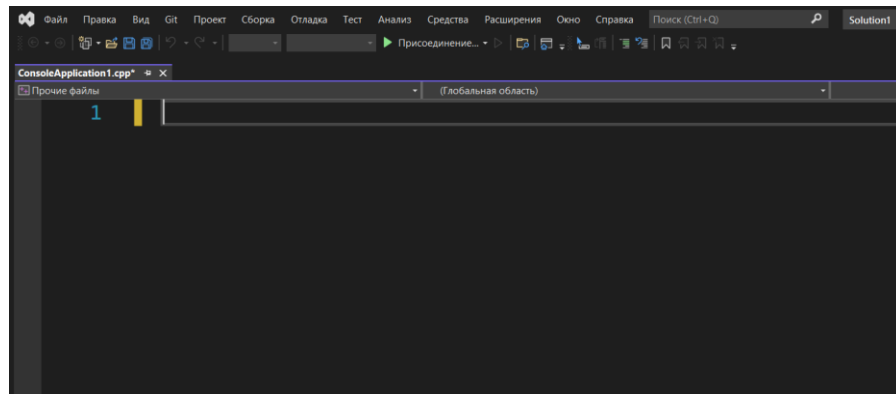


Рисунок 31. Удаление кода в файле

```
C:\Users\Asus\OneDrive\Рабочий стол\Git\GihHub-2\Программа\ConsoleApplication1\ConsoleApplication1> git checkout -- ConsoleApplication1.cpp
C:\Users\Asus\OneDrive\Рабочий стол\Git\GihHub-2\Программа\ConsoleApplication1\ConsoleApplication1> |
```

Рисунок 32. Команда checkout

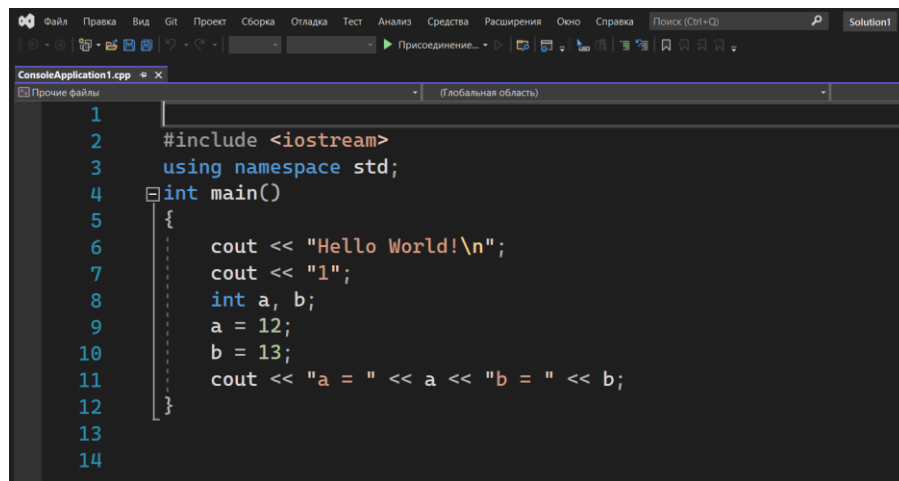


Рисунок 33. Изменение после команды checkout

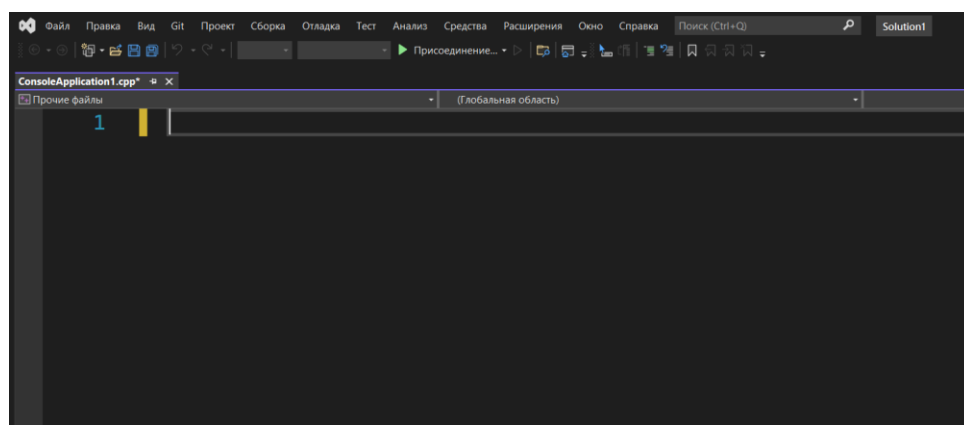


Рисунок 34. Удаление кода в файле

```
Windows PowerShell
(C) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

Установите последнюю версию PowerShell для новых функций и улучшения! https://aka.ms/PSWindows

PS C:\Users\Asus\OneDrive\Рабочий стол\Git\GihHub-2\Программа\ConsoleApplication1\ConsoleApplication1> git commit -am 'Delete 8 commit'
[main 8c61585] Delete 8 commit
1 file changed, 1 insertion(+), 13 deletions(-)
PS C:\Users\Asus\OneDrive\Рабочий стол\Git\GihHub-2\Программа\ConsoleApplication1\ConsoleApplication1>
```

Рисунок 35. Коммит изменений

Вывод: Команда `git checkout --<имя_файла>` удаляет изменения в репозитории по коммита.

Контрольные вопросы:

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

Наиболее простой и в то же время мощный инструмент для этого — команда `git log`. По умолчанию, без аргументов, `git log` выводит список коммитов созданных в данном репозитории в обратном хронологическом порядке. То есть самые последние коммиты показываются первыми.

Одна из опций, когда вы хотите увидеть сокращенную статистику для каждого коммита, вы можете использовать опцию `–stat`.

Вторая опция (одна 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 (пример команды `git log -p -2`).

Третья действительно полезная опция это `--pretty`. Она 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. Особенно это может быть полезным, когда вы хотите сгенерировать вывод для автоматического анализа — так как вы указываете формат явно, он не будет изменен даже после обновления Git.

Для опции `git log --pretty=format` существуют различного рода опции для изменения формата отображения.

2. Как ограничить вывод при просмотре истории коммитов?

Для ограничения может использоваться функция `git log <n>`, где `n` число записей.

Также, существуют опции для ограничения вывода по времени, такие как `--since` и `--until`, они являются очень удобными. Например, следующая команда покажет список коммитов, сделанных за последние две недели:

```
git log --since=2.weeks
```

Это команда работает с большим количеством форматов — вы можете указать определенную дату вида `2008-01-15` или же относительную дату, например `2 years 1 day 3 minutes ago`.

Также вы можете фильтровать список коммитов по заданным параметрам.

Опция `--author` дает возможность фильтровать по автору коммита, а

опция `--grep` (показывает только коммиты, сообщение которых содержит указанную строку) искать по ключевым словам в сообщении коммита.

Функция `-S` показывает только коммиты, в которых изменение в коде повлекло за собой добавление или удаление указанной строки.

3. Как внести изменения в уже сделанный коммит?

Внести изменения можно с помощью команды `git commit --amend`

Эта команда берёт индекс и применяет его к последнему коммиту. Если после последнего коммита не было никаких проиндексированных изменений (например, вы запустили приведённую команду сразу после предыдущего коммита), то состояние проекта будет абсолютно таким же и всё, что мы изменим, это комментарий к коммиту.

Для того, чтобы внести необходимые изменения - нам нужно проиндексировать их и выполнить команду `git commit --amend`.

```
git commit -m 'initial commit'
```

```
git add forgotten_file
```

```
git commit --amend
```

Эффект от выполнения этой команды такой, как будто мы не выполнили предыдущий коммит, а еще раз выполнили команду `git add` и выполнили коммит.

4. Как отменить индексацию файла в Git?

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` напомним вам: Прямо под текстом «Changes to be committed» говорится: используйте `git reset HEAD <file>` для исключения из индекса.

5. Как отменить изменения в файле?

С помощью команды `git checkout -- <file>`.

6. Что такое удаленный репозиторий Git?

Удалённый репозиторий это своего рода наше облако, в которое мы сохраняем те или иные изменения в нашей программе/коде/файлах.

7. Как выполнить просмотр удаленных репозиториях данного локального репозитория?

Для того, чтобы просмотреть список настроенных удалённых репозиториях, необходимо запустить команду `git remote`.

Также можно указать ключ `-v`, чтобы просмотреть адреса для чтения и записи, привязанные к репозиторию. Пример: `git remote -v`

8. Как добавить удаленный репозиторий для данного локального репозитория?

Для того, чтобы добавить удалённый репозиторий и присвоить ему имя (shortname), просто выполните команду `git remote add <shortname> <url>`.

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

Если необходимо получить изменения, которые есть у Пола, но нету у вас, вы можете выполнить команду `git fetch <Название репозитория>`. Важно отметить, что команда `git fetch` забирает данные в ваш локальный репозиторий, но не сливает их с какими-либо вашими наработками и не

модифицирует то, над чем вы работаете в данный момент. Вам необходимо вручную слить эти данные с вашими, когда вы будете готовы.

Если ветка настроена на отслеживание удалённой ветки, то вы можете использовать команду `git pull` чтобы автоматически получить изменения из удалённой ветки и слить их со своей текущей. Выполнение `git pull`, как правило, извлекает (`fetch`) данные с сервера, с которого вы изначально клонировали, и автоматически пытается слить (`merge`) их с кодом, над которым вы в данный момент работаете.

Чтобы отправить изменения на удалённый репозиторий необходимо отправить их в удалённый репозиторий. Команда для этого действия проста:

```
git push <remote-name> <branch-name>.
```

10. Как выполнить просмотр удаленного репозитория?

Для просмотра удалённого репозитория, можно использовать команду `git remote show <remote>`.

11. Каково назначение тэгов Git?

Теги - это ссылки указывающие на определённые версии кода/написанной программы. Они удобны чтобы в случае чего вернуться к нужному моменту. Также при помощи тегов можно помечать важные моменты.

12. Как осуществляется работа с тэгами Git?

Просмотреть наличие тегов можно с помощью команды: `git tag`.

А назначить (указать, добавить тег) можно с помощью команды `git tag -a v1.4(версия изначальная) -m "Название"`.

С помощью команды `git show` вы можете посмотреть данные тега вместе с коммитом: `git show v1.4`.

Отправка тегов, по умолчанию, команда `git push` не отправляет теги на удалённые сервера. После создания теги нужно отправлять явно на удалённый сервер. Процесс аналогичен отправке веток — достаточно выполнить команду `git push origin <tagname>`. Для отправки всех тегов можно

использовать команду `git push origin tags`.

Для удаления тега в локальном репозитории достаточно выполнить команду `git tag -d <tagname>`. Например, удалить созданный ранее легковесный тег можно следующим образом: `git tag -d v1.4-lw`

Для удаления тега из внешнего репозитория используется команда `git push origin --delete <tagname>`.

Если вы хотите получить версии файлов, на которые указывает тег, то вы можете сделать `git checkout` для тега пример: `git checkout -b version2 v2.0.0`.

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

`Git fetch --prune` команда получения всех изменений с репозитория GitHub.

В команде `git push --prune` удаляет удаленные ветки, у которых нет локального аналога