



МИНОБРНАУКИ РОССИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технологический университет
«СТАНКИН»
(ФГБОУ ВО «МГТУ «СТАНКИН»)

Институт автоматизации и робототехники
Кафедра компьютерных систем управления

Дисциплина «Основы системного программного обеспечения»

Отчет по лабораторной работе №_1_

Тема: «Работа с системами контроля версий на примере Git Hub»

Выполнил

студент гр. АДБ-20-08 : _____ Денисенко М.М.

(дата)

(подпись)

Проверил

к.т.н., доцент _____ Ковалев И.А.

(дата)

(подпись)

Москва 2023 г.

Цель работы: Познакомиться с принципом работы систем контроля версий на примере Git Hub.

Краткая теория

Система контроля версий (СКВ) — это система, регистрирующая изменения в одном или нескольких файлах с тем, чтобы в дальнейшем была возможность вернуться к определённым старым версиям этих файлов. Для примеров будем использовать исходные коды программ, но на самом деле под версионный контроль можно поместить файлы практически любого типа.

Индивидуальные задания

Создание локального репозитория

В проводнике windows был создан текстовый файл, и, соответственно, записана текстовая информация в виде фамилии

```
C:\Users\maks\Documents\Denisenko>git status
On branch master

No commits yet

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

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

В нашем случае файл новый и система еще не знает, нужно ли следить за изменениями в файле или его можно просто игнорировать. Для того, чтобы начать отслеживать новый файл, нужно его специальным образом объявить.

Фиксация изменений в области заготовленных файлов

Если хотим добавить только один файл, можно написать:

```
git add test.txt
```

Если измененных файлов много, то можно сделать так:

```
git add .
```

Выполнив нужную команду снова посмотрим статус репозитория

```
C:\Users\maks\Documents\Denisenko>git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   text.txt

C:\Users\maks\Documents\Denisenko>
```

Файл готов к коммиту. Сообщение о состоянии также говорит нам о том, какие изменения относительно файла были проведены в области подготовки — в данном случае это новый файл.

Чтобы зафиксировать изменения, нам нужно хотя бы одно изменение в области подготовки (мы как раз создали новый файл):

`git commit -m "first commit"`

```
C:\Users\maksi\Documents\Denisenko>git commit -m "first commit"
[master (root-commit) 168129c] first commit
1 file changed, 1 insertion(+)
create mode 100644 text.txt
```

Отправка коминат на сервер

Свяжем наш локальный репозиторий с облачным репозиторием, который мы создали на github.com

Можно вызвать команду, для просмотра, к какому проекту мы подключены `git remote -v`

```
C:\Users\maksi\Documents\Denisenko>git remote add origin https://github.com/MaksDenisenko/Denisenko

C:\Users\maksi\Documents\Denisenko>git remote -v
origin https://github.com/MaksDenisenko/Denisenko (fetch)
origin https://github.com/MaksDenisenko/Denisenko (push)

C:\Users\maksi\Documents\Denisenko>git pull origin master --allow-unrelated-histories
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 597 bytes | 29.00 KiB/s, done.
From https://github.com/MaksDenisenko/Denisenko
* branch      master      -> FETCH_HEAD
* [new branch] master      -> origin/master
Merge made by the 'ort' strategy.
 README.md | 1 +
1 file changed, 1 insertion(+)
create mode 100644 README.md

C:\Users\maksi\Documents\Denisenko>git remote -v
origin https://github.com/MaksDenisenko/Denisenko (fetch)
origin https://github.com/MaksDenisenko/Denisenko (push)

C:\Users\maksi\Documents\Denisenko>_
```

Перешлем локальный коммит на сервер

Теперь отправим коммит на сервер, команда, предназначенная для этого — `push`. Она принимает два параметра: имя удаленного репозитория (мы назвали наш `origin`) и ветку, в которую необходимо внести изменения (`master` — это ветка по умолчанию для всех репозиториях).

`git push origin master`

Посмотрим наши изменения с использованием команды

`git log`

Видны коммиты по изменениям в файле `Readme.md` и в файле с именем (`first commit`)

```

C:\Users\maksim\Documents\Denisenko>git push origin master
info: please complete authentication in your browser...
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (5/5), 527 bytes | 527.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/MaksDenisenko/Denisenko
    62ea9cb..38354ef  master -> master

C:\Users\maksim\Documents\Denisenko>git log
commit 38354ef34112a6951952b0f65a50e11fbc078670 (HEAD -> master, origin/master)
Merge: 168129c 62ea9cb
Author: MaksDenisenko <maksim.denisenko.20@mail.ru>
Date:   Thu Apr 20 16:44:18 2023 +0300

    Merge branch 'master' of https://github.com/MaksDenisenko/Denisenko

commit 168129c5626d2712a1454c92ad98edac26a68b1b
Author: MaksDenisenko <maksim.denisenko.20@mail.ru>
Date:   Thu Apr 20 16:42:52 2023 +0300

    first commit

commit 62ea9cbf531ed347cc536f07400b6f68d836b91a
Author: MaksDenisenko <131355400+MaksDenisenko@users.noreply.github.com>
Date:   Thu Apr 20 16:32:04 2023 +0300

    Initial commit

C:\Users\maksim\Documents\Denisenko>_

```

Изменения на github.com

master
1 branch
0 tags
Go to file
Add file
Code

MaksDenisenko Merge branch 'master' of <https://github.com/MaksDenisenko/Denisenko>
38354ef 3 minutes ago
3 commits

README.md	Initial commit	15 minutes ago
text.txt	first commit	4 minutes ago

README.md

Denisenko

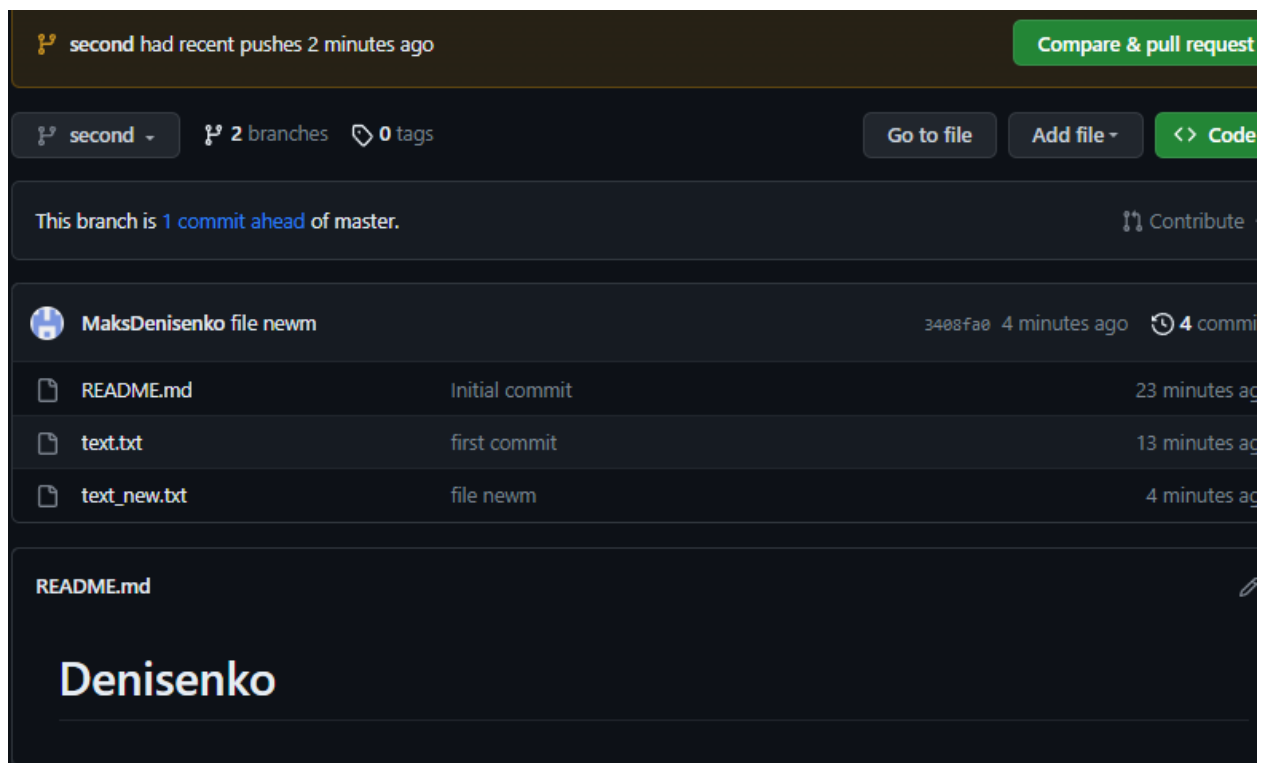
Создание новой ветки

Во время разработки новой функциональности считается хорошей практикой работать с копией оригинального проекта, которую называют веткой. Ветви имеют свою собственную историю и изолированные друг от друга изменения до тех пор, пока вы не решаете слить изменения вместе. Это происходит по набору причин:

- Уже рабочая, стабильная версия кода сохраняется.
- Различные новые функции могут разрабатываться параллельно разными программистами.
- Разработчики могут работать с собственными ветками без риска, что кодовая база поменяется из-за чужих изменений.
- В случае сомнений, различные реализации одной и той же идеи могут быть разработаны в разных ветках и затем сравниваться.

```
C:\Users\maks\Documents\Denisenko>git branch second  
  
C:\Users\maks\Documents\Denisenko>git branch  
* master  
  second  
  
C:\Users\maks\Documents\Denisenko>git checkout second  
Switched to branch 'second'
```

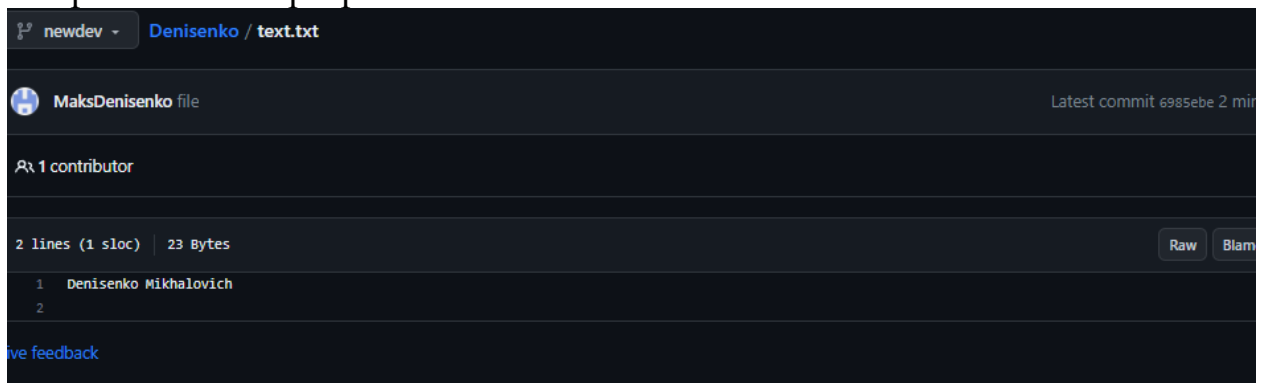
В ветке second появился новый файл, который создали в проводнике windows



Слияние веток

Если через проводник windows в папку с проектом, то файла test_new.txt не будет виден, потому что мы переключились обратно на ветку master, в которой такого файла не существует. Чтобы он появился, нужно воспользоваться merge для объединения веток (применения изменений из ветки second к основной версии проекта).

Попробуем создать ветку newdev, а в файл с фамилией добавим отчество. Отправляем на сервер.



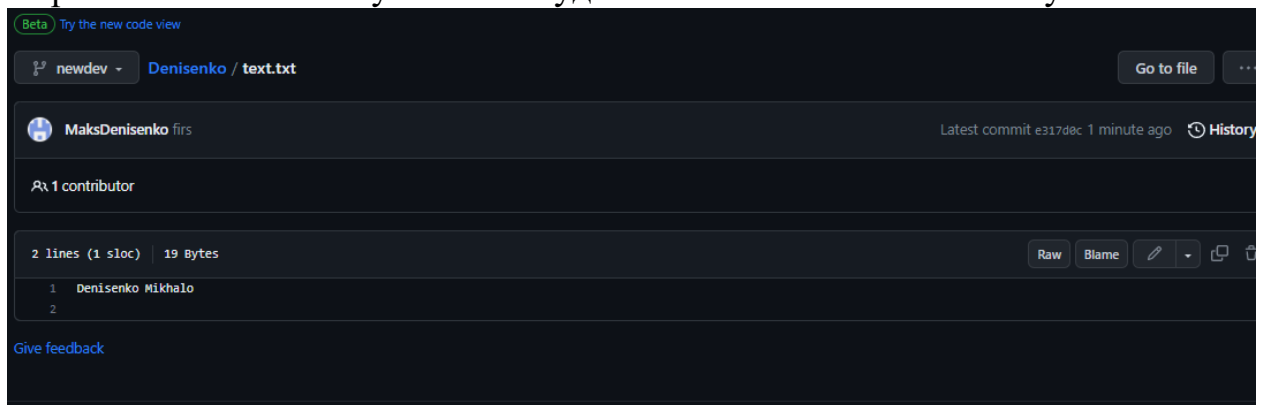
Переключимся обратно на ветку master и посмотрим файл: в нем только фамилия, отчество в другой ветке

Объединим ветки

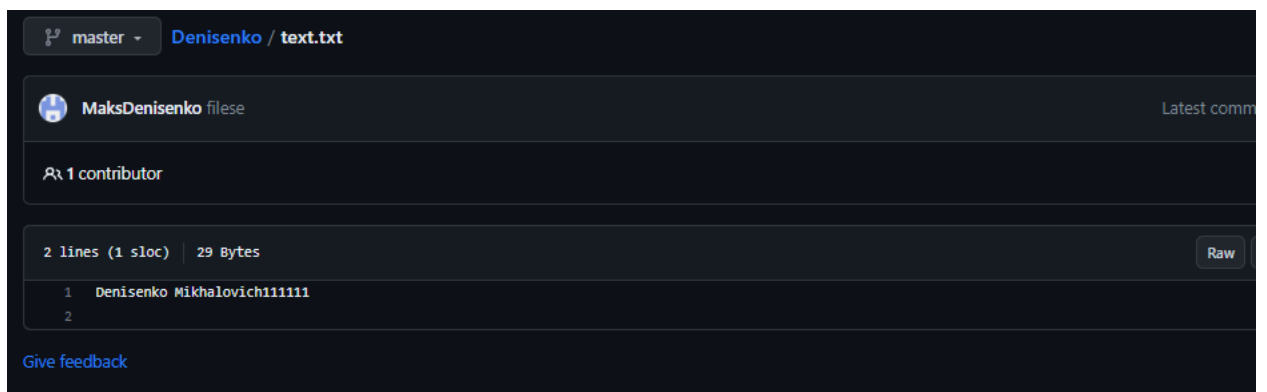
`git merge newdev`

Откроем файл снова, там есть и фамилия, и отчество

Переключимся на ветку newdev и удалим в отчестве несколько букв



Переключимся на ветку master, добавьте к отчеству несколько букв



Объединим ветки

```
git merge newdev
```

Теперь ничего не получится, т.к. есть изменения в обеих ветках

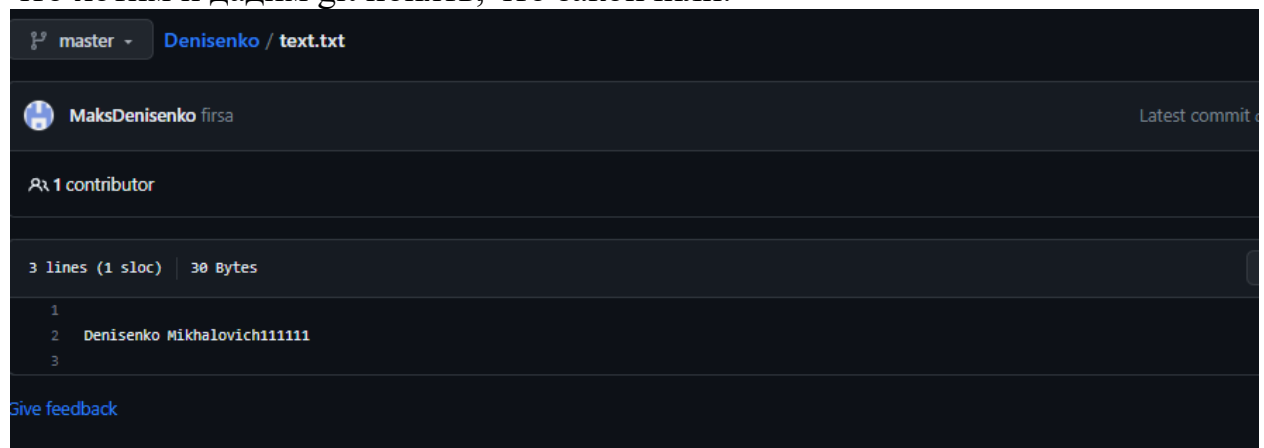
Просмотр изменений и разрешение конфликтов

Наберите команду, для просмотра изменений

```
git diff
```

```
C:\Users\maks\Documents\Denisenko>git diff
diff --cc text.txt
index be49fc7,2158d53..0000000
--- a/text.txt
+++ b/text.txt
@@@ -1,2 -1,2 +1,6 @@@
++<<<<<< HEAD
+Denisenko Mikhailovich111111
++=====
+ Denisenko Mikhalo
++>>>>>> newdev
```

Над разделителем ===== мы видим последний (HEAD) коммит, а под ним — конфликтующий. Таким образом, мы можем увидеть, чем они отличаются и решать, какая версия лучше. Или вовсе написать новую. В этой ситуации мы так и поступим, удалив разделители (HEAD, ==, <<<,>>>), оставим то, что хотим и дадим git понять, что закончили.

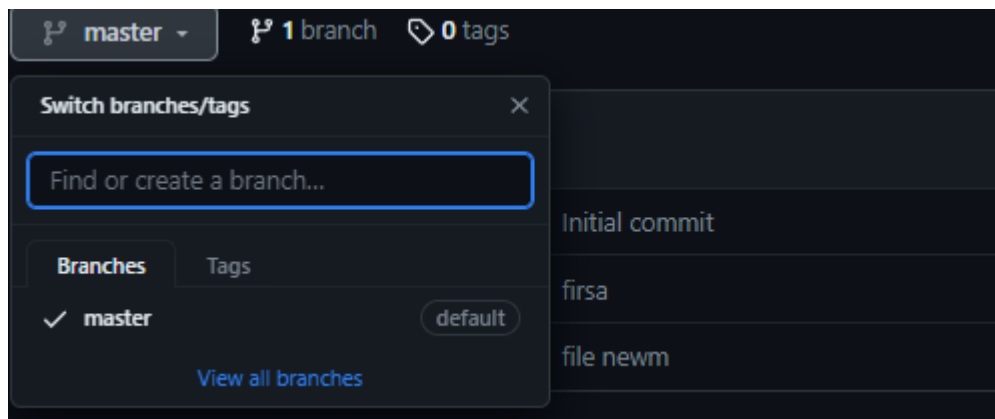


Удаление веток на сервере

```
C:\Users\maksi\Documents\Denisenko>git push origin --delete newdev
To https://github.com/MaksDenisenko/Denisenko
- [deleted]          newdev

C:\Users\maksi\Documents\Denisenko>git push origin --delete second
To https://github.com/MaksDenisenko/Denisenko
- [deleted]          second
```

Все ветки удалены



Возврат к предыдущему состоянию

Гит позволяет вернуть выбранный файл к состоянию на момент определенного коммита. Это делается уже знакомой нам командой `checkout`, которую мы ранее использовали для переключения между ветками. Но она также может быть использована для переключения между коммитами.

Чтобы посмотреть все комиты, можно использовать команду

`git log`

Выберете любой коммит, на который хотите откатится, достаточно указать его

первые несколько символов (перед выполнением посмотрите свои файлы, чтобы запомнить, что там находится)

`git checkout [hash commit]`


```

C:\Users\maksi\Documents\Denisenko>git log
commit d530510daafcb23b461285f9fd29e54a036d93ad (HEAD -> master, origin/master)
Merge: 03fe239 e317d0c
Author: MaksDenisenko <maksim.denisenko.20@mail.ru>
Date: Thu Apr 20 17:36:11 2023 +0300

    firs

commit 03fe23934ecbf118f966fb2e2ea602ed3d1df649
Author: MaksDenisenko <maksim.denisenko.20@mail.ru>
Date: Thu Apr 20 17:23:23 2023 +0300

    filese

commit e317d0c7067feafbea8648a31a88e2ae15045ea1
Author: MaksDenisenko <maksim.denisenko.20@mail.ru>
Date: Thu Apr 20 17:15:55 2023 +0300

    firs

commit 6985ebe878cc2c0f48bb50db6733e4c12988bea2
Author: MaksDenisenko <maksim.denisenko.20@mail.ru>
Date: Thu Apr 20 17:10:09 2023 +0300

    file

commit 3408fa061e26d213da730d4d3d676459a6791f83
Author: MaksDenisenko <maksim.denisenko.20@mail.ru>
Date: Thu Apr 20 16:51:22 2023 +0300

    file newm

commit 38354ef34112a6951952b0f65a50e11fbc078670
Merge: 168129c 62ea9cb
Author: MaksDenisenko <maksim.denisenko.20@mail.ru>
Date: Thu Apr 20 16:44:18 2023 +0300

    Merge branch 'master' of https://github.com/MaksDenisenko/Denisenko

commit 168129c5626d2712a1454c92ad98edac26a68b1b
Author: MaksDenisenko <maksim.denisenko.20@mail.ru>
Date: Thu Apr 20 16:42:52 2023 +0300

    first commit

commit 62ea9cbf531ed347cc536f07400b6f68d836b91a
Author: MaksDenisenko <131355400+MaksDenisenko@users.noreply.github.com>
Date: Thu Apr 20 16:32:04 2023 +0300

    Initial commit

C:\Users\maksi\Documents\Denisenko>

```

Если посмотреть файлы теперь, то увидите, что они поменялись — мы вернулись назад

```

C:\Users\maksi\Documents\Denisenko>git branch
* (HEAD detached at 6985ebe)
master

```

Исправление коммита

Файлы которые закрепили, но не коммитили, можно убрать из области закрепления командой

`git reset HEAD`

```
C:\Users\maks\Documents\Denisenko>git add .

C:\Users\maks\Documents\Denisenko>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   text.txt

C:\Users\maks\Documents\Denisenko>git reset HEAD
Unstaged changes after reset:
M       text.txt

C:\Users\maks\Documents\Denisenko>git status
On branch master
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:   text.txt

no changes added to commit (use "git add" and/or "git commit -a")

C:\Users\maks\Documents\Denisenko>_
```

Можно изменить комментарий, вызвав команду:

`git commit --amend`

```
C:\Users\maks\Documents\Denisenko>git commit --amend
[master 5a3ca3d] firr
Date: Thu Apr 20 18:16:22 2023 +0300
1 file changed, 1 insertion(+), 1 deletion(-)

C:\Users\maks\Documents\Denisenko>_
```

Отправка только нужных файлов на сервер

```
C:\Users\maks\Documents\Denisenko>git status
warning: unable to access '.gitignore': Permission denied
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   .gitignore/team.txt
        new file:   .gitignore/track.txt

C:\Users\maks\Documents\Denisenko>git commit -m "newww"
[master 84de5a8] newww
2 files changed, 2 insertions(+)
create mode 100644 .gitignore/team.txt
create mode 100644 .gitignore/track.txt
```

Совместная работа с git

```
C:\Users\maksi\Documents\Denisenko\qqq>git clone https://github.com/Suden25d2/Lab1
Cloning into 'Lab1'...
remote: Enumerating objects: 43, done.
remote: Counting objects: 100% (43/43), done.
remote: Compressing objects: 100% (19/19), done.
remote: Total 43 (delta 11), reused 38 (delta 9), pack-reused 0
Receiving objects: 100% (43/43), done.
Resolving deltas: 100% (11/11), done.
```

```
C:\Users\maksi\Documents\Denisenko\qqq>git push origin master
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 300 bytes | 150.00 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/MaksDenisenko/Denisenko
 84de5a8..f2a27d5  master -> master

C:\Users\maksi\Documents\Denisenko\qqq>git log
commit f2a27d5303f13d2797e15ff4fd6602b98d54bd2a (HEAD -> master, origin/master)
Author: MaksDenisenko <maksim.denisenko.20@mail.ru>
Date: Thu May 25 12:32:07 2023 +0300

    fir git

commit 84de5a8aa82bc22020653991321cab928de0964b
Author: MaksDenisenko <maksim.denisenko.20@mail.ru>
Date: Thu Apr 20 19:39:10 2023 +0300

    newww

commit 5a3ca3d86d8822c87a57af7c98d17d11524aeace
Author: MaksDenisenko <maksim.denisenko.20@mail.ru>
Date: Thu Apr 20 18:16:22 2023 +0300

    firr

commit 35a722fadd78c8de5e0ac5d9556ef3a9cb563c3f
Author: MaksDenisenko <maksim.denisenko.20@mail.ru>
Date: Thu Apr 20 18:13:10 2023 +0300

    fir

commit d530510daafcb23b461285f9fd29e54a036d93ad
Merge: 03fe239 e317d0c
Author: MaksDenisenko <maksim.denisenko.20@mail.ru>
Date: Thu Apr 20 17:36:11 2023 +0300

...skipping...
commit f2a27d5303f13d2797e15ff4fd6602b98d54bd2a (HEAD -> master, origin/master)
Author: MaksDenisenko <maksim.denisenko.20@mail.ru>
Date: Thu May 25 12:32:07 2023 +0300

    fir git
```

Вывод: В данной лабораторной работе я изучил с принцип работы систем контроля версий на примере Git Hub.