

Федеральное государственное автономное
образовательное учреждение высшего
образования
«Национальный исследовательский университет
ИТМО»

Факультет Информационных технологий и программирования

Работа: **Лабораторная работа 2:** Изучение системы
управления версиями Git

Выполнили: Аниськин Климентий

Проверил:

Санкт-Петербург

2022 г.

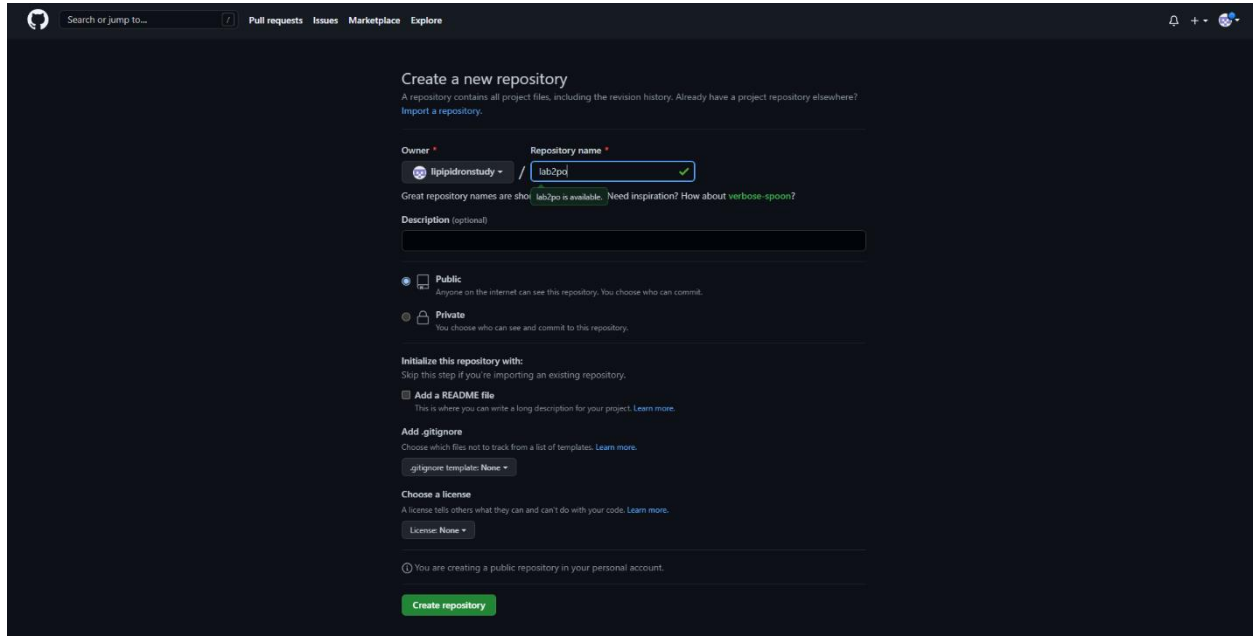
Лабораторная работа 2: Изучение системы управления версиями Git
Нужно изучить и описать систему управления версиями Git в объёме не более пяти страниц, не считая титульника. Формат сдачи следующий: я задаю вопросы по основным командам Git, вам нужно дать ответы, опираясь на ваши отчеты.
За каждый неправильный ответ отнимается один балл от числа максимального количества баллов за работу.

Разбиваемся по парам и создаем

- 1) Первый студент заводит репозиторий, второй делает в нее Pull request.
- 2) Каждому:
 - a. Создание веток по модели GitFlow - https://danielkummer.github.io/git-flow-cheatsheet/index.ru_RU.html - Обязательное наличие веток фич, релиз, девелоп, хот фикс.
 - b. В репозитории обязательно оформлен ReadMe.
 - c. Наличие тегов.
 - d. Submodules
 - e. LFS
- 3) Также всем студентам обязательно подготовить справочник по основным командам Git с примерами. Уметь кратко ответить на вопросы о предназначении основных команд Git.

Пункт 1.

1) Создание нового репозитория на гитхабе.



Create a new repository

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

Owner **lipidronstudy** / Repository name **lab2po** ✓

Great repository names are short. **lab2po** is available. Need inspiration? How about **verbose-spoon**?

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.](#)

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

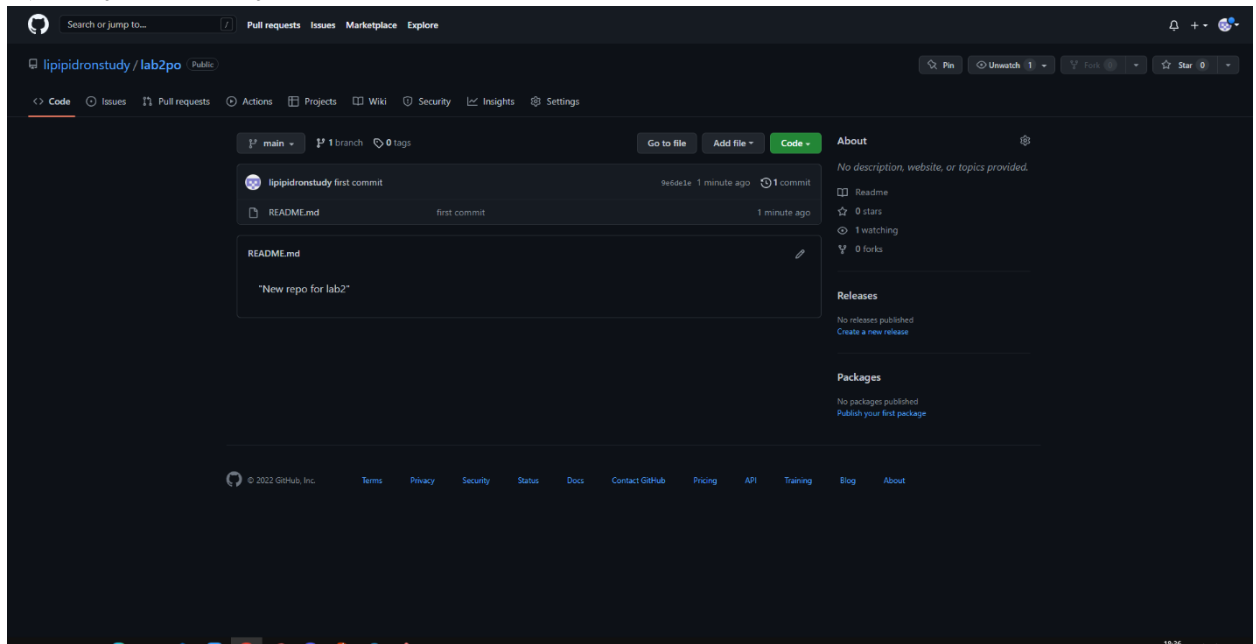
☐ You are creating a public repository in your personal account.

[Create repository](#)

2) Пушинг в репозиторий файла README.md

```
C:\Users\VadimAlg>cd C:\Users\VadimAlg\Desktop\git\lab2po
C:\Users\VadimAlg\Desktop\git\lab2po>echo "New repo for lab2" >> README.md
C:\Users\VadimAlg\Desktop\git\lab2po>git init
Initialized empty Git repository in C:/Users/VadimAlg/Desktop/git/lab2po/.git/
C:\Users\VadimAlg\Desktop\git\lab2po>git add README.md
C:\Users\VadimAlg\Desktop\git\lab2po>git commit -m "first commit"
[master (root-commit) 9e6de1e] first commit
1 file changed, 1 insertion(+)
create mode 100644 README.md
C:\Users\VadimAlg\Desktop\git\lab2po>git branch -M main
C:\Users\VadimAlg\Desktop\git\lab2po>git remote add origin https://github.com/lipidronstudy/lab2po.git
C:\Users\VadimAlg\Desktop\git\lab2po>git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 235 bytes | 117.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/lipidronstudy/lab2po.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
C:\Users\VadimAlg\Desktop\git\lab2po>
```

3)Результат пушинга на гитхабе



4)Пушинг нового .cpp файла

```
C:\Users\VadimAlg\Desktop\git\lab2po>git status
On branch main
Your branch is up to date with 'origin/main'.

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

nothing added to commit but untracked files present (use "git add" to track)

C:\Users\VadimAlg\Desktop\git\lab2po>git add main.cpp
warning: in the working copy of 'main.cpp', LF will be replaced by CRLF the next time Git touches it

C:\Users\VadimAlg\Desktop\git\lab2po>git commit -m "code can read and print int"
[main b48cc7d] code can read and print int
 1 file changed, 9 insertions(+)
 create mode 100644 main.cpp

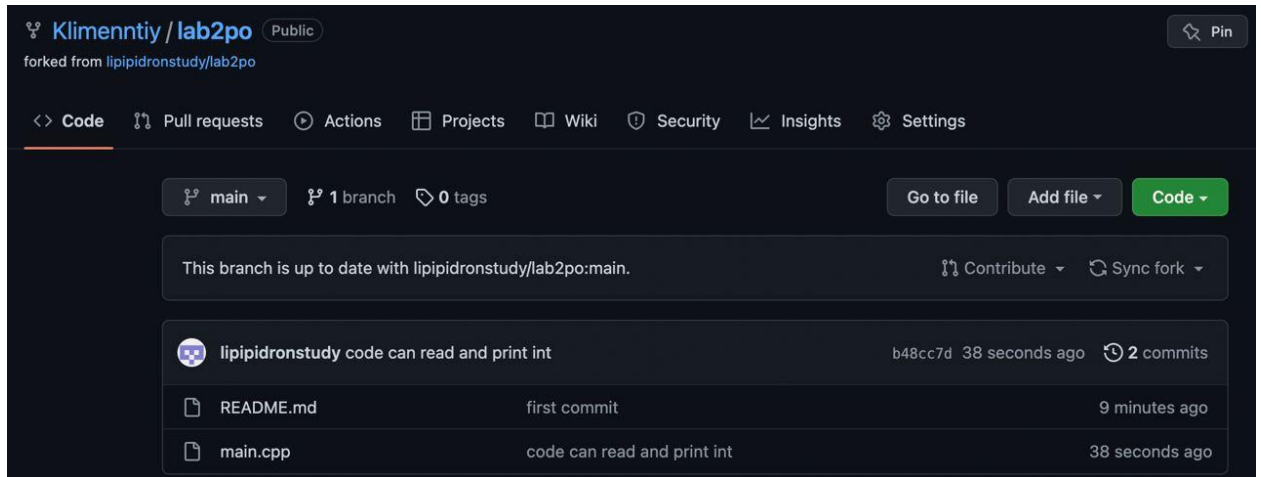
C:\Users\VadimAlg\Desktop\git\lab2po>git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

C:\Users\VadimAlg\Desktop\git\lab2po>git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 374 bytes | 187.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/lipidronstudy/lab2po.git
   9e6de1e..b48cc7d  main -> main

C:\Users\VadimAlg\Desktop\git\lab2po>
```

5)Форк репозитория другим пользователем(Климентий)



6)Клон репозитория(Климентий)

```
[klimenntiy@MacBook-Pro-Klimentij ~ % cd Documents/itmo/po
[klimenntiy@MacBook-Pro-Klimentij po % git clone git@github.com:Klimenntiy/lab2po.git
Cloning into 'lab2po'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 6 (delta 0), reused 6 (delta 0), pack-reused 0
Receiving objects: 100% (6/6), done.
klimenntiy@MacBook-Pro-Klimentij po %
```

7)Создание и переключение на новую ветку(Климентий)

```
[klimenntiy@MacBook-Pro-Klimentij lab2po % git branch meow
[klimenntiy@MacBook-Pro-Klimentij lab2po % ls -l
total 16
-rw-r--r--  1 klimenntiy  staff   21 30 сен 18:45 README.md
-rw-r--r--  1 klimenntiy  staff  120 30 сен 18:45 main.cpp
[klimenntiy@MacBook-Pro-Klimentij lab2po % git checkout meow
Switched to branch 'meow'
klimenntiy@MacBook-Pro-Klimentij lab2po %
```

8) Внес изменения в main.cpp и закоммитил это (Климентий)

```
[klimenntiy@MacBook-Pro-Klimentij lab2po % git add 'main.cpp'
[klimenntiy@MacBook-Pro-Klimentij lab2po % git commit -m 'add float munber'
[meow c63a35f] add float munber
 1 file changed, 3 insertions(+)
klimenntiy@MacBook-Pro-Klimentij lab2po %
```

9)Запустил ветку в форк(Климентий)

```
klimenntiy@MacBook-Pro-Klimentij lab2po % git push --set-upstream origin meow
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), 381 bytes | 381.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'meow' on GitHub by visiting:
remote:   https://github.com/Klimenntiy/lab2po/pull/new/meow
remote:
To github.com:Klimenntiy/lab2po.git
 * [new branch]      meow -> meow
Branch 'meow' set up to track remote branch 'meow' from 'origin'.
klimenntiy@MacBook-Pro-Klimentij lab2po %
```

10)Произвел Pull Request(Климентий)

The screenshot shows a GitHub Pull Request titled "add float number #1". The PR is created by "Klimenntiy" and targets the "main" branch of the "lipidronstudy" repository. The PR description is "No description provided." and the commit hash is "c63a35f". The interface includes tabs for "Conversation", "Commits", "Checks", and "Files changed". A green checkmark icon indicates that the branch has no conflicts with the base branch. A message states: "This branch has no conflicts with the base branch. Only those with write access to this repository can merge pull requests."

Пункт 2.

1) Клон репозитория


```

[klimenntiy@MacBook-Pro-Klimentij lab2po % cd
[klimenntiy@MacBook-Pro-Klimentij ~ % cd Documents/itmo/po
[klimenntiy@MacBook-Pro-Klimentij po % git clone git@github.com:Klimenntiy/lab2po.git
Cloning into 'lab2po'...
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 9 (delta 1), reused 8 (delta 0), pack-reused 0
Receiving objects: 100% (9/9), done.
Resolving deltas: 100% (1/1), done.
[klimenntiy@MacBook-Pro-Klimentij po % █

```

2) Создание веток фич, девелоп, хотфикс и релиз

```

[klimenntiy@MacBook-Pro-Klimentij lab2po % git checkout meow
Branch 'meow' set up to track remote branch 'meow' from 'origin'.
Switched to a new branch 'meow'
[klimenntiy@MacBook-Pro-Klimentij lab2po % git branch meow_fitc
[klimenntiy@MacBook-Pro-Klimentij lab2po % git branch meow_develop
[klimenntiy@MacBook-Pro-Klimentij lab2po % git branch meow_realese
[klimenntiy@MacBook-Pro-Klimentij lab2po % git branch meow_hot_fix
[klimenntiy@MacBook-Pro-Klimentij lab2po % git branch -a
main
* meow
meow_develop
meow_fitc
meow_hot_fix
meow_realese

```

3) Добавление новых функций в ветку фич, добавление тега.

```

[klimenntiy@MacBook-Pro-Klimentij lab2po % git checkout meow_fitc
Switched to branch 'meow_fitc'
[klimenntiy@MacBook-Pro-Klimentij lab2po % git add 'main.cpp'
[klimenntiy@MacBook-Pro-Klimentij lab2po % git coommit -m 'add char number'
git: 'coommit' is not a git command. See 'git --help'.

The most similar command is
commit
[klimenntiy@MacBook-Pro-Klimentij lab2po % git tag v1.1-char
[klimenntiy@MacBook-Pro-Klimentij lab2po % git tag
v1.1-char
[klimenntiy@MacBook-Pro-Klimentij lab2po % █

```

3) Фикс багов и тег

```

[klimenntiy@MacBook-Pro-Klimentij lab2po % git checkout meow_hot_fix
M      main.cpp
Switched to branch 'meow_hot_fix'
[klimenntiy@MacBook-Pro-Klimentij lab2po % git add 'main.cpp'
[klimenntiy@MacBook-Pro-Klimentij lab2po % git commit -m 'fix add ;'
[meow_hot_fix f505475] fix add ;
1 file changed, 5 insertions(+), 2 deletions(-)
[klimenntiy@MacBook-Pro-Klimentij lab2po % git tag v1.2-fix
[klimenntiy@MacBook-Pro-Klimentij lab2po % git tag
v1.1-char
v1.2-fix

```

4) Добавление основного файла в ветки релиз и девелоп

```
[klimenntiy@MacBook-Pro-Klimentij lab2po % git add 'main.cpp'
[klimenntiy@MacBook-Pro-Klimentij lab2po % git commit -m 'add start file'
[meow_develop ddb57d8] add start file
1 file changed, 2 insertions(+)
[klimenntiy@MacBook-Pro-Klimentij lab2po % git checkout meow_realese
Switched to branch 'meow_realese'
[klimenntiy@MacBook-Pro-Klimentij lab2po % git add 'main.cpp'
[klimenntiy@MacBook-Pro-Klimentij lab2po % git commit -m 'add start file'
[meow_realese b93a067] add start file
1 file changed, 1 insertion(+)
klimenntiy@MacBook-Pro-Klimentij lab2po %
```

5) Добавление submodule

```
[klimenntiy@MacBook-Pro-Klimentij lab2po % git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
[klimenntiy@MacBook-Pro-Klimentij lab2po % git submodule add git@github.com:lipipidronstudyy
-c-func.git
Cloning into '/Users/klimenntiy/Documents/itmo/po/lab2po/c-func'...
remote: Enumerating objects: 12, done.
remote: Counting objects: 100% (12/12), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 12 (delta 1), reused 9 (delta 1), pack-reused 0
Receiving objects: 100% (12/12), done.
Resolving deltas: 100% (1/1), done.
klimenntiy@MacBook-Pro-Klimentij lab2po %
```

6) Установка lfs

```
klimenntiy@MacBook-Pro-Klimentij lab2po % eval "$(curl -s https://raw.githubusercontent.com/git-lfs/git-lfs/master/bin/install.sh)"
klimenntiy@MacBook-Pro-Klimentij lab2po % brew install git-lfs
--> Downloading https://ghcr.io/v2/homebrew/core/git-lfs/manifests/3.2.0
##### 100.0%
--> Downloading https://ghcr.io/v2/homebrew/core/git-lfs/blobs/sha256:a81d5b29d1c9283f592e8af7b25ccb95431489ffa8dc595dbb1c39f1c08cfb46
--> Downloading from https://pkg-containers.githubusercontent.com/ghcr1/blobs/sha256:a81d5b29d1c9283f592e8af7b25ccb95431489ffa8dc595dbb1c39f1c08cfb46?se=2022-09-30T18%3A45%3A00Z&sig=vFjeSy7ou881BEUxr3H0M4Y9HJ3Y
##### 100.0%
--> Pouring git-lfs--3.2.0.arm64_monterey.bottle.tar.gz
--> Caveats
Update your git config to finish installation:
# Update global git config
$ git lfs install
# Update system git config
$ git lfs install --system
--> Summary
📦 /opt/homebrew/Cellar/git-lfs/3.2.0: 3 files, 11.0MB
--> Running 'brew cleanup git-lfs'...
Disable this behaviour by setting HOMEBREW_NO_INSTALL_CLEANUP.
Hide these hints with HOMEBREW_NO_ENV_HINTS (see 'man brew').
klimenntiy@MacBook-Pro-Klimentij lab2po %
```

7) Добавление большого файла


```
klimenntiy@MacBook-Pro-Klimentij lab2po % git lfs track dis1.exe
Tracking "dis1.exe"
klimenntiy@MacBook-Pro-Klimentij lab2po % git add .gitattributes
klimenntiy@MacBook-Pro-Klimentij lab2po % git add dis1.exe
fatal: pathspec 'dis1.exe' did not match any files
klimenntiy@MacBook-Pro-Klimentij lab2po % git add dis1.exe
klimenntiy@MacBook-Pro-Klimentij lab2po % git commit -m "add dis1.exe"
[main bb32b27] add dis1.exe
 2 files changed, 2 insertions(+)
 create mode 100644 .gitattributes
 create mode 100644 dis1.exe
klimenntiy@MacBook-Pro-Klimentij lab2po %
```

Справочник основных команд GIT.

- **git init**

Эта команда используется для создания GIT репозитория.

Пример использования:

git init

- **git config**

Одна из самых популярных команд, используется для указания пользовательских настроек, эл.почта, имя пользователя и т.д.

Например, данная команда используется для установки имени пользователя:

git config --global user.name имя пользователя

- **git clone**

Используется для клонирования репозитория

Пример:

git clone <URL репозитория> (если находится на удаленном сервере)

git clone /путь/до/репозитория (если репозиторий локальный)

- **git add**

Используется для добавления файла в индекс

Например, следующая команда добавит файл под названием file.txt присутствующий в локальном каталоге в индекс:

git add file.txt

- **git status**

Отображает список измененный файлов + еще не добавленные или ожидающие коммита

Пример:

```
git status
```

- **git push**

Одна из главных команд. Позволяет поместить изменения в главную ветку удаленного хранилища связанного с рабочим каталогом. Например:

```
git push origin master
```

- **git branch**

Может быть использована для отображения, создания или удаления веток. Для отображения всех существующих веток в репозитории введите:

```
git branch
```

Для удаления ветки:

```
git branch -d
```

- **git checkout**

Может быть использована для создания веток или переключения между ними.

К примеру, следующий код создаст новую ветку и переключится на нее:

```
git checkout -b
```

Чтобы просто переключиться между ветками используйте:

```
git checkout
```

- **git commit**

Используется для коммита изменений в файлах проекта.

Применение:

git commit -m "Сообщение идущее вместе с коммитом"

- **git pull**

Используется для объединения изменений, присутствующих в удаленном репозитории, в локальный рабочий каталог.

Применение:

git pull

- **git merge**

Используется для объединения ветки в активную ветвь.

Применение:

git merge

- **git log**

Отобразит список всех коммитов в ветке вместе с соответствующими сведениями.

Пример использования:

git log

commit 15f4b6c44b3c8344caasdac9e4be13246e21saw

Author: Iven Ivanov <iveni@gmail.com>

Date: Tue Sep 12 11:00:00 2022 -0600

- **git tag**

Используется для маркировки определенных коммитов с помощью простых меток. Примером может быть эта команда:

git tag 1.1.0 <вставьте ключевое слово>

- **git remote**

Позволяет пользователю подключиться к удаленному репозиторию.

Данная команда отобразит список удаленных репозиторий, настроенных в данный момент:

```
git remote -v
```

Позволит пользователю подключить локальный репозиторий к удаленному серверу:

```
git remote add origin <адрес удаленного сервера>
```

- **git diff**

Используется для выявления различий между ветками.

Для выявления различий с базовыми файлами, используйте

```
git diff --base
```

Следующая команда используется для просмотра различий между ветками, которые должны быть объединены, до их объединения:

```
git diff <ветвь – источник> <ветвь – цель>
```

Для простого отображения существующих различий, используйте:

```
git diff
```

- **git reset**

Используется для сброса индекса и рабочего каталога до последнего состояния коммита. Применение:

```
git reset --hard HEAD
```

- **git rm**

Используется для удаления файлов из индекса и рабочего каталога.

Применение:

git rm имяфайла.txt

- **git stash**

Одна из самых малоизвестных команд git. Она помогает в сохранении изменений на временной основе, эти изменения не попадут в коммит сразу.

Применение:

git stash

- **git show**

Используется для просмотра информации о любом git объекте. Для примера:

git show

- **git fetch**

Позволяет пользователю доставить все объекты из удаленного репозитория, которые не присутствуют в локальном рабочем каталоге.

Пример:

git fetch origin

- **git gc**

Используется для оптимизации репозитория, удаляет и оптимизирует ненужные файлы

Пример:

```
git gc
```