МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»
РТУ МИРЭА

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных технологий

Отчет по практической работе №1

по дисциплине «Технологии разработки программных приложений»

Тема практической работы: «Системы контроля версий»

Выполнил:
Студент группы ИКБО-41-23 Гольд Д.В.

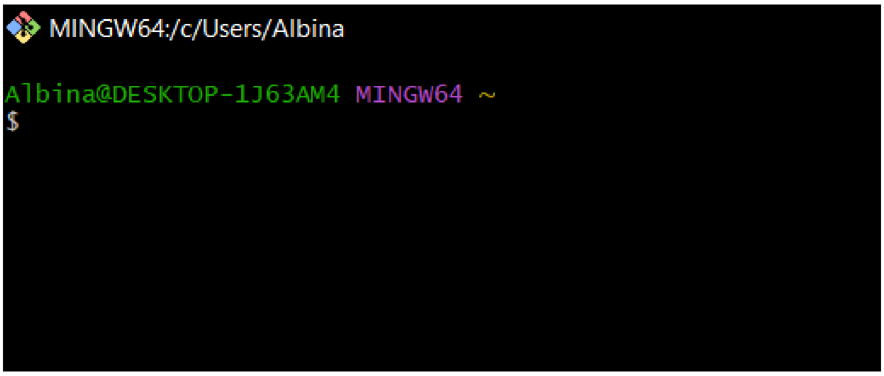
Проверил:
Доцент кафедры МОСИТ,
кандидат технических наук, доцент
Жматов Д.В.

Москва 2025

Оглавление

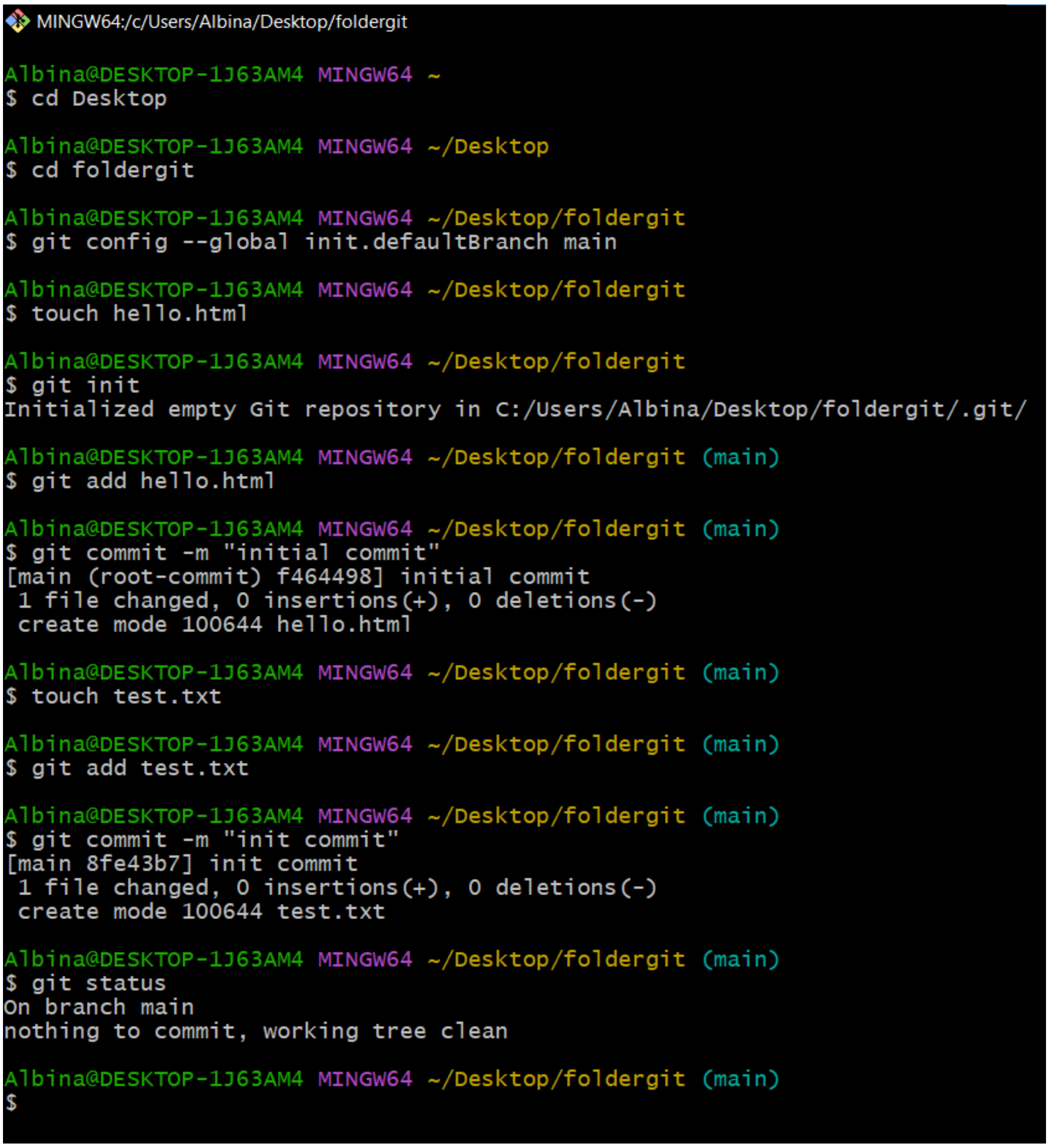
ЧАСТЬ 1. ОСНОВНЫЕ КОМАНДЫ GIT

1. Установить и настроить клиент git на своей рабочей станции.



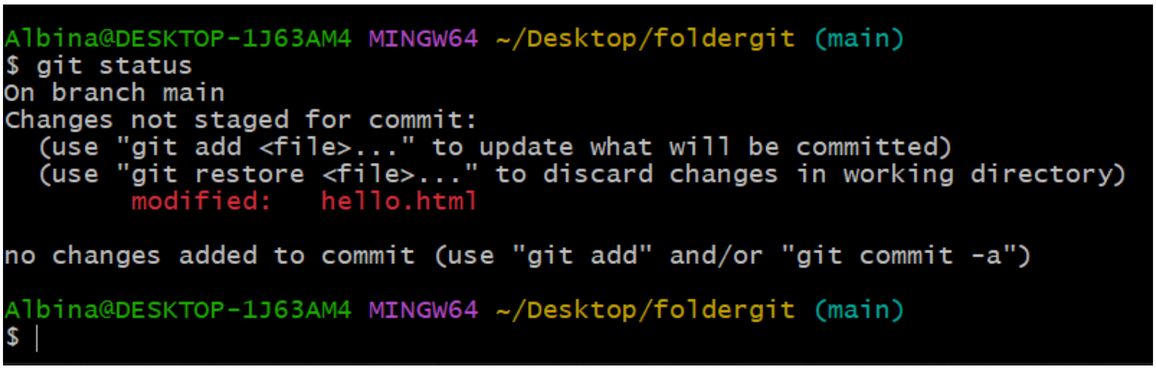
```
MINGW64:/c/Users/Albina

Albina@DESKTOP-1J63AM4 MINGW64 ~
$
```

2. Создать локальный репозиторий и добавить в него несколько файлов.



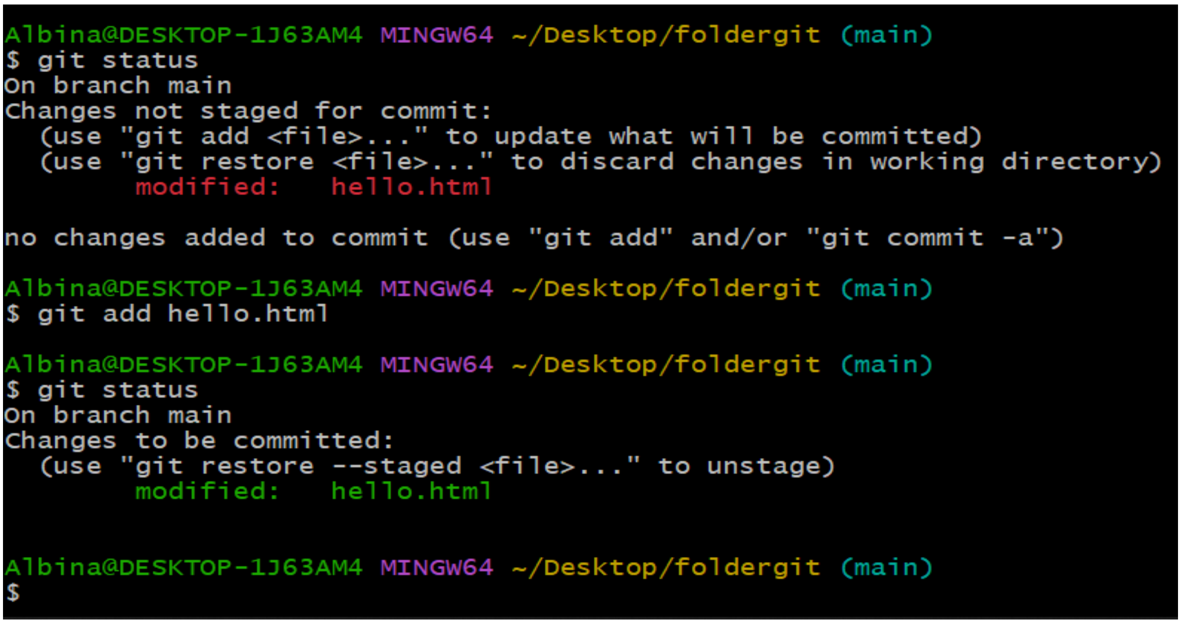
```
MINGW64:/c/Users/Albina/Desktop/foldergit

Albina@DESKTOP-1J63AM4 MINGW64 ~
$ cd Desktop

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop
$ cd foldergit

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit
$ git config --global init.defaultBranch main

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit
$ touch hello.html

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit
$ git init
Initialized empty Git repository in C:/Users/Albina/Desktop/foldergit/.git/

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit (main)
$ git add hello.html

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit (main)
$ git commit -m "initial commit"
[main (root-commit) f464498] initial commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 hello.html

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit (main)
$ touch test.txt

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit (main)
$ git add test.txt

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit (main)
$ git commit -m "init commit"
[main 8fe43b7] init commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 test.txt

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit (main)
$ git status
On branch main
nothing to commit, working tree clean

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit (main)
$
```

3. Внести изменения в один из файлов



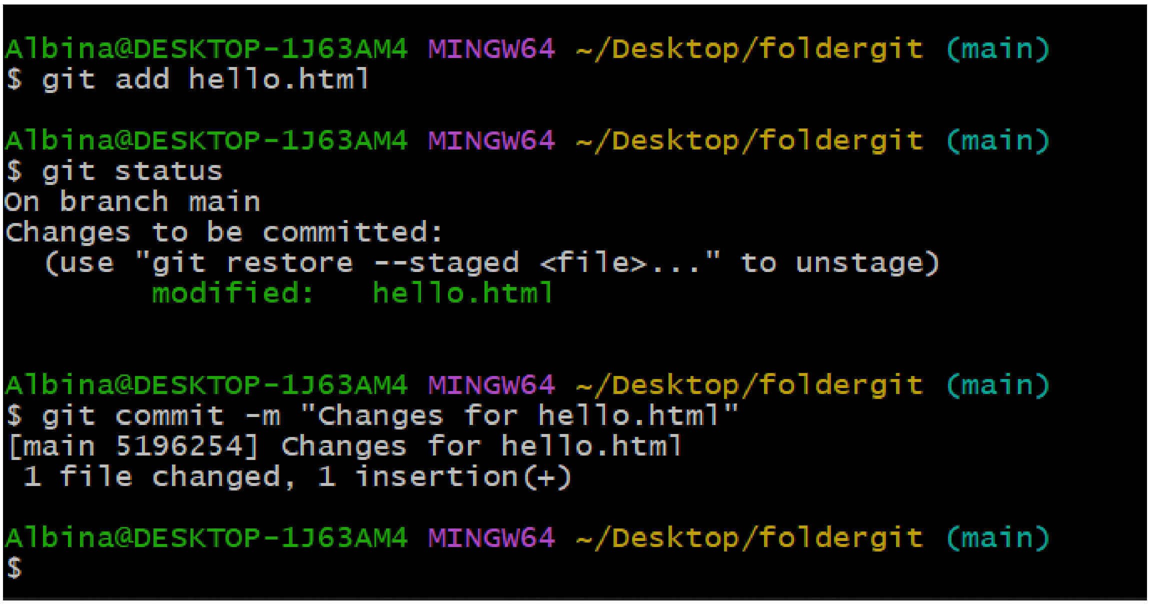
```
Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit (main)
$ git status
On branch main
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   hello.html

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

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit (main)
$ |
```

4. Проиндексировать изменение и проверьте состояние.



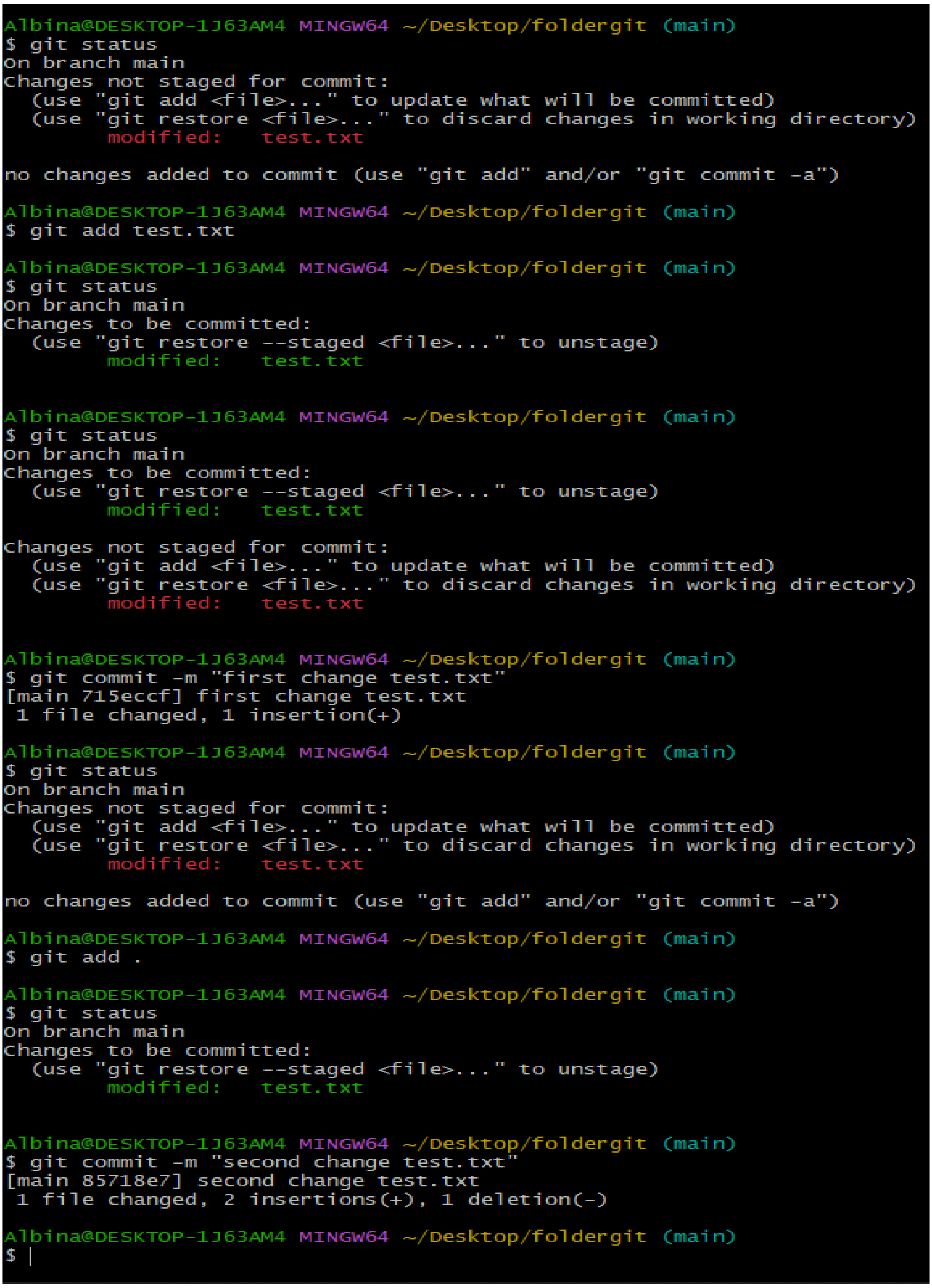
```
Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit (main)
$ git status
On branch main
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   hello.html

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

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit (main)
$ git add hello.html

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit (main)
$ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   hello.html


Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit (main)
$
```

5. Сделать коммит того, что было проиндексировано в репозиторий. Добавить к коммиту комментарий.



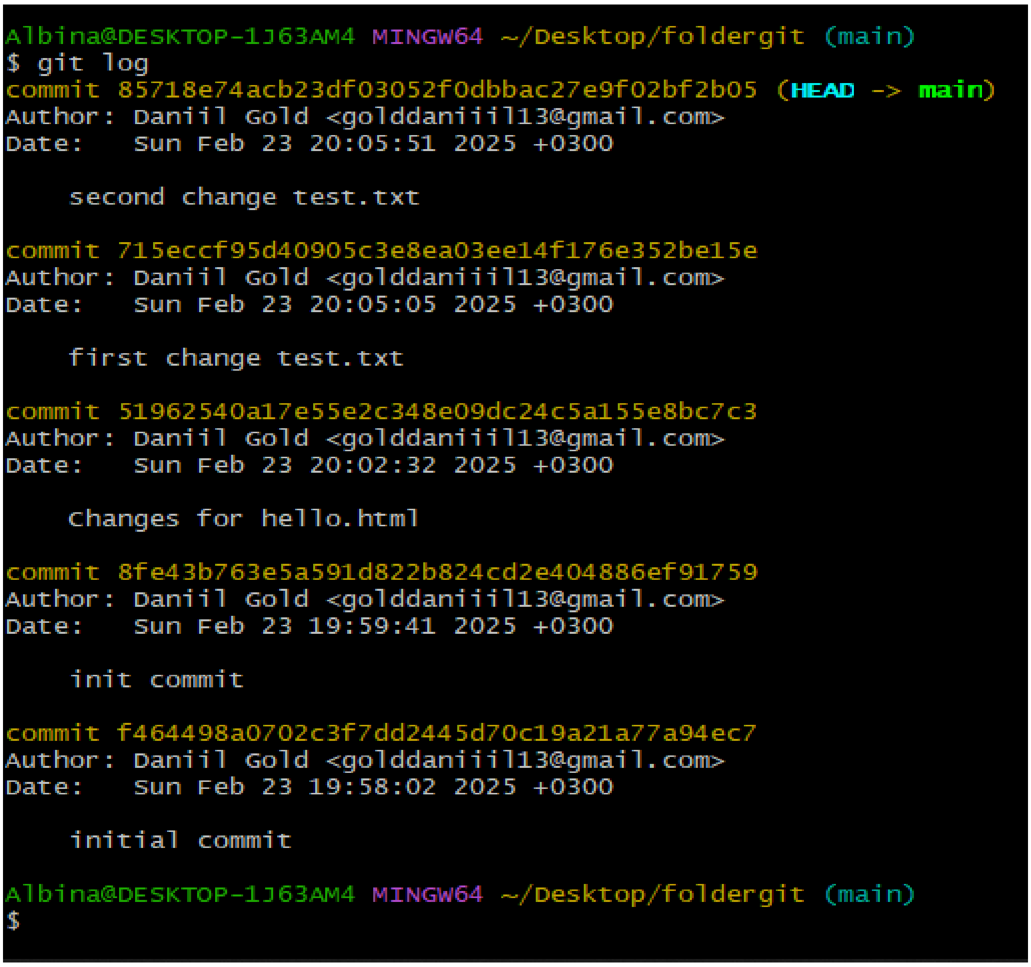
```
Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit (main)
$ git add hello.html

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit (main)
$ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   hello.html


Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit (main)
$ git commit -m "Changes for hello.html"
[main 5196254] Changes for hello.html
 1 file changed, 1 insertion(+)

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit (main)
$
```

6. Изменить еще один файл. Добавить это изменение в индекс git. Изменить файл еще раз. Проверить состояние и произведите коммит проиндексированного изменения. Теперь добавить второе изменение в индекс, а затем проверить состояние с помощью команды git status. Сделать коммит второго изменения.

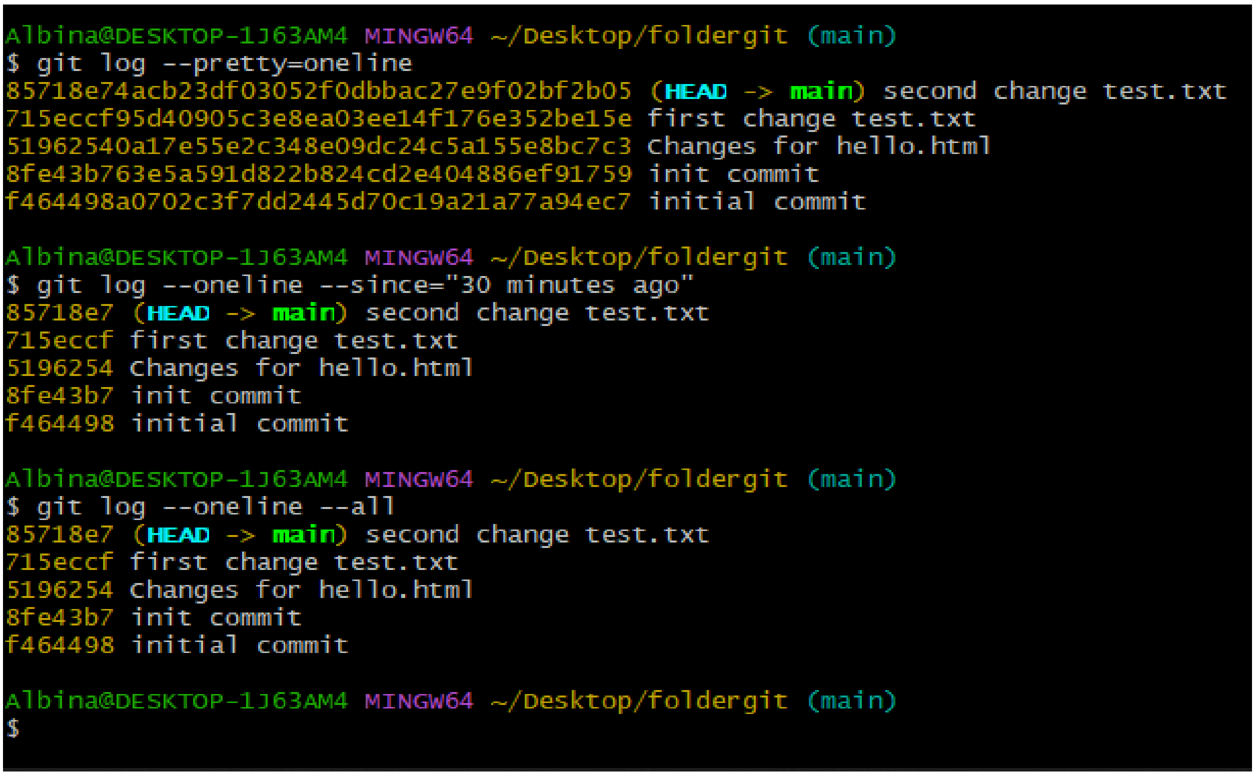


```
Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit (main)
$ git status
On branch main
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   test.txt

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

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit (main)
$ git add test.txt

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit (main)
$ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   test.txt


Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit (main)
$ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   test.txt

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:   test.txt


Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit (main)
$ git commit -m "first change test.txt"
[main 715eccf] first change test.txt
 1 file changed, 1 insertion(+)

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit (main)
$ git status
On branch 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:   test.txt

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

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit (main)
$ git add .

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit (main)
$ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   test.txt


Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit (main)
$ git commit -m "second change test.txt"
[main 85718e7] second change test.txt
 1 file changed, 2 insertions(+), 1 deletion(-)

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit (main)
$ |
```

7. Просмотреть историю коммитов с помощью команды git log. Ознакомиться с параметрами команды и использовать некоторые из них для различного формата отображения истории коммитов.

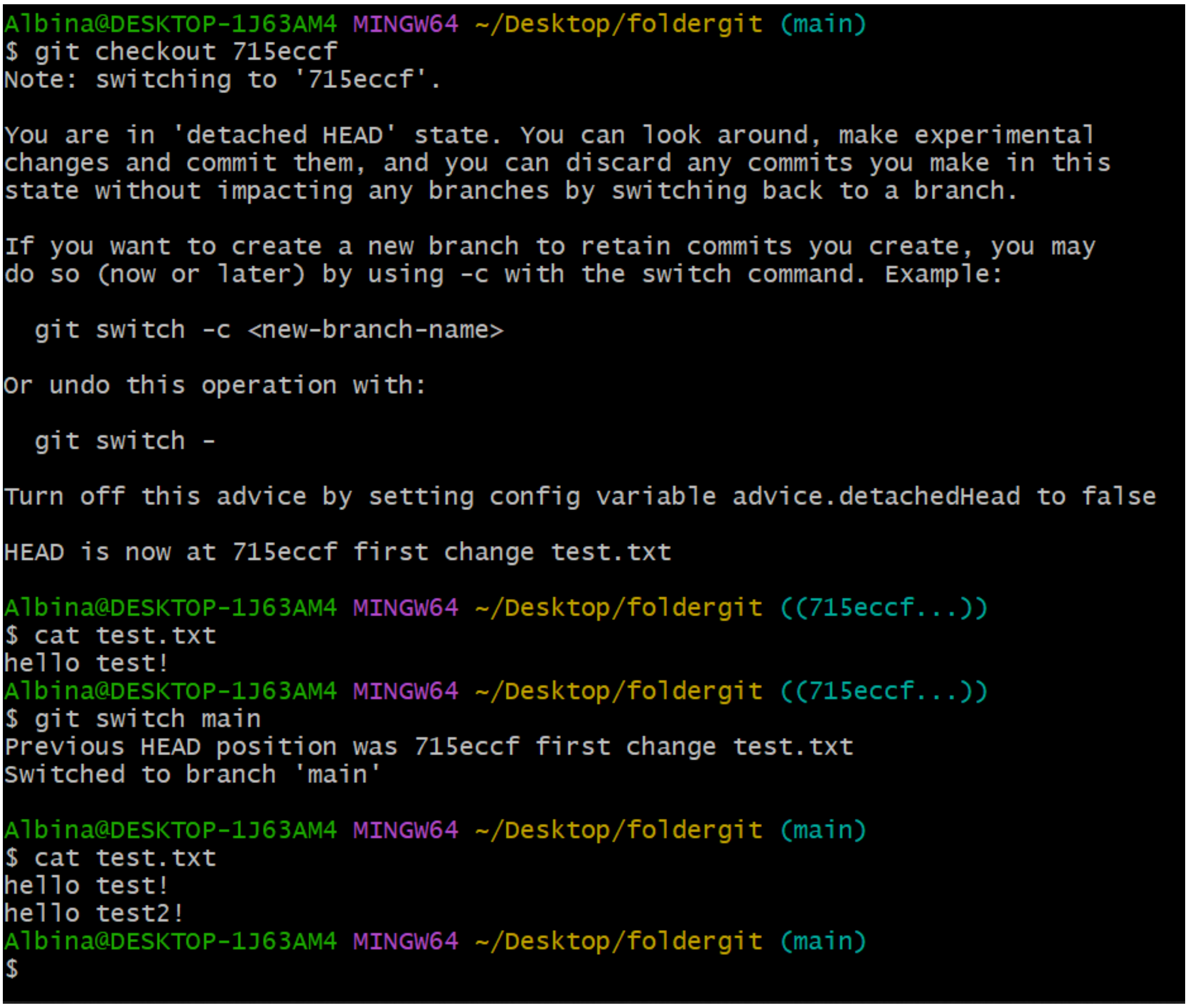


```
Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit (main)
$ git log
commit 85718e74acb23df03052f0dbbac27e9f02bf2b05 (HEAD -> main)
Author: Daniil Gold <golddaniiil13@gmail.com>
Date:   Sun Feb 23 20:05:51 2025 +0300

    second change test.txt

commit 715eccf95d40905c3e8ea03ee14f176e352be15e
Author: Daniil Gold <golddaniiil13@gmail.com>
Date:   Sun Feb 23 20:05:05 2025 +0300

    first change test.txt

commit 51962540a17e55e2c348e09dc24c5a155e8bc7c3
Author: Daniil Gold <golddaniiil13@gmail.com>
Date:   Sun Feb 23 20:02:32 2025 +0300

    Changes for hello.html

commit 8fe43b763e5a591d822b824cd2e404886ef91759
Author: Daniil Gold <golddaniiil13@gmail.com>
Date:   Sun Feb 23 19:59:41 2025 +0300

    init commit

commit f464498a0702c3f7dd2445d70c19a21a77a94ec7
Author: Daniil Gold <golddaniiil13@gmail.com>
Date:   Sun Feb 23 19:58:02 2025 +0300

    initial commit

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit (main)
$
```



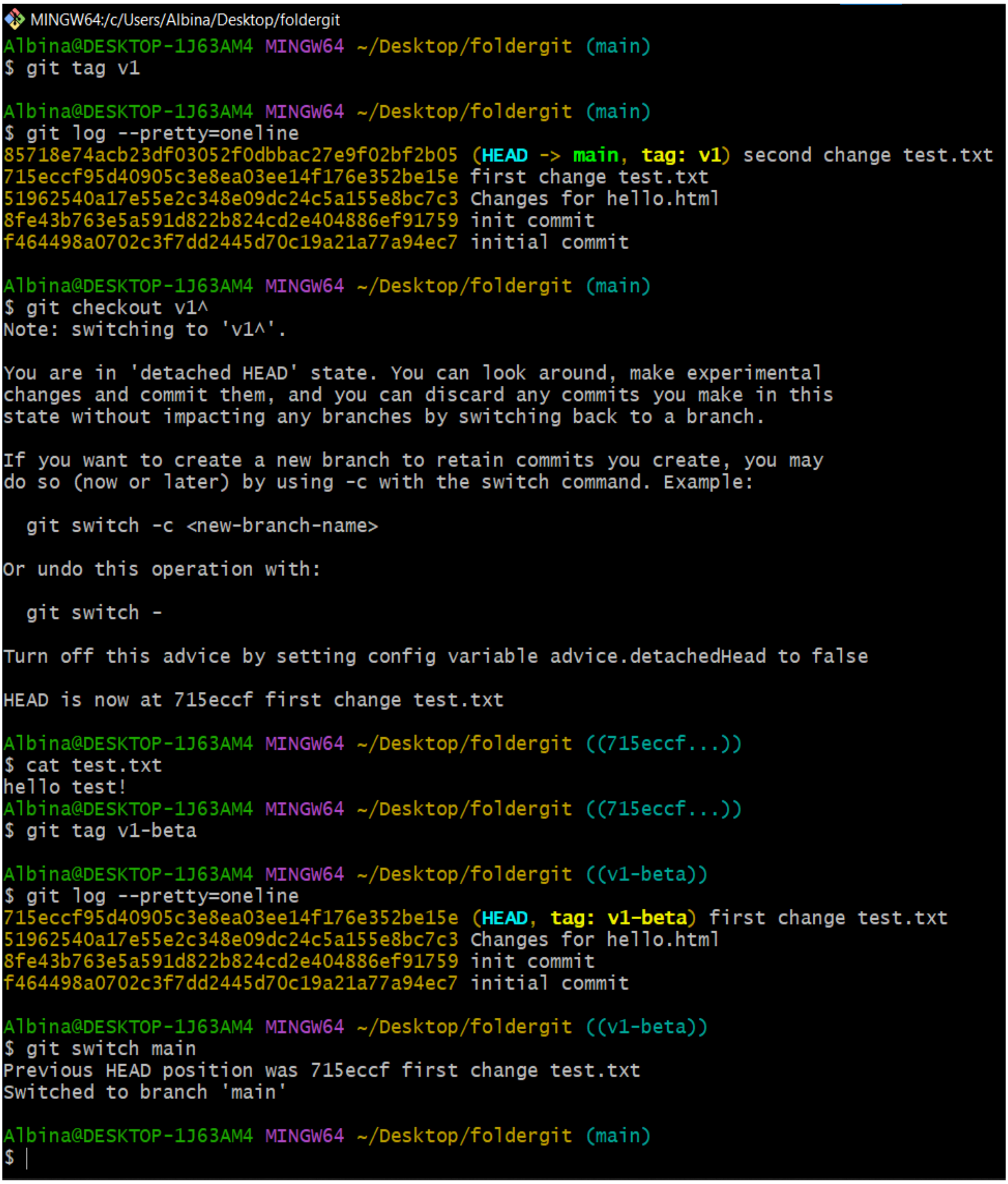
```
Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit (main)
$ git log --pretty=oneline
85718e74acb23df03052f0dbbac27e9f02bf2b05 (HEAD -> main) second change test.txt
715eccf95d40905c3e8ea03ee14f176e352be15e first change test.txt
51962540a17e55e2c348e09dc24c5a155e8bc7c3 Changes for hello.html
8fe43b763e5a591d822b824cd2e404886ef91759 init commit
f464498a0702c3f7dd2445d70c19a21a77a94ec7 initial commit

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit (main)
$ git log --oneline --since="30 minutes ago"
85718e7 (HEAD -> main) second change test.txt
715eccf first change test.txt
5196254 Changes for hello.html
8fe43b7 init commit
f464498 initial commit

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit (main)
$ git log --oneline --all
85718e7 (HEAD -> main) second change test.txt
715eccf first change test.txt
5196254 Changes for hello.html
8fe43b7 init commit
f464498 initial commit

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit (main)
$
```

8. Вернуть рабочий каталог к одному из предыдущих состояний.



```
Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit (main)
$ git checkout 715eccf
Note: switching to '715eccf'.

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 715eccf first change test.txt

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit ((715eccf...))
$ cat test.txt
hello test!
Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit ((715eccf...))
$ git switch main
Previous HEAD position was 715eccf first change test.txt
Switched to branch 'main'

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit (main)
$ cat test.txt
hello test!
hello test2!
Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit (main)
$
```

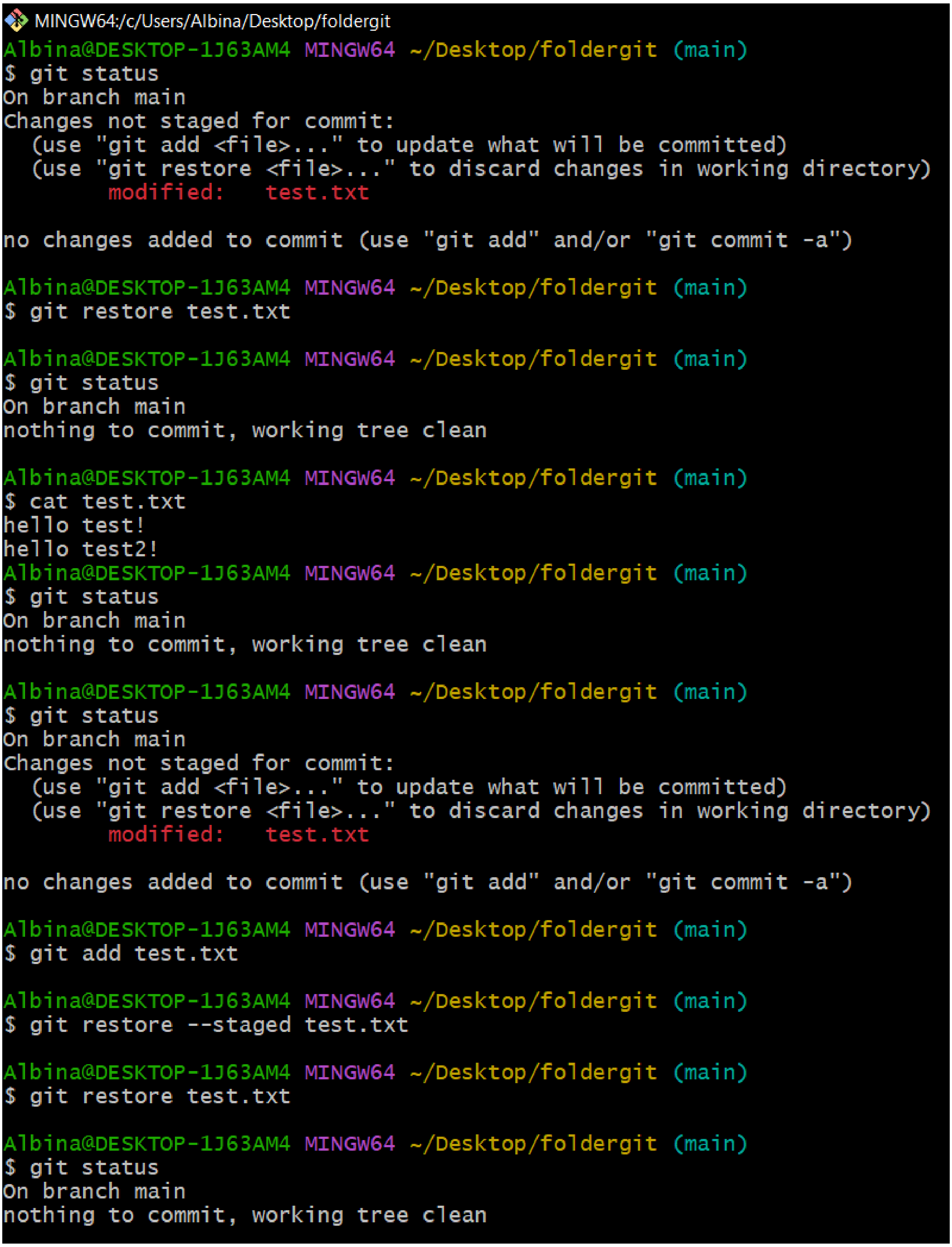
9. Изучить, как создавать теги для коммитов для использования в будущем.



MINGW64:/c/Users/Albina/Desktop/foldergit

```
Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit (main)
$ git tag v1

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit (main)
$ git log --pretty=oneline
85718e74acb23df03052f0dbbac27e9f02bf2b05 (HEAD -> main, tag: v1) second change test.txt
715eccf95d40905c3e8ea03ee14f176e352be15e first change test.txt
51962540a17e55e2c348e09dc24c5a155e8bc7c3 Changes for hello.html
8fe43b763e5a591d822b824cd2e404886ef91759 init commit
f464498a0702c3f7dd2445d70c19a21a77a94ec7 initial commit

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit (main)
$ git checkout v1^
Note: switching to 'v1^'.

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 715eccf first change test.txt

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit ((715eccf...))
$ cat test.txt
hello test!
Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit ((715eccf...))
$ git tag v1-beta

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit ((v1-beta))
$ git log --pretty=oneline
715eccf95d40905c3e8ea03ee14f176e352be15e (HEAD, tag: v1-beta) first change test.txt
51962540a17e55e2c348e09dc24c5a155e8bc7c3 Changes for hello.html
8fe43b763e5a591d822b824cd2e404886ef91759 init commit
f464498a0702c3f7dd2445d70c19a21a77a94ec7 initial commit

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit ((v1-beta))
$ git switch main
Previous HEAD position was 715eccf first change test.txt
Switched to branch 'main'

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit (main)
$ |
```

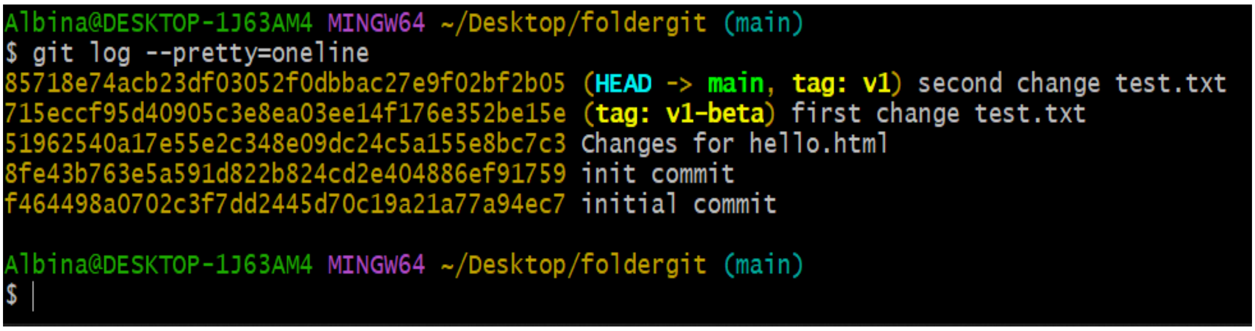
10. Отмените некоторые изменения в рабочем каталоге (до и после индексирования)



MINGW64:/c/Users/Albina/Desktop/foldergit

```
Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit (main)
$ git status
On branch 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:   test.txt

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

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit (main)
$ git restore test.txt

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit (main)
$ git status
On branch main
nothing to commit, working tree clean

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit (main)
$ cat test.txt
hello test!
hello test2!
Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit (main)
$ git status
On branch main
nothing to commit, working tree clean

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit (main)
$ git status
On branch 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:   test.txt

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

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit (main)
$ git add test.txt

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit (main)
$ git restore --staged test.txt

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit (main)
$ git restore test.txt

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit (main)
$ git status
On branch main
nothing to commit, working tree clean
```

11. Отменить один из коммитов в локальном репозитории.

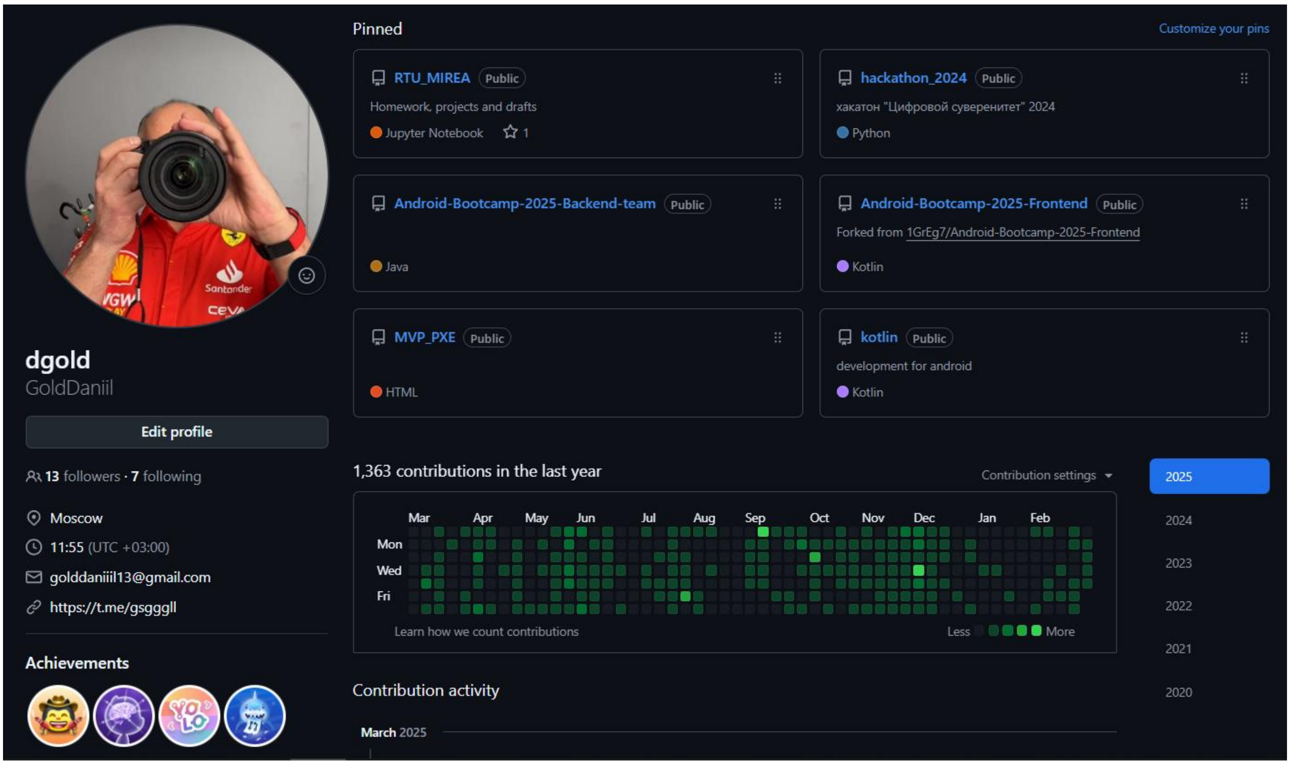


```
Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit (main)
$ git log --pretty=oneline
85718e74acb23df03052f0dbbac27e9f02bf2b05 (HEAD -> main, tag: v1) second change test.txt
715eccf95d40905c3e8ea03ee14f176e352be15e (tag: v1-beta) first change test.txt
51962540a17e55e2c348e09dc24c5a155e8bc7c3 Changes for hello.html
8fe43b763e5a591d822b824cd2e404886ef91759 init commit
f464498a0702c3f7dd2445d70c19a21a77a94ec7 initial commit

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/foldergit (main)
$ 
```

ЧАСТЬ 2. СИСТЕМЫ УПРАВЛЕНИЯ РЕПОЗИТОРИЯМИ

1. Создать аккаунты на GitHub.

Уже 5 лет создан:



Pinned

Customize your pins

RTU_MIREA Public
Homework, projects and drafts
Jupyter Notebook ☆ 1

hackathon_2024 Public
хакатон "Цифровой суверенитет" 2024
Python

Android-Bootcamp-2025-Backend-team Public
Java

Android-Bootcamp-2025-Frontend Public
Forked from 1GrEg7/Android-Bootcamp-2025-Frontend
Kotlin

MVP_PXE Public
HTML

kotlin Public
development for android
Kotlin

dgold
GoldDaniil

Edit profile

13 followers · 7 following

Moscow
11:55 (UTC +03:00)
golddaniiil13@gmail.com
https://t.me/gsgggll

Achievements

1,363 contributions in the last year

Contribution settings

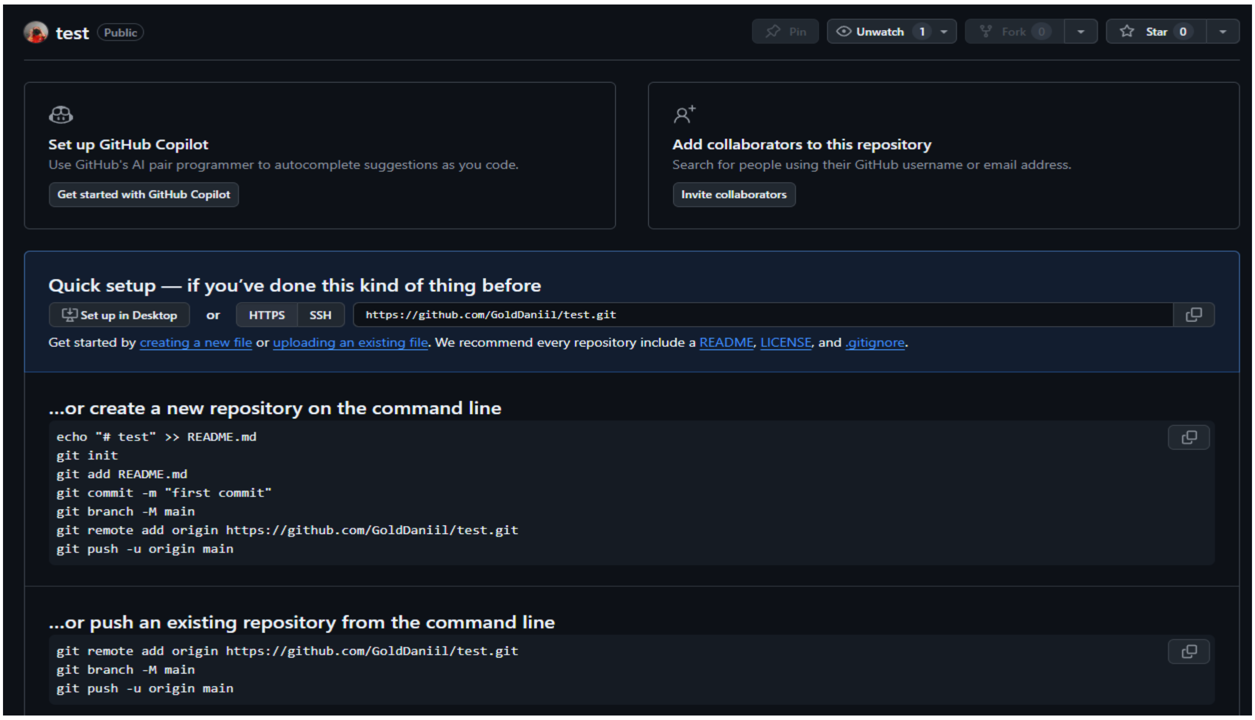
2025
2024
2023
2022
2021
2020

Contribution activity

March 2025

2. Создать репозиторий на GitHub и на локальной машине, согласно выбранной теме проекта

Репозиторий test:



test Public

Set up GitHub Copilot
Use GitHub's AI pair programmer to autocomplete suggestions as you code.
Get started with GitHub Copilot

Add collaborators to this repository
Search for people using their GitHub username or email address.
Invite collaborators

Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS SSH https://github.com/GoldDaniil/test.git

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

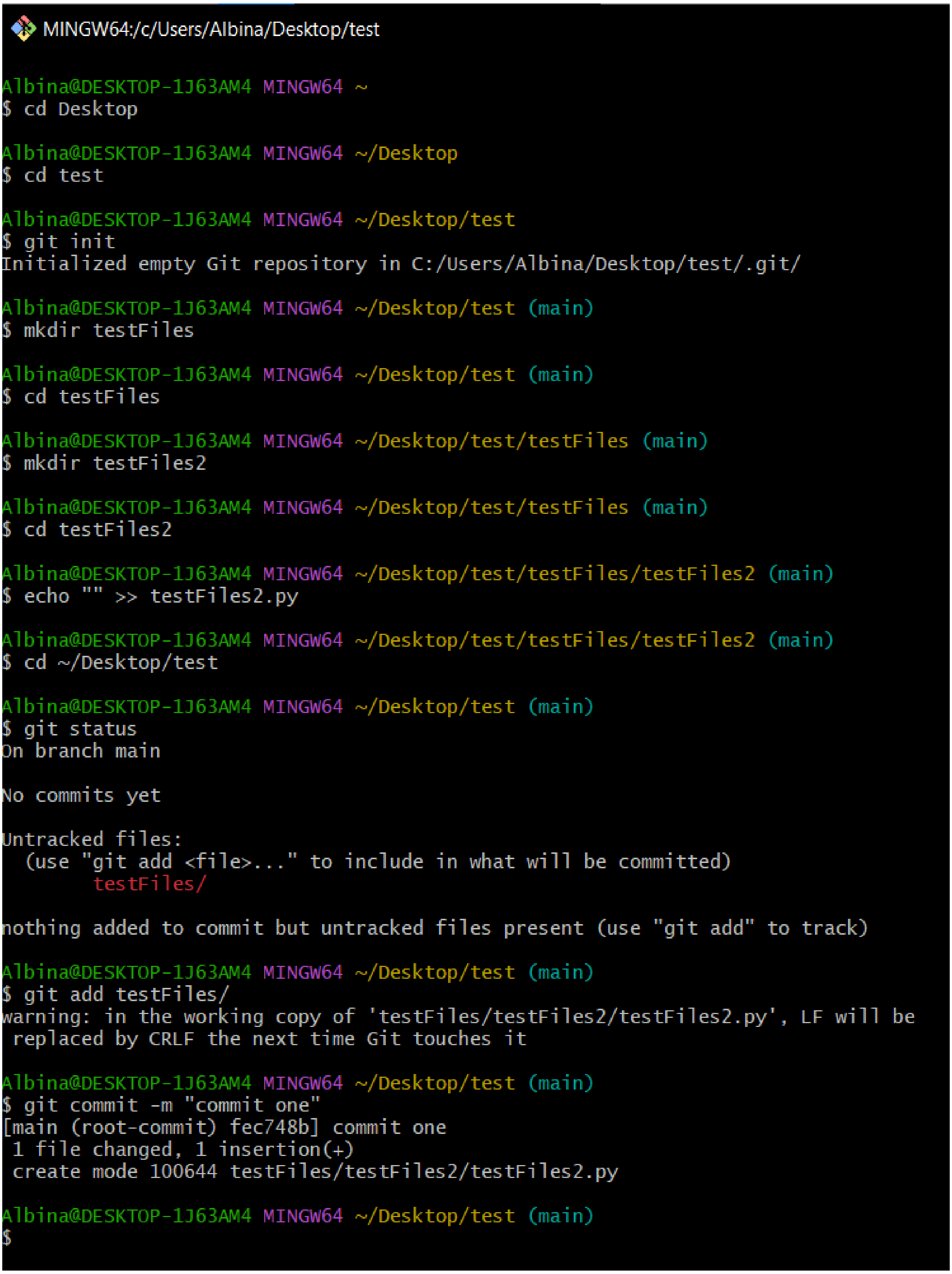
...or create a new repository on the command line

```
echo "# test" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/GoldDaniil/test.git
git push -u origin main
```

...or push an existing repository from the command line

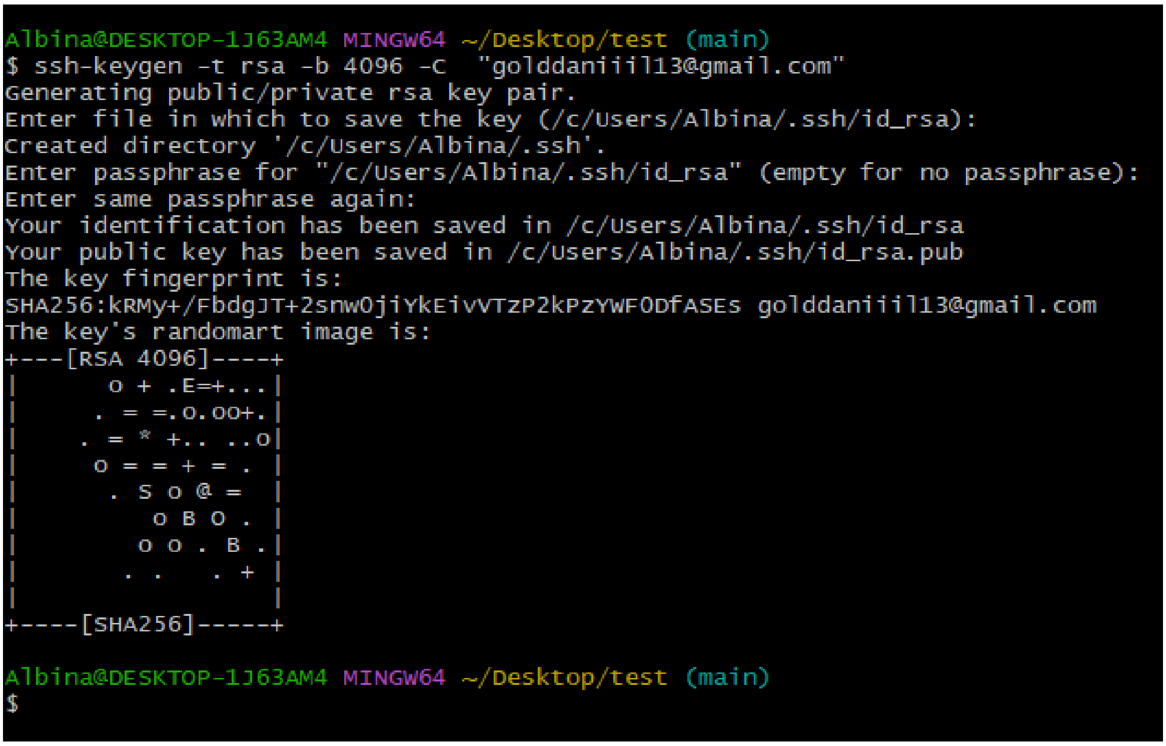
```
git remote add origin https://github.com/GoldDaniil/test.git
git branch -M main
git push -u origin main
```

3. Создайте несколько файлов на локальной машине при помощи консоли

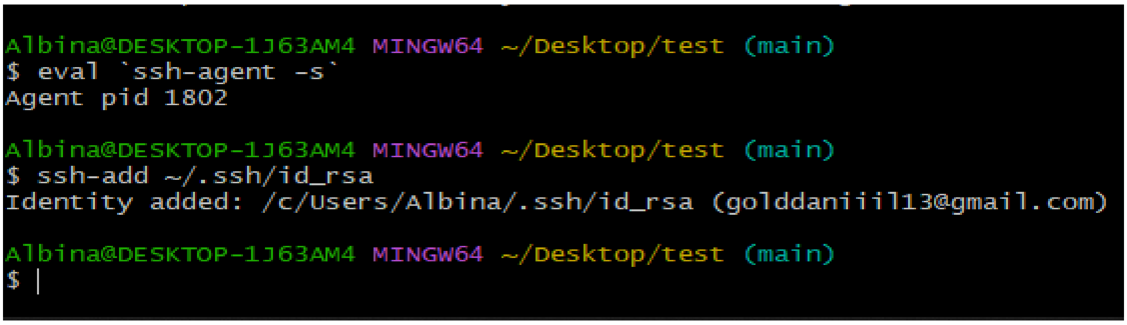


```
MINGW64:/c/Users/Albina/Desktop/test

Albina@DESKTOP-1J63AM4 MINGW64 ~
$ cd Desktop

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop
$ cd test

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/test
$ git init
Initialized empty Git repository in C:/Users/Albina/Desktop/test/.git/

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/test (main)
$ mkdir testFiles

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/test (main)
$ cd testFiles

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/test/testFiles (main)
$ mkdir testFiles2

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/test/testFiles (main)
$ cd testFiles2

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/test/testFiles/testFiles2 (main)
$ echo "" >> testFiles2.py

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/test/testFiles/testFiles2 (main)
$ cd ~/Desktop/test

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/test (main)
$ git status
On branch main

No commits yet

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

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

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/test (main)
$ git add testFiles/
warning: in the working copy of 'testFiles/testFiles2/testFiles2.py', LF will be
 replaced by CRLF the next time Git touches it

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/test (main)
$ git commit -m "commit one"
[main (root-commit) fec748b] commit one
 1 file changed, 1 insertion(+)
 create mode 100644 testFiles/testFiles2/testFiles2.py

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/test (main)
$
```

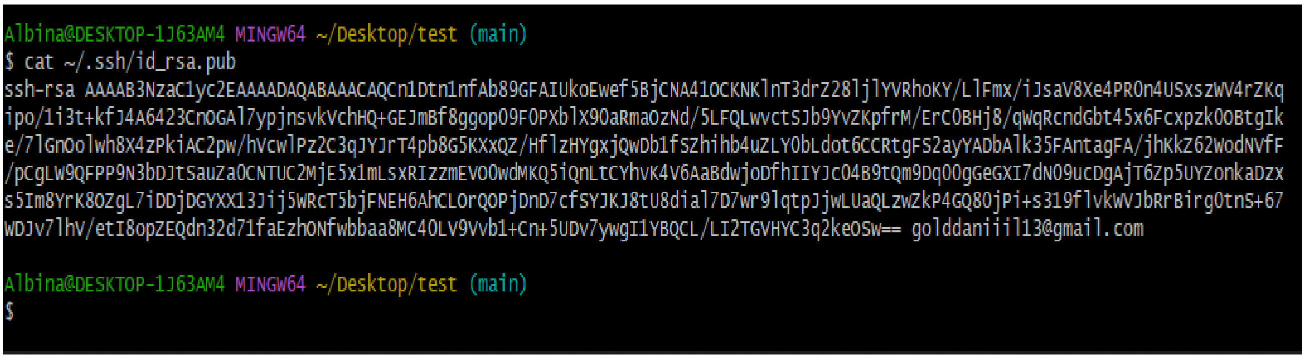
4. Создайте SSH-ключ для авторизации



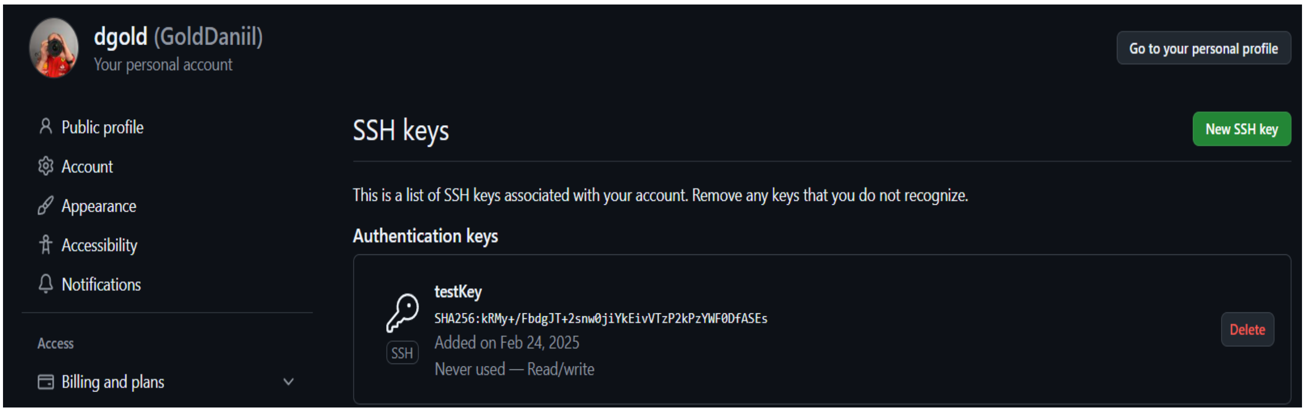
```
Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/test (main)
$ ssh-keygen -t rsa -b 4096 -C  "golddaniiil13@gmail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/Albina/.ssh/id_rsa):
Created directory '/c/Users/Albina/.ssh'.
Enter passphrase for "/c/Users/Albina/.ssh/id_rsa" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/Albina/.ssh/id_rsa
Your public key has been saved in /c/Users/Albina/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:kRMy+/FbdgJT+2snw0jiYkEivVTzP2kPzYWF0DfASEs golddaniiil13@gmail.com
The key's randomart image is:
+---[RSA 4096]----+
|      o + .E=+...|
|     . = =.o.oo+.|
|    . = * +.. ..o|
|     o = = + = . |
|      . S o @ =  |
|         o B o . |
|        o o . B .|
|       . .   . + |
|                 |
+----[SHA256]-----+

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/test (main)
$
```



```
Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/test (main)
$ eval `ssh-agent -s`
Agent pid 1802

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/test (main)
$ ssh-add ~/.ssh/id_rsa
Identity added: /c/Users/Albina/.ssh/id_rsa (golddaniiil13@gmail.com)

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/test (main)
$ |
```



```
Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/test (main)
$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQCn1Dtn1nfAb89GFAIUkoEwef5BjCNA41OCKNKlnT3drZ28ljlYVRhoKY/LlFmx/iJsaV8Xe4PROn4USxszWV4rZKq
ipo/1i3t+kfJ4A6423CnOGAl7ypjnsvkVchHQ+GEJmBf8ggop09F0PXblX90aRmaOzNd/5LFQLwvctSJb9YvZKpfrM/ErC0BHj8/qWqRcndGbt45x6FcxpzkO0BtgIk
e/7lGnOolwh8X4zPkiAC2pw/hVcwlPz2C3qJYJrT4pb8G5KXxQZ/HflzHYgxjQwDb1fSZhihb4uZLY0bLdot6CCRtgFS2ayYADbAlk35FAntagFA/jhKkZ62WodNVfF
/pCgLW9QFPP9N3bDJtSauZa0CNTUC2MjE5x1mLsxRIzzmEVO0wdMKQ5iQnLtCYhvK4V6AaBdwjoDfhIIYJc04B9tQm9Dq00gGeGXI7dN09ucDgAjT6Zp5UYZonkaDzx
s5Im8YrK8OZgL7iDDjDGYXX13Jij5WRcT5bjFNEH6AhCLOrQOPjDnD7cfSYJKJ8tU8dial7D7wr9lqtpJjwLUaQLzwZkP4GQ80jPi+s319flvkWVJbRrBirg0tnS+67
WDJv7lhv/etI8opZEQdn32d71faEzhONfwbbaa8MC40LV9Vvb1+Cn+5UDv7ywgI1YBQCL/LI2TGVHYC3q2keOSW== golddaniiil13@gmail.com

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/test (main)
$
```



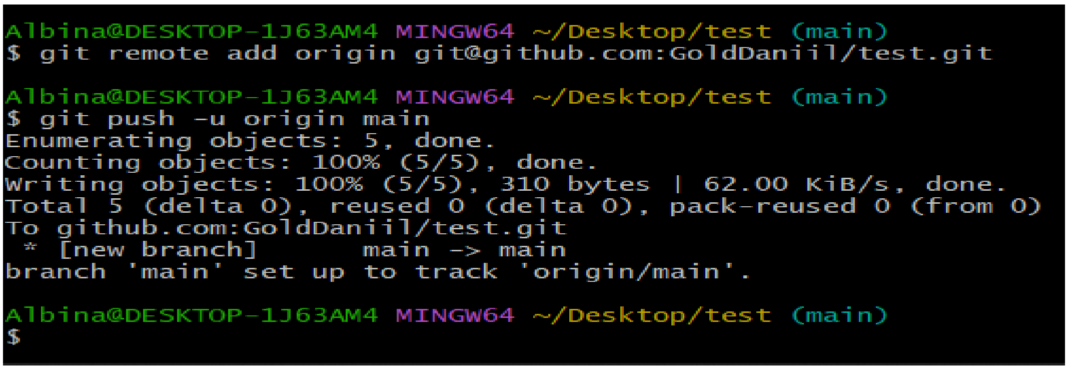
SSH keys

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

Authentication keys

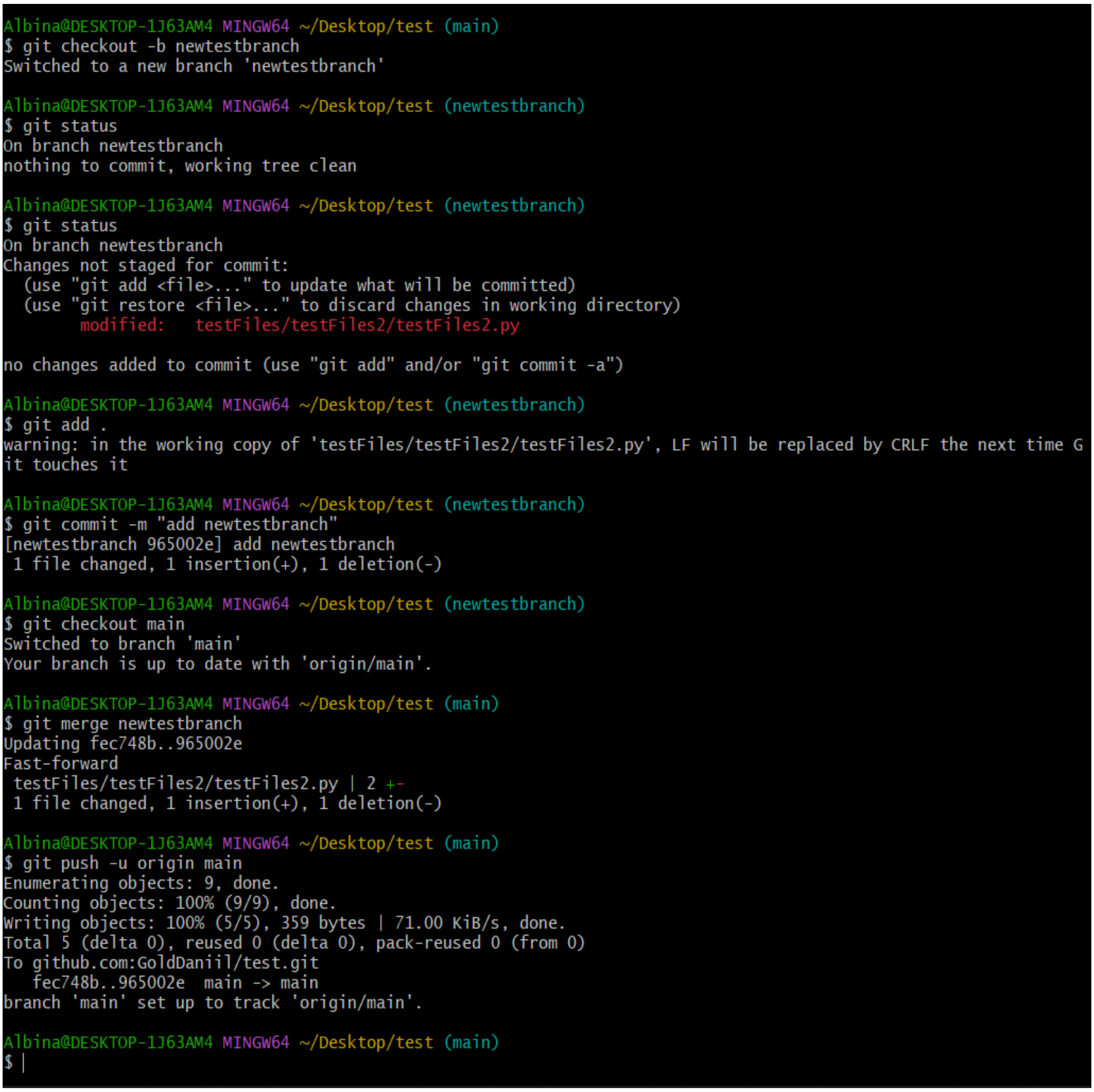
testKey
SHA256:kRMy+/FbdgJT+2snw0jiYkEivVTzP2kPzYWF0DfASEs
SSH
Added on Feb 24, 2025
Never used — Read/write

5. Свяжите репозиторий локальной машины с репозиторием на GitHub при помощи консоли



```
Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/test (main)
$ git remote add origin git@github.com:GoldDaniil/test.git

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/test (main)
$ git push -u origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Writing objects: 100% (5/5), 310 bytes | 62.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:GoldDaniil/test.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/test (main)
$
```

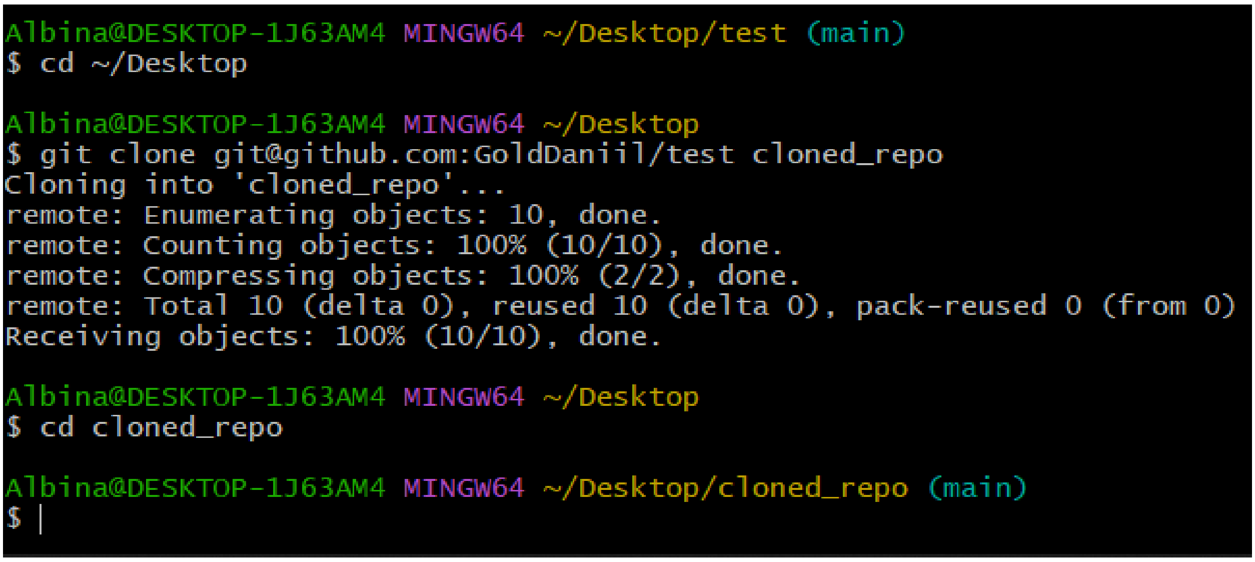
6. Создайте новую ветку в репозитории с помощью команды, произведите в ней какие-нибудь изменения, а после слейте с веткой main



```
Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/test (main)
$ git checkout -b newtestbranch
Switched to a new branch 'newtestbranch'

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/test (newtestbranch)
$ git status
On branch newtestbranch
nothing to commit, working tree clean

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/test (newtestbranch)
$ git status
On branch newtestbranch
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:   testFiles/testFiles2/testFiles2.py

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

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/test (newtestbranch)
$ git add .
warning: in the working copy of 'testFiles/testFiles2/testFiles2.py', LF will be replaced by CRLF the next time G
it touches it

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/test (newtestbranch)
$ git commit -m "add newtestbranch"
[newtestbranch 965002e] add newtestbranch
 1 file changed, 1 insertion(+), 1 deletion(-)

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/test (newtestbranch)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/test (main)
$ git merge newtestbranch
Updating fec748b..965002e
Fast-forward
 testFiles/testFiles2/testFiles2.py | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/test (main)
$ git push -u origin main
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Writing objects: 100% (5/5), 359 bytes | 71.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:GoldDaniil/test.git
   fec748b..965002e  main -> main
branch 'main' set up to track 'origin/main'.

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/test (main)
$ |
```

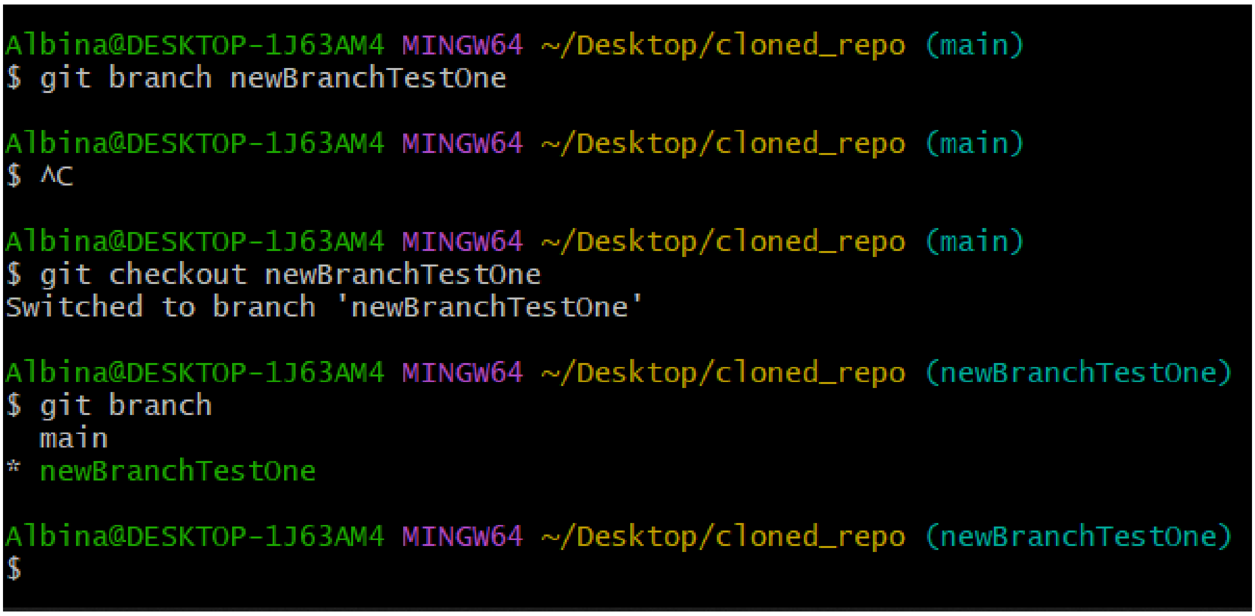
7. Выполните цепочку действий в репозитории, согласно вариантам (Вариант 4):

1) Клонируйте непустой удаленный репозиторий на локальную машину:



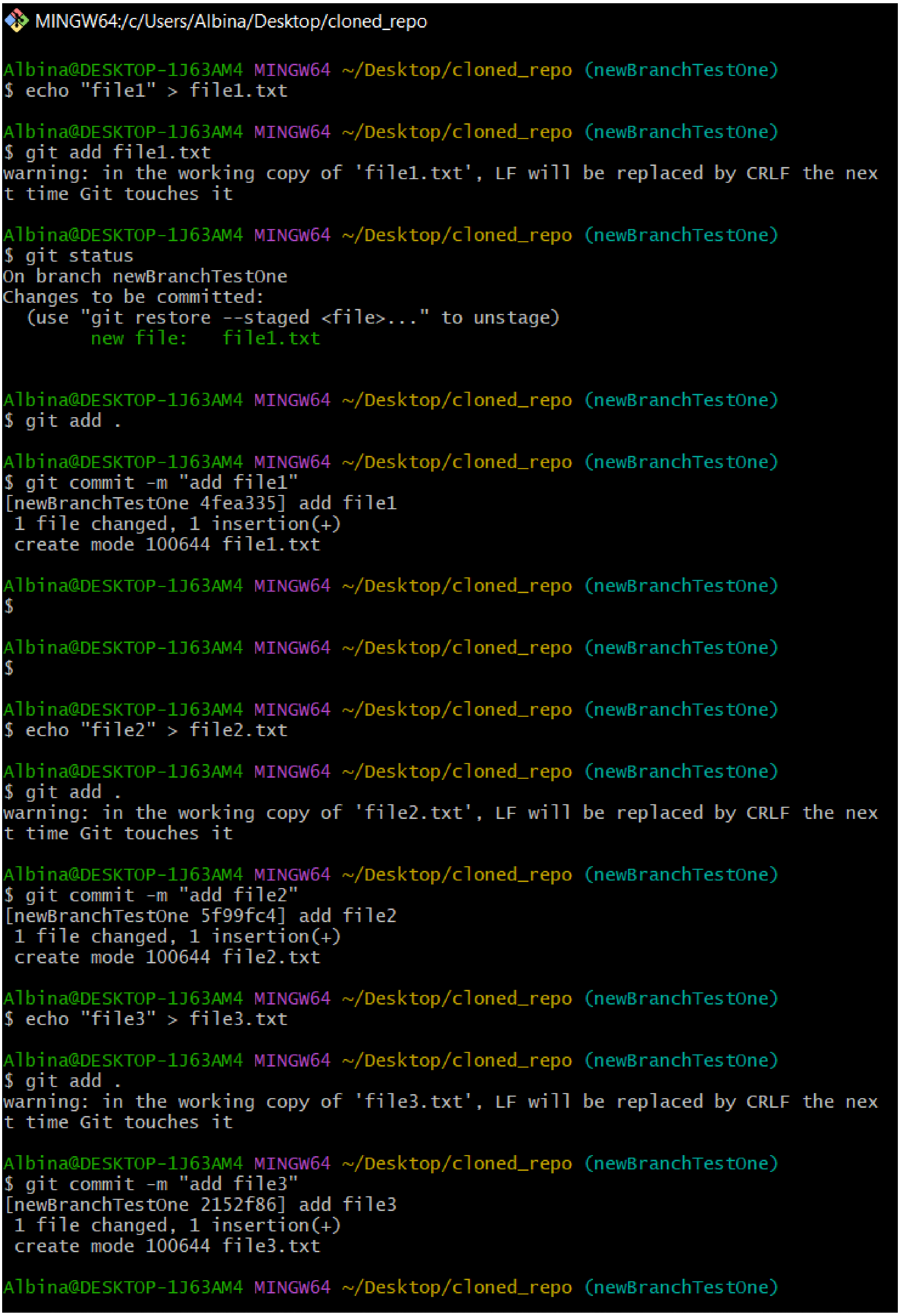
```
Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/test (main)
$ cd ~/Desktop

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop
$ git clone git@github.com:GoldDaniil/test cloned_repo
Cloning into 'cloned_repo'...
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 10 (delta 0), reused 10 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (10/10), done.

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop
$ cd cloned_repo

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/cloned_repo (main)
$ |
```

2) Создайте новую ветку и выведите список всех веток:



```
Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/cloned_repo (main)
$ git branch newBranchTestOne

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/cloned_repo (main)
$ ^C

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/cloned_repo (main)
$ git checkout newBranchTestOne
Switched to branch 'newBranchTestOne'

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/cloned_repo (newBranchTestOne)
$ git branch
  main
* newBranchTestOne

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/cloned_repo (newBranchTestOne)
$
```

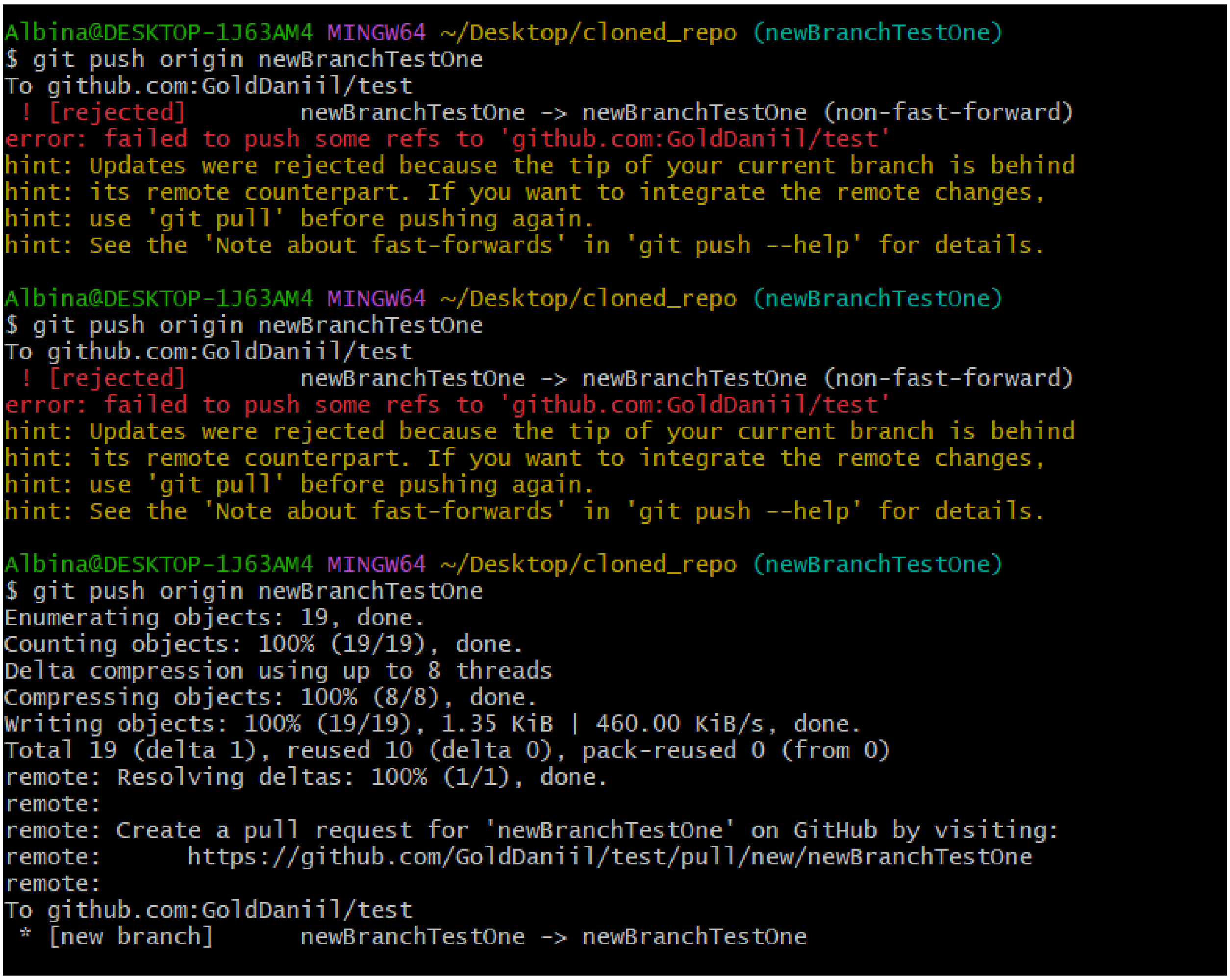
3) Произведите 3 коммитов в новой ветке в разные файлы:



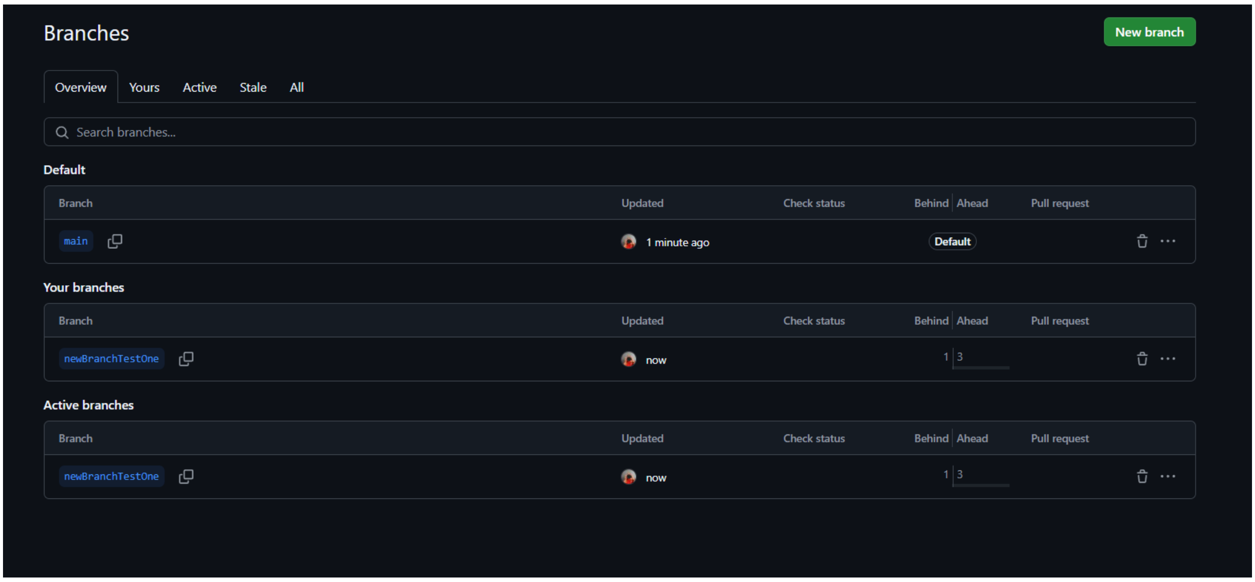
MINGW64:/c/Users/Albina/Desktop/cloned_repo

```
Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/cloned_repo (newBranchTestOne)
$ echo "file1" > file1.txt

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/cloned_repo (newBranchTestOne)
$ git add file1.txt
warning: in the working copy of 'file1.txt', LF will be replaced by CRLF the nex
t time Git touches it

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/cloned_repo (newBranchTestOne)
$ git status
On branch newBranchTestOne
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   file1.txt


Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/cloned_repo (newBranchTestOne)
$ git add .

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/cloned_repo (newBranchTestOne)
$ git commit -m "add file1"
[newBranchTestOne 4fea335] add file1
 1 file changed, 1 insertion(+)
 create mode 100644 file1.txt

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/cloned_repo (newBranchTestOne)
$

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/cloned_repo (newBranchTestOne)
$

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/cloned_repo (newBranchTestOne)
$ echo "file2" > file2.txt

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/cloned_repo (newBranchTestOne)
$ git add .
warning: in the working copy of 'file2.txt', LF will be replaced by CRLF the nex
t time Git touches it

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/cloned_repo (newBranchTestOne)
$ git commit -m "add file2"
[newBranchTestOne 5f99fc4] add file2
 1 file changed, 1 insertion(+)
 create mode 100644 file2.txt

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/cloned_repo (newBranchTestOne)
$ echo "file3" > file3.txt

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/cloned_repo (newBranchTestOne)
$ git add .
warning: in the working copy of 'file3.txt', LF will be replaced by CRLF the nex
t time Git touches it

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/cloned_repo (newBranchTestOne)
$ git commit -m "add file3"
[newBranchTestOne 2152f86] add file3
 1 file changed, 1 insertion(+)
 create mode 100644 file3.txt

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/cloned_repo (newBranchTestOne)
```

4) Выгрузите изменения в удаленный репозиторий:



```
Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/cloned_repo (newBranchTestOne)
$ git push origin newBranchTestOne
To github.com:GoldDaniil/test
 ! [rejected]        newBranchTestOne -> newBranchTestOne (non-fast-forward)
error: failed to push some refs to 'github.com:GoldDaniil/test'
hint: Updates were rejected because the tip of your current branch is behind
hint: its remote counterpart. If you want to integrate the remote changes,
hint: use 'git pull' before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/cloned_repo (newBranchTestOne)
$ git push origin newBranchTestOne
To github.com:GoldDaniil/test
 ! [rejected]        newBranchTestOne -> newBranchTestOne (non-fast-forward)
error: failed to push some refs to 'github.com:GoldDaniil/test'
hint: Updates were rejected because the tip of your current branch is behind
hint: its remote counterpart. If you want to integrate the remote changes,
hint: use 'git pull' before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/cloned_repo (newBranchTestOne)
$ git push origin newBranchTestOne
Enumerating objects: 19, done.
Counting objects: 100% (19/19), done.
Delta compression using up to 8 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (19/19), 1.35 KiB | 460.00 KiB/s, done.
Total 19 (delta 1), reused 10 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
remote:
remote: Create a pull request for 'newBranchTestOne' on GitHub by visiting:
remote:      https://github.com/GoldDaniil/test/pull/new/newBranchTestOne
remote:
To github.com:GoldDaniil/test
 * [new branch]      newBranchTestOne -> newBranchTestOne
```



Default

Branch	Updated	Check status	Behind | Ahead	Pull request
main	1 minute ago		Default	

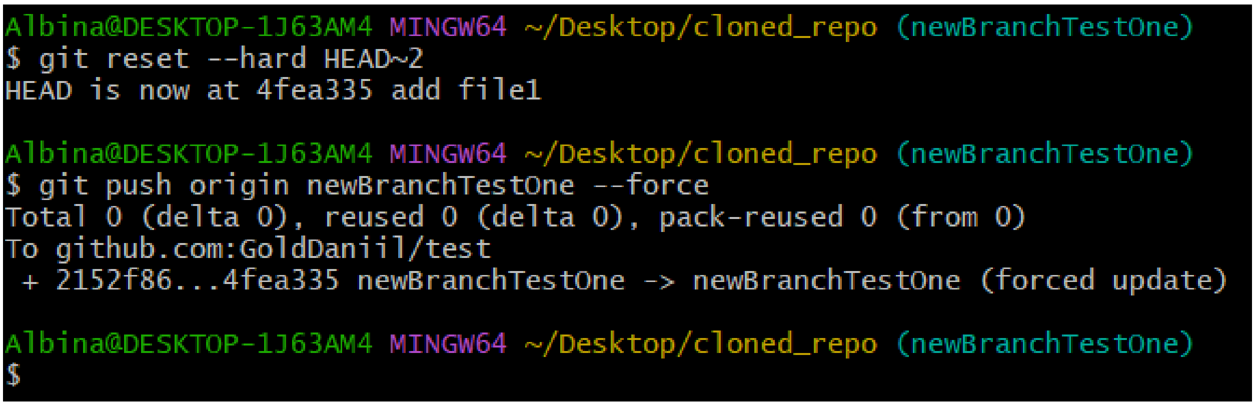
Your branches

Branch	Updated	Check status	Behind | Ahead	Pull request
newBranchTestOne	now		1 | 3	

Active branches

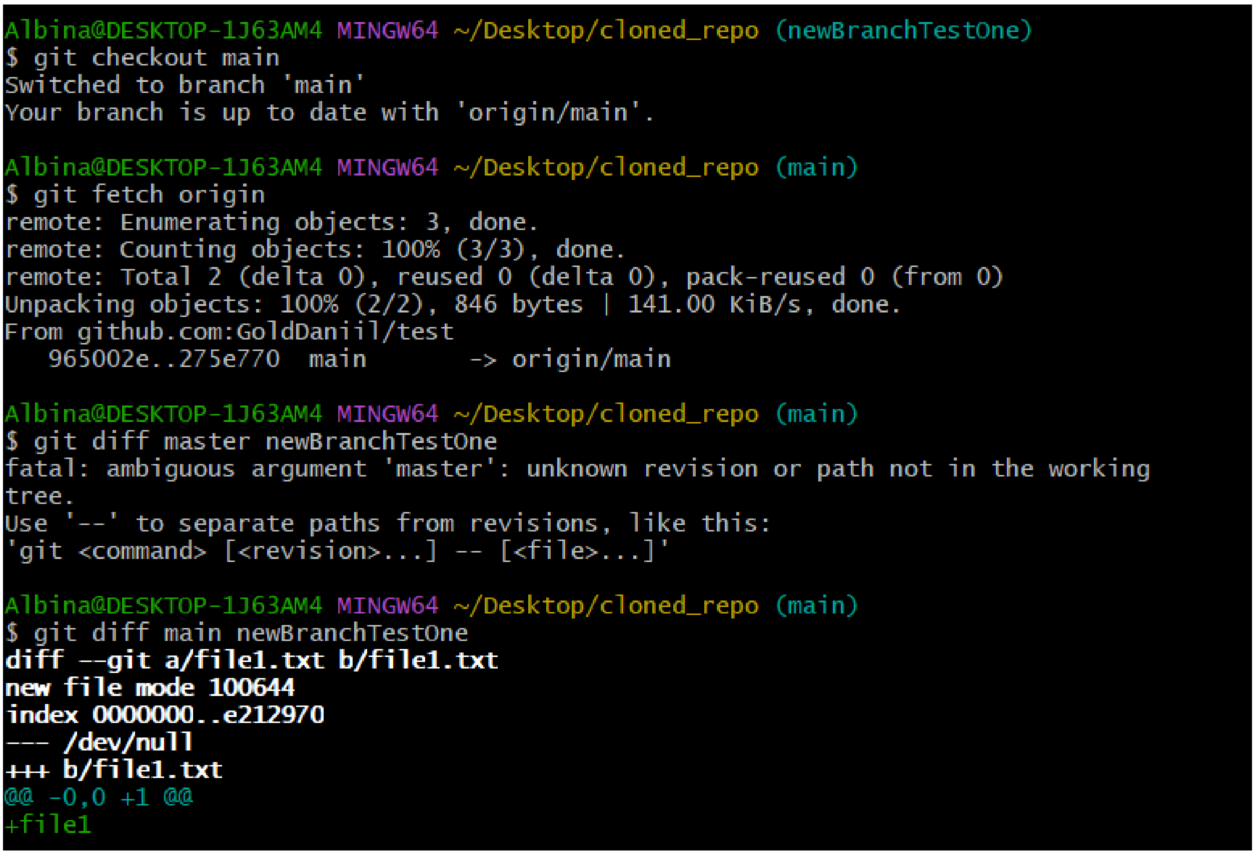
Branch	Updated	Check status	Behind | Ahead	Pull request
newBranchTestOne	now		1 | 3	

5) Откатите в новой ветке предпоследний коммит (в том числе в удаленном репозитории):



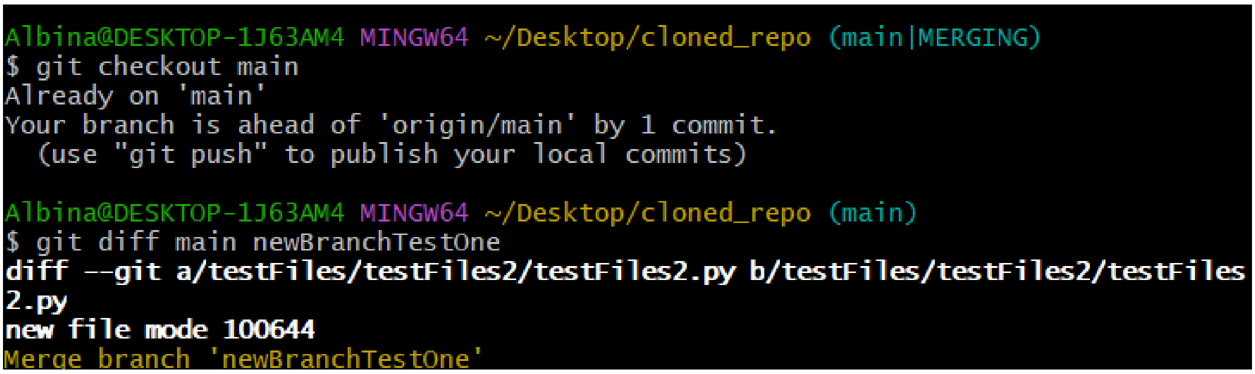
```
Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/cloned_repo (newBranchTestOne)
$ git reset --hard HEAD~2
HEAD is now at 4fea335 add file1

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/cloned_repo (newBranchTestOne)
$ git push origin newBranchTestOne --force
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:GoldDaniil/test
 + 2152f86...4fea335 newBranchTestOne -> newBranchTestOne (forced update)

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/cloned_repo (newBranchTestOne)
$
```

6) Выведите в консоли различия между веткой master и новой веткой:

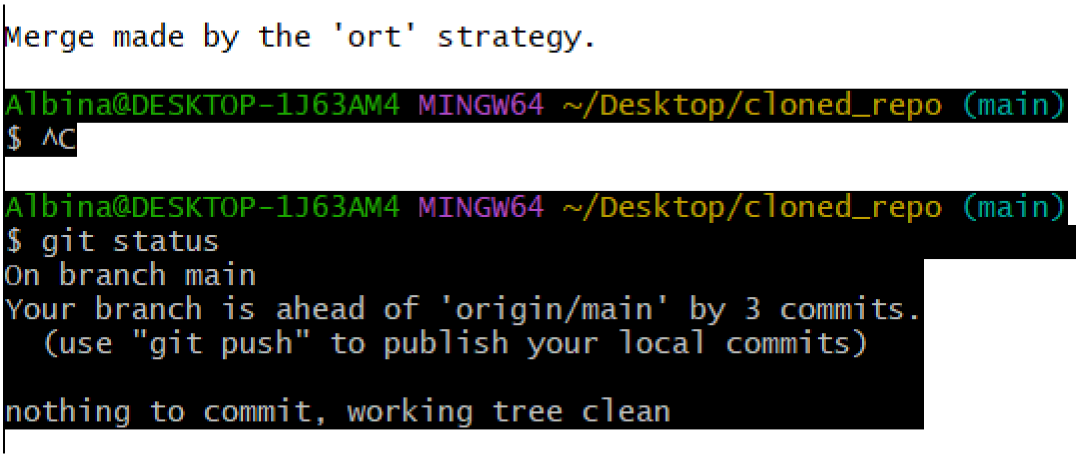


```
Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/cloned_repo (newBranchTestOne)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/cloned_repo (main)
$ git fetch origin
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 2 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (2/2), 846 bytes | 141.00 KiB/s, done.
From github.com:GoldDaniil/test
   965002e..275e770  main       -> origin/main

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/cloned_repo (main)
$ git diff master newBranchTestOne
fatal: ambiguous argument 'master': unknown revision or path not in the working
tree.
Use '--' to separate paths from revisions, like this:
'git <command> [<revision>...] -- [<file>...]'

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/cloned_repo (main)
$ git diff main newBranchTestOne
diff --git a/file1.txt b/file1.txt
new file mode 100644
index 0000000..e212970
--- /dev/null
+++ b/file1.txt
@@ -0,0 +1 @@
+file1
```

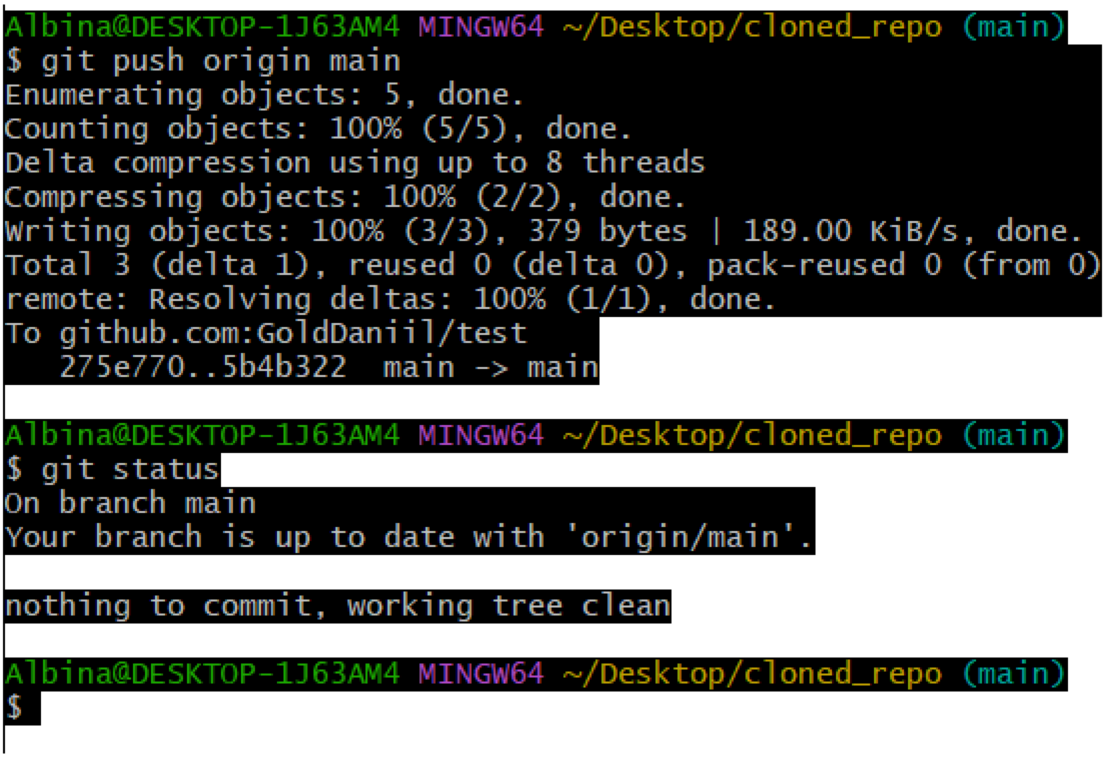
7) Слейте новую ветку с master при помощи merge:



```
Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/cloned_repo (main|MERGING)
$ git checkout main
Already on 'main'
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/cloned_repo (main)
$ git diff main newBranchTestOne
diff --git a/testFiles/testFiles2/testFiles2.py b/testFiles/testFiles2/testFiles
2.py
new file mode 100644
Merge branch 'newBranchTestOne'
```



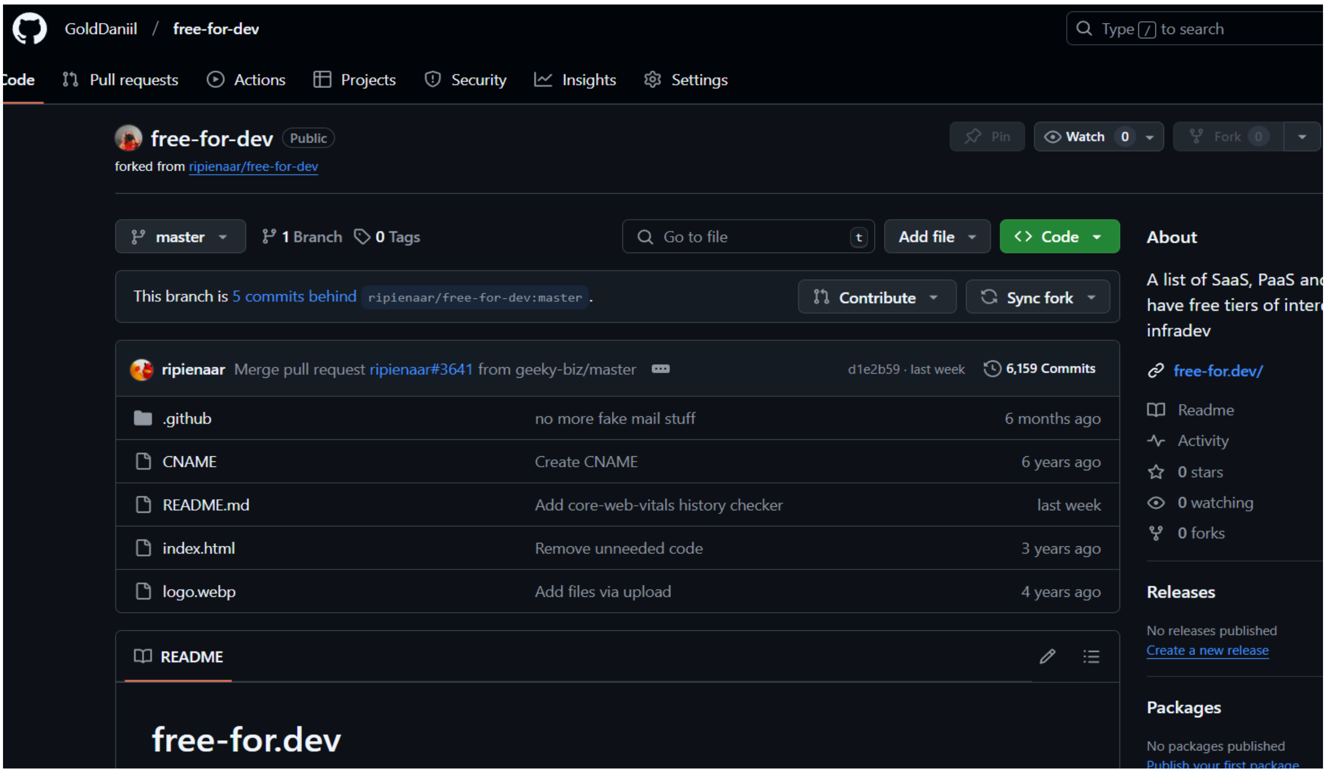
```
Merge made by the 'ort' strategy.

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/cloned_repo (main)
$ ^C

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/cloned_repo (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 3 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```



```
Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/cloned_repo (main)
$ git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 379 bytes | 189.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
To github.com:GoldDaniil/test
   275e770..5b4b322  main -> main

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/cloned_repo (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/cloned_repo (main)
$
```

ЧАСТЬ 3. РАБОТА С ВЕТВЛЕНИЕМ И ОФОРМЛЕНИЕ КОДА

1. Сделайте форк репозитория в соответствии с вашим вариантом



free-for-dev Public

forked from ripienaar/free-for-dev

This branch is 5 commits behind `ripienaar/free-for-dev:master` .

ripienaar Merge pull request ripienaar#3641 from geeky-biz/master d1e2b59 · last week 6,159 Commits

.github	no more fake mail stuff	6 months ago
CNAME	Create CNAME	6 years ago
README.md	Add core-web-vitals history checker	last week
index.html	Remove unneeded code	3 years ago
logo.webp	Add files via upload	4 years ago

free-for.dev

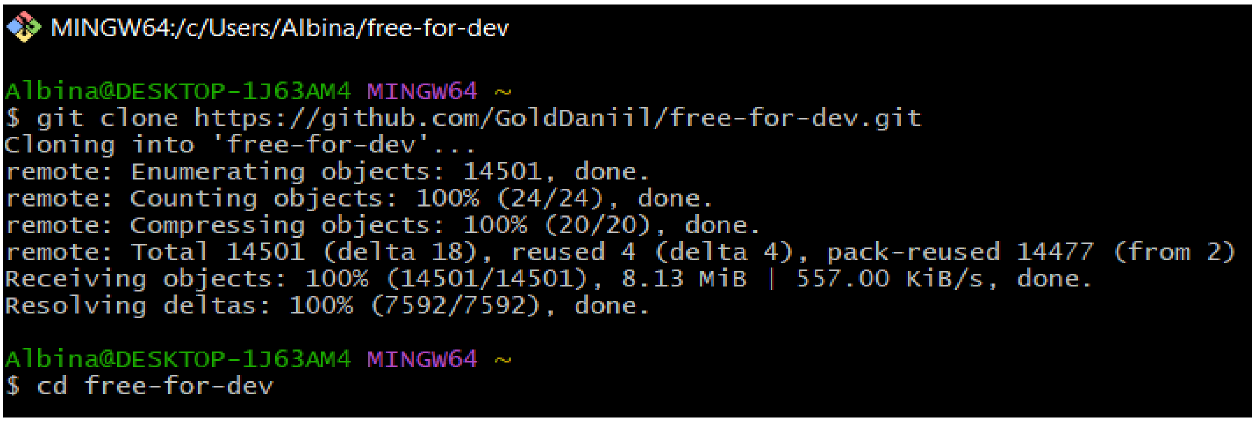
A list of SaaS, PaaS and have free tiers of intere infradev

free-for.dev/

No releases published
Create a new release

No packages published
Publish your first package

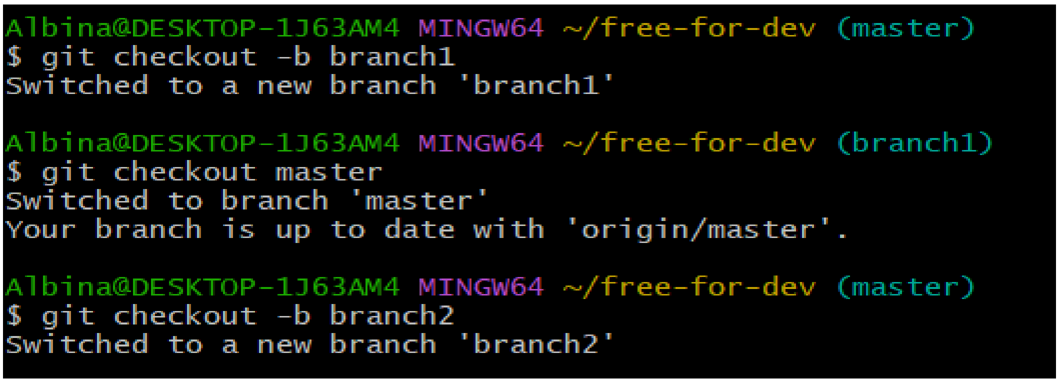
2. Склонируйте его на локальную машину



MINGW64:/c/Users/Albina/free-for-dev

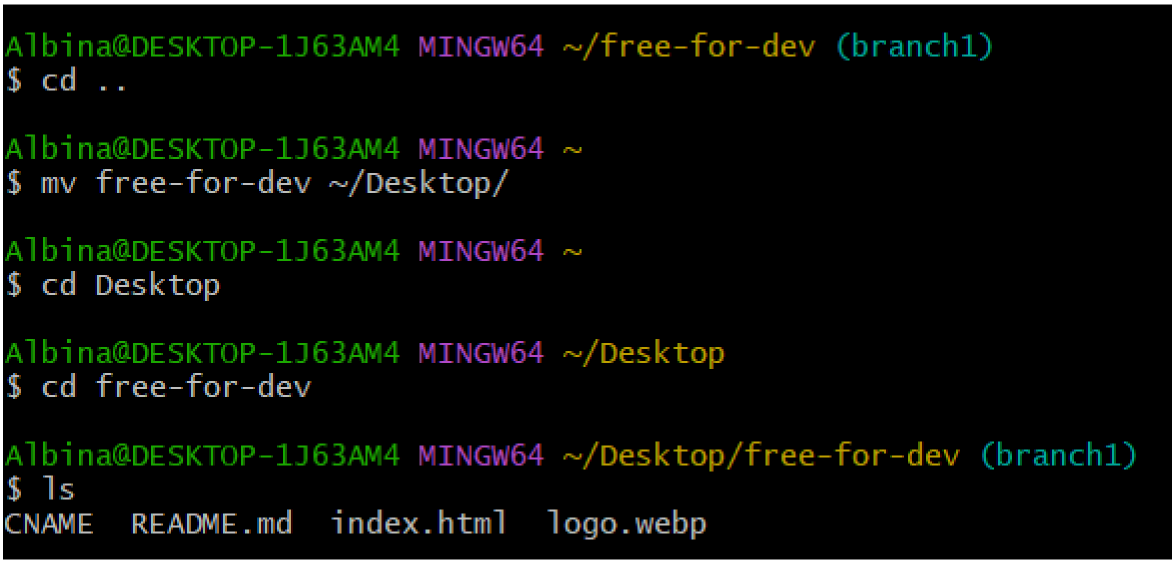
```
Albina@DESKTOP-1J63AM4 MINGW64 ~
$ git clone https://github.com/GoldDaniil/free-for-dev.git
Cloning into 'free-for-dev'...
remote: Enumerating objects: 14501, done.
remote: Counting objects: 100% (24/24), done.
remote: Compressing objects: 100% (20/20), done.
remote: Total 14501 (delta 18), reused 4 (delta 4), pack-reused 14477 (from 2)
Receiving objects: 100% (14501/14501), 8.13 MiB | 557.00 KiB/s, done.
Resolving deltas: 100% (7592/7592), done.

Albina@DESKTOP-1J63AM4 MINGW64 ~
$ cd free-for-dev
```

3. Создайте две ветки branch1 и branch2 от последнего коммита в master



```
Albina@DESKTOP-1J63AM4 MINGW64 ~/free-for-dev (master)
$ git checkout -b branch1
Switched to a new branch 'branch1'

Albina@DESKTOP-1J63AM4 MINGW64 ~/free-for-dev (branch1)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

Albina@DESKTOP-1J63AM4 MINGW64 ~/free-for-dev (master)
$ git checkout -b branch2
Switched to a new branch 'branch2'
```

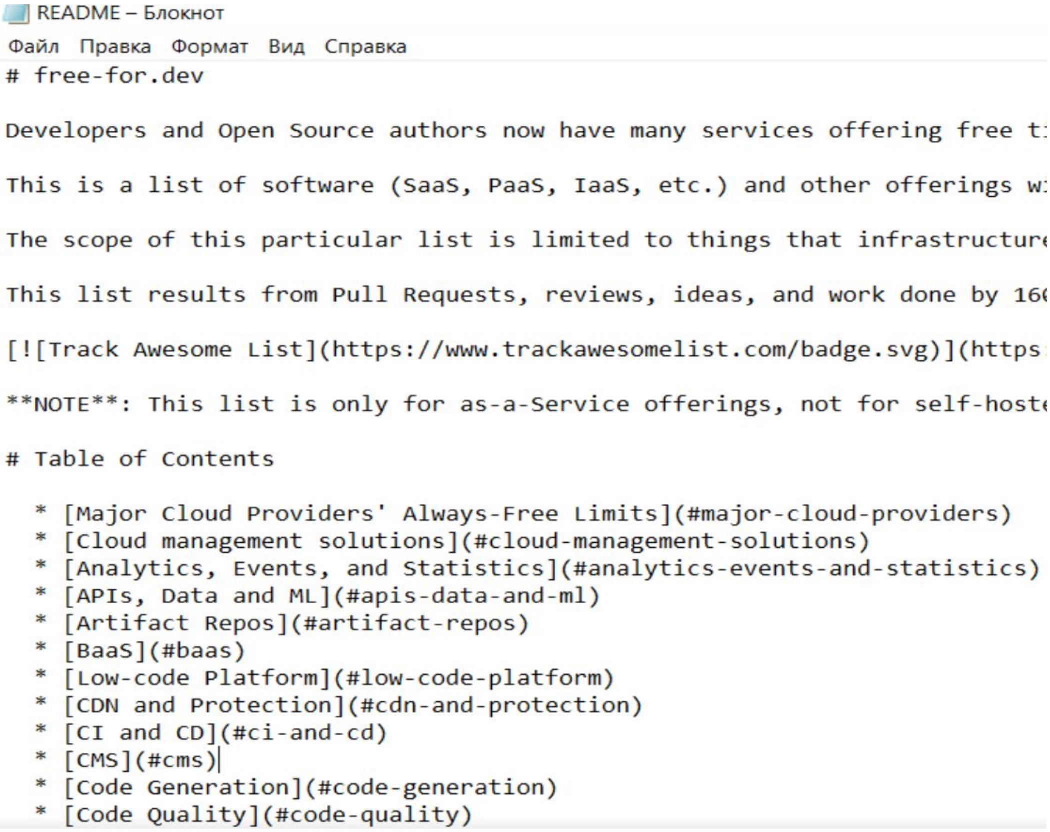
free-for-dev перемещена на рабочий стол:



```
Albina@DESKTOP-1J63AM4 MINGW64 ~/free-for-dev (branch1)
$ cd ..

Albina@DESKTOP-1J63AM4 MINGW64 ~
$ mv free-for-dev ~/Desktop/

Albina@DESKTOP-1J63AM4 MINGW64 ~
$ cd Desktop

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop
$ cd free-for-dev

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev (branch1)
$ ls
CNAME  README.md  index.html  logo.webp
```

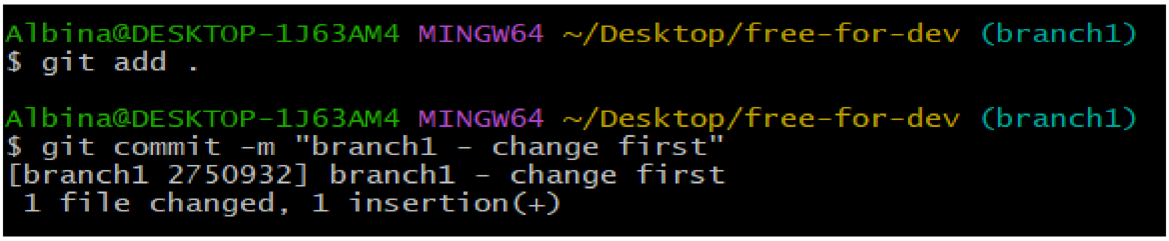
4. Проведите по 3 коммита в каждую из веток, которые меняют один и тот же кусочек файла

Файл README.md до изменений:

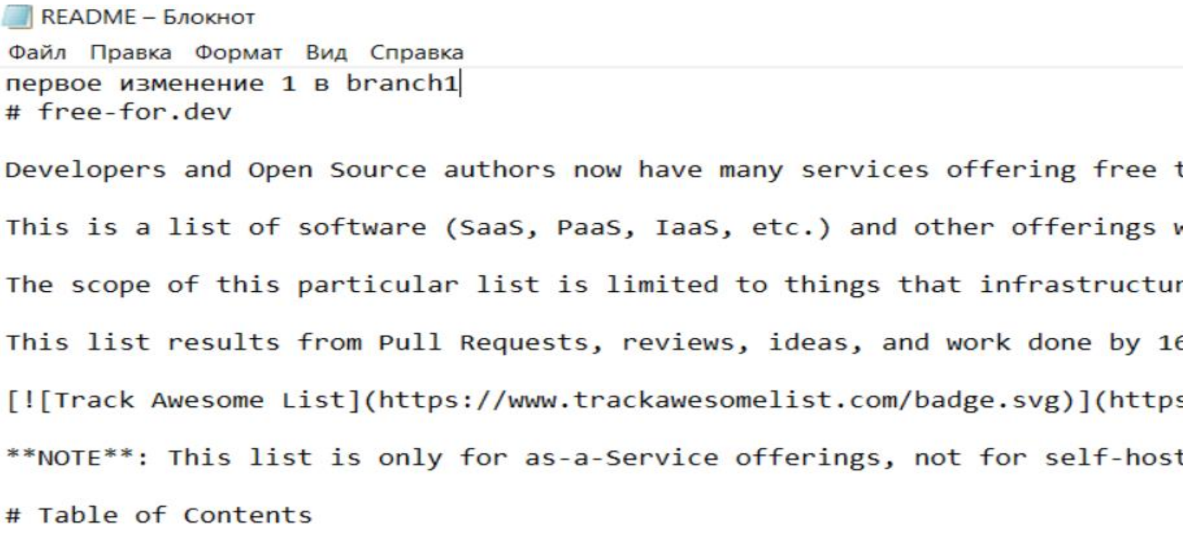


```
# free-for.dev

Developers and Open Source authors now have many services offering free t

This is a list of software (SaaS, PaaS, IaaS, etc.) and other offerings w

The scope of this particular list is limited to things that infrastructur

This list results from Pull Requests, reviews, ideas, and work done by 16

[![Track Awesome List](https://www.trackawesomelist.com/badge.svg)](https

**NOTE**: This list is only for as-a-Service offerings, not for self-host

# Table of Contents

  * [Major Cloud Providers' Always-Free Limits](#major-cloud-providers)
  * [Cloud management solutions](#cloud-management-solutions)
  * [Analytics, Events, and Statistics](#analytics-events-and-statistics)
  * [APIs, Data and ML](#apis-data-and-ml)
  * [Artifact Repos](#artifact-repos)
  * [BaaS](#baas)
  * [Low-code Platform](#low-code-platform)
  * [CDN and Protection](#cdn-and-protection)
  * [CI and CD](#ci-and-cd)
  * [CMS](#cms)
  * [Code Generation](#code-generation)
  * [Code Quality](#code-quality)
```

Первое изменение, первый коммит, branch1:

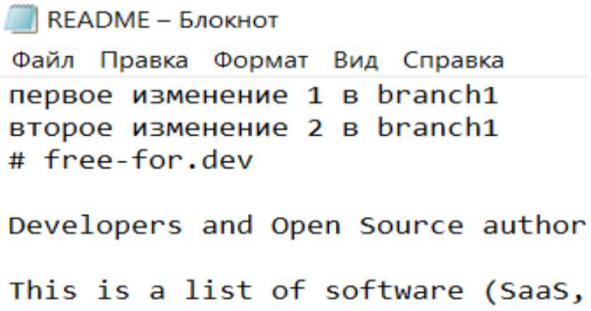


```
Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev (branch1)
$ git add .

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev (branch1)
$ git commit -m "branch1 - change first"
[branch1 2750932] branch1 - change first
 1 file changed, 1 insertion(+)
```

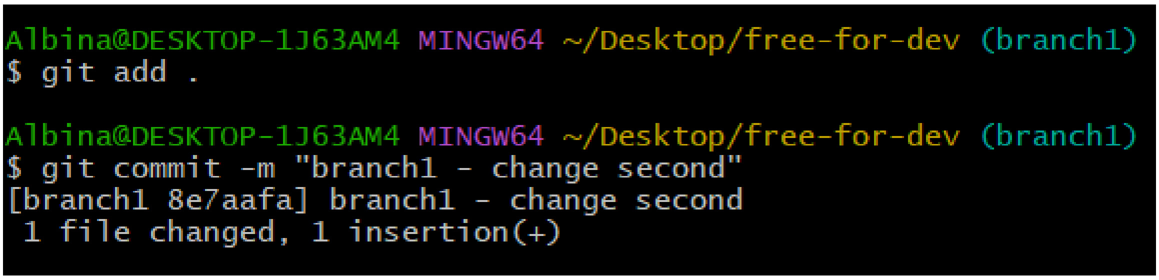


```
первое изменение 1 в branch1
# free-for.dev

Developers and Open Source authors now have many services offering free t

This is a list of software (SaaS, PaaS, IaaS, etc.) and other offerings w

The scope of this particular list is limited to things that infrastructur

This list results from Pull Requests, reviews, ideas, and work done by 16

[![Track Awesome List](https://www.trackawesomelist.com/badge.svg)](https

**NOTE**: This list is only for as-a-Service offerings, not for self-host

# Table of Contents
```

Второе изменение, второй коммит, branch1:

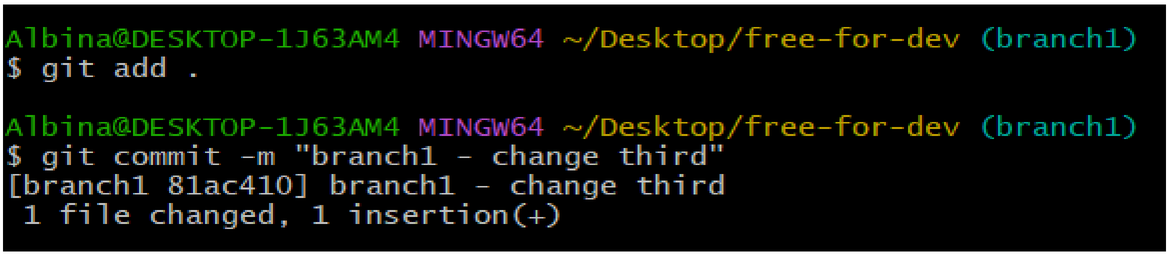


```
первое изменение 1 в branch1
второе изменение 2 в branch1
# free-for.dev

Developers and Open Source author

This is a list of software (SaaS,
```

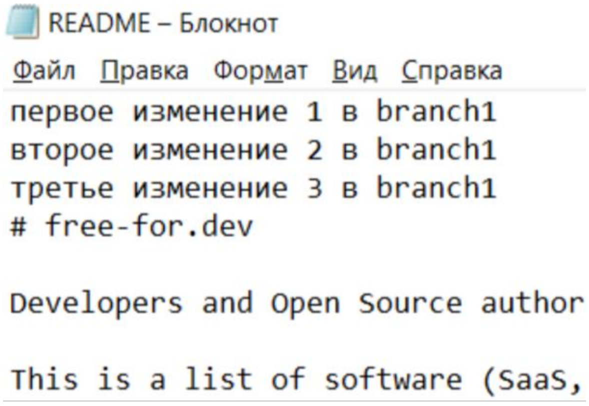


```
Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev (branch1)
$ git add .

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev (branch1)
$ git commit -m "branch1 - change second"
[branch1 8e7aafa] branch1 - change second
 1 file changed, 1 insertion(+)
```

Третье изменение, третий коммит, branch1:

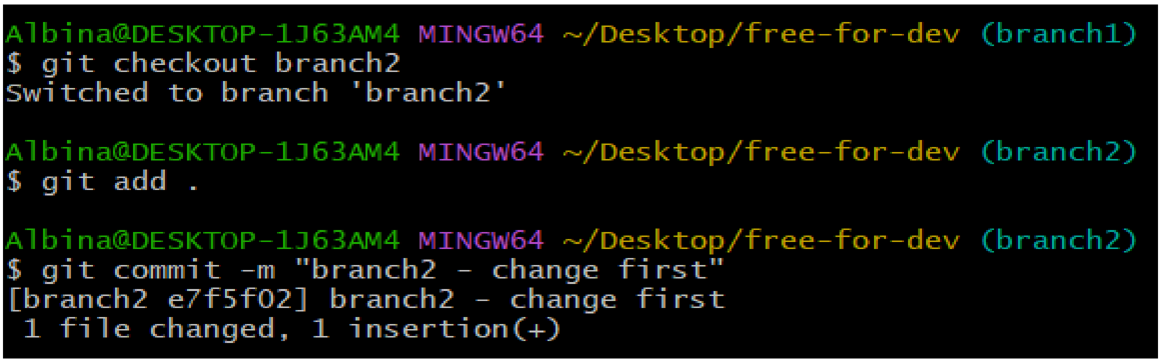


```
Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev (branch1)
$ git add .

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev (branch1)
$ git commit -m "branch1 - change third"
[branch1 81ac410] branch1 - change third
 1 file changed, 1 insertion(+)
```

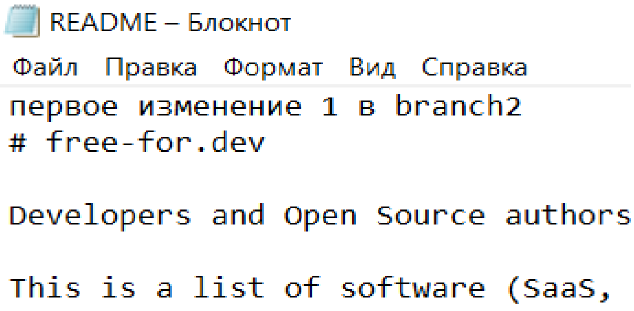


```
первое изменение 1 в branch1
второе изменение 2 в branch1
третье изменение 3 в branch1
# free-for.dev

Developers and Open Source author

This is a list of software (SaaS,
```

Первое изменение, первый коммит, branch2:

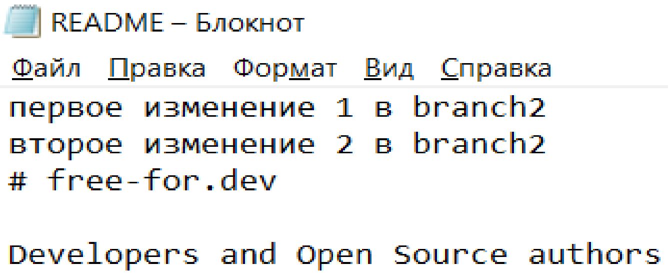


```
Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev (branch1)
$ git checkout branch2
Switched to branch 'branch2'

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev (branch2)
$ git add .

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev (branch2)
$ git commit -m "branch2 - change first"
[branch2 e7f5f02] branch2 - change first
 1 file changed, 1 insertion(+)
```

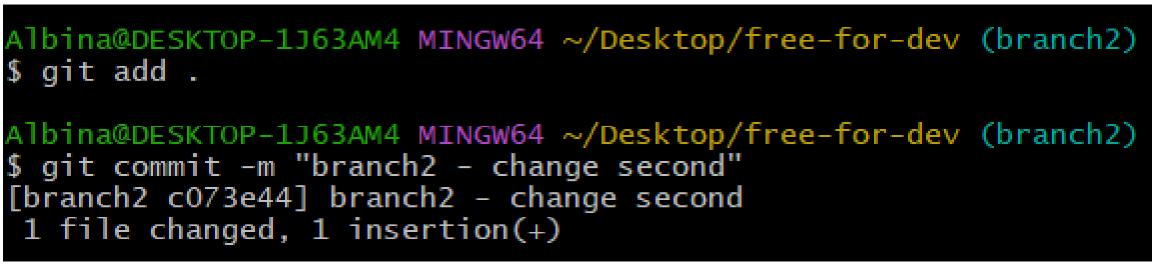


```
первое изменение 1 в branch2
# free-for.dev

Developers and Open Source authors

This is a list of software (SaaS,
```

Второе изменение, второй коммит, branch2:

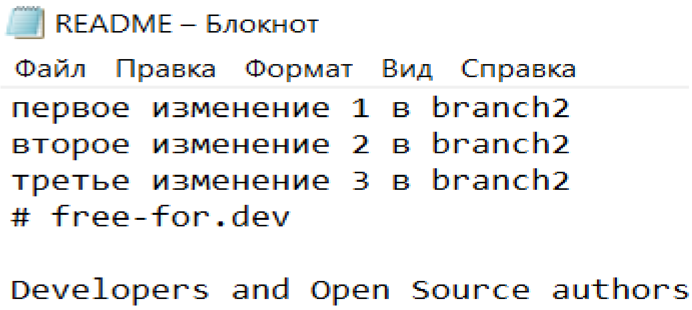


```
первое изменение 1 в branch2
второе изменение 2 в branch2
# free-for.dev

Developers and Open Source authors
```

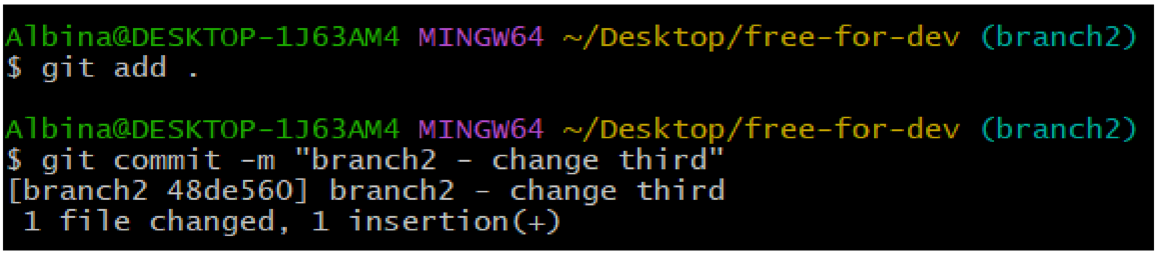


```
Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev (branch2)
$ git add .

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev (branch2)
$ git commit -m "branch2 - change second"
[branch2 c073e44] branch2 - change second
 1 file changed, 1 insertion(+)
```

Третье изменение, третий коммит, branch2:



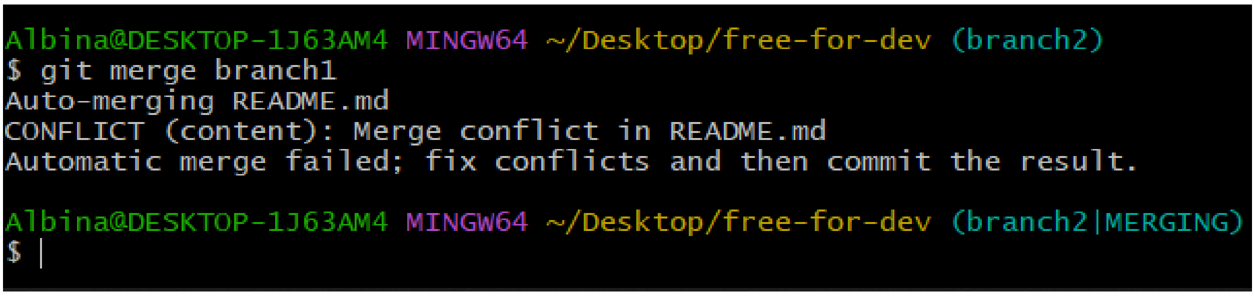
```
первое изменение 1 в branch2
второе изменение 2 в branch2
третье изменение 3 в branch2
# free-for.dev

Developers and Open Source authors
```



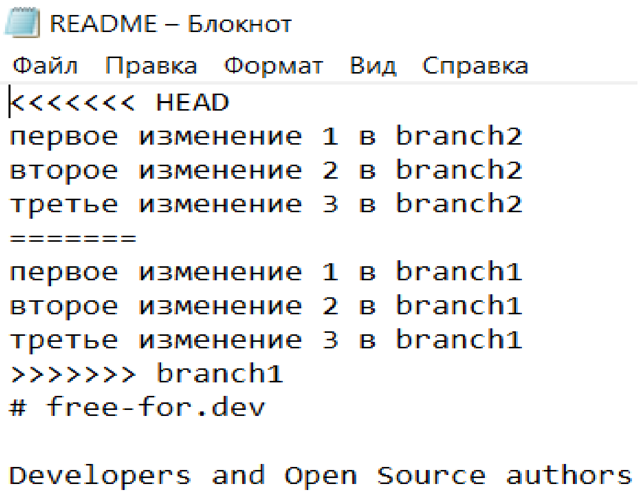
```
Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev (branch2)
$ git add .

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev (branch2)
$ git commit -m "branch2 - change third"
[branch2 48de560] branch2 - change third
 1 file changed, 1 insertion(+)
```

5. Выполните слияние ветки branch1 в ветку branch2, разрешив конфликты при этом

Конфликт:

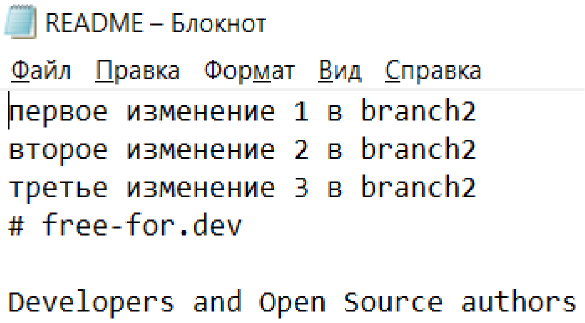


```
Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev (branch2)
$ git merge branch1
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev (branch2|MERGING)
$ |
```

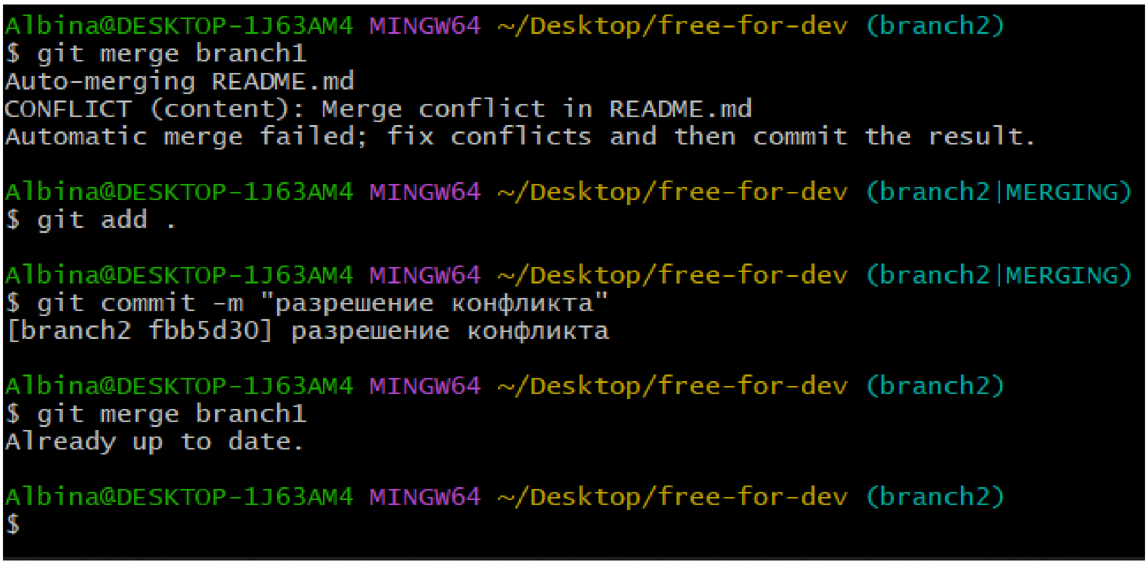


```
<<<<<<< HEAD
первое изменение 1 в branch2
второе изменение 2 в branch2
третье изменение 3 в branch2
=======
первое изменение 1 в branch1
второе изменение 2 в branch1
третье изменение 3 в branch1
>>>>>>> branch1
# free-for.dev

Developers and Open Source authors
```

Разрешение конфликта:

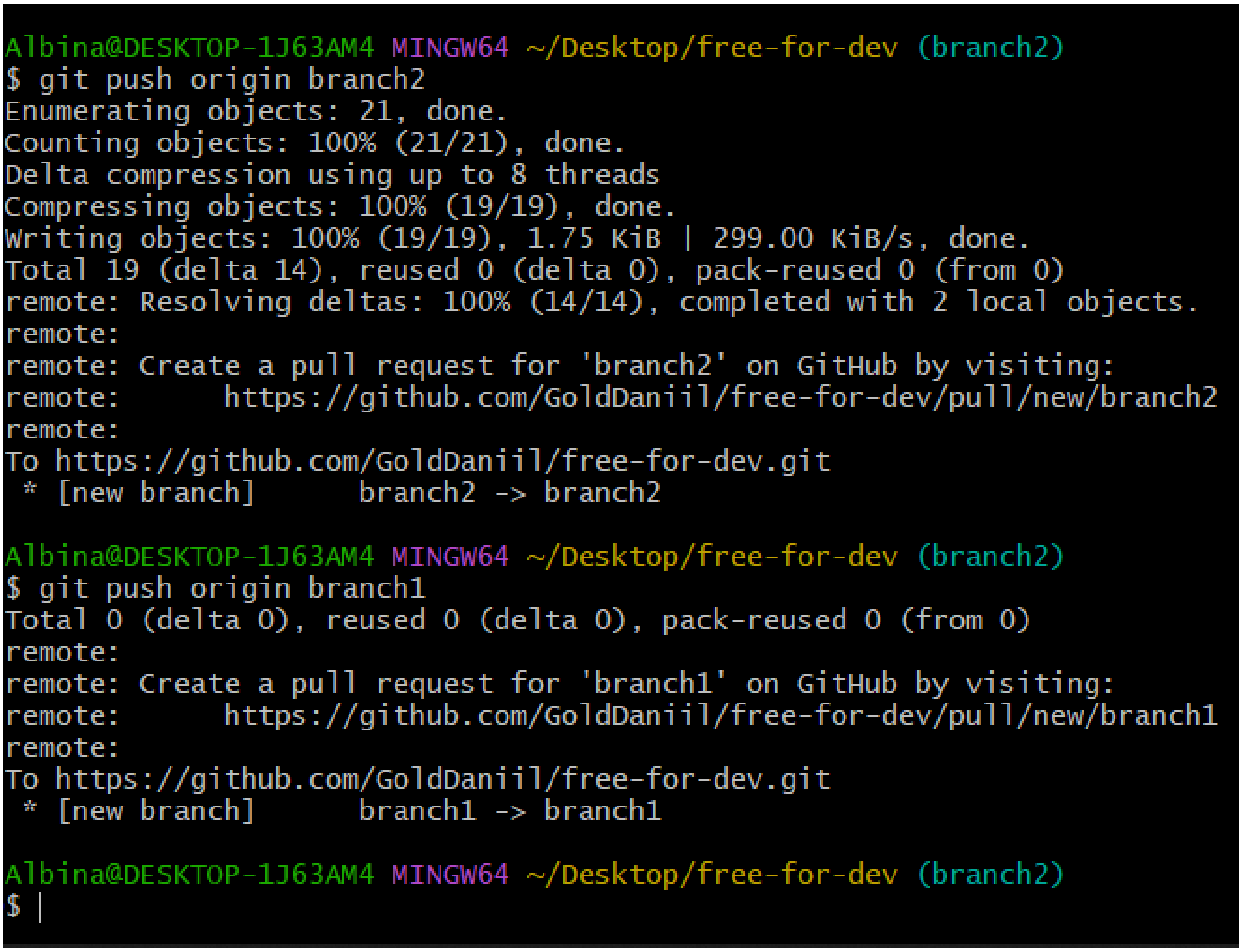


```
первое изменение 1 в branch2
второе изменение 2 в branch2
третье изменение 3 в branch2
# free-for.dev

Developers and Open Source authors
```



```
Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev (branch2)
$ git merge branch1
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev (branch2|MERGING)
$ git add .

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev (branch2|MERGING)
$ git commit -m "разрешение конфликта"
[branch2 fbb5d30] разрешение конфликта

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev (branch2)
$ git merge branch1
Already up to date.

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev (branch2)
$
```

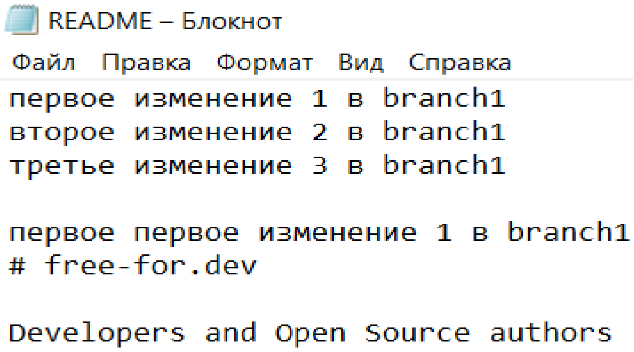
6. Выгрузите все изменения во всех ветках в удаленный репозиторий



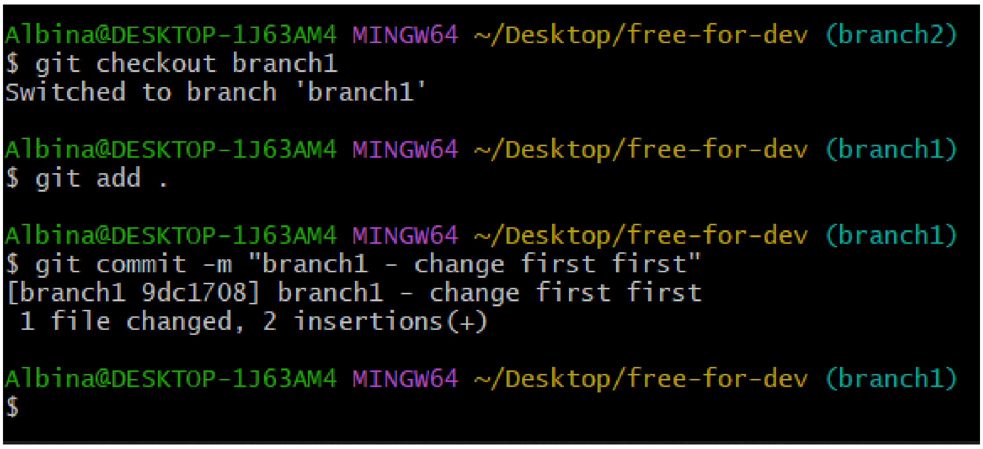
```
Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev (branch2)
$ git push origin branch2
Enumerating objects: 21, done.
Counting objects: 100% (21/21), done.
Delta compression using up to 8 threads
Compressing objects: 100% (19/19), done.
Writing objects: 100% (19/19), 1.75 KiB | 299.00 KiB/s, done.
Total 19 (delta 14), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (14/14), completed with 2 local objects.
remote:
remote: Create a pull request for 'branch2' on GitHub by visiting:
remote:      https://github.com/GoldDaniil/free-for-dev/pull/new/branch2
remote:
To https://github.com/GoldDaniil/free-for-dev.git
 * [new branch]      branch2 -> branch2

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev (branch2)
$ git push origin branch1
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'branch1' on GitHub by visiting:
remote:      https://github.com/GoldDaniil/free-for-dev/pull/new/branch1
remote:
To https://github.com/GoldDaniil/free-for-dev.git
 * [new branch]      branch1 -> branch1

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev (branch2)
$ |
```

7. Проведите еще 3 коммита в ветку branch1

Первое изменение, первый коммит, branch1:

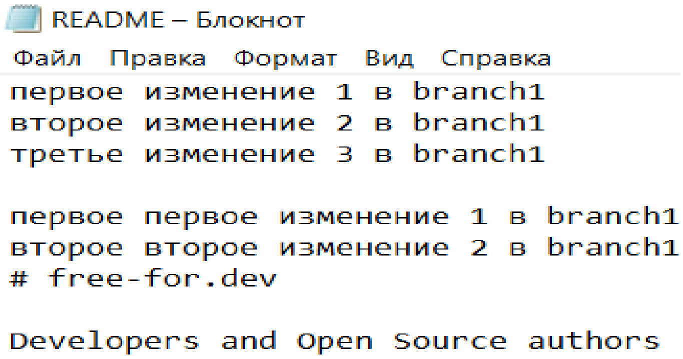


```
первое изменение 1 в branch1
второе изменение 2 в branch1
третье изменение 3 в branch1

первое первое изменение 1 в branch1
# free-for.dev

Developers and Open Source authors
```

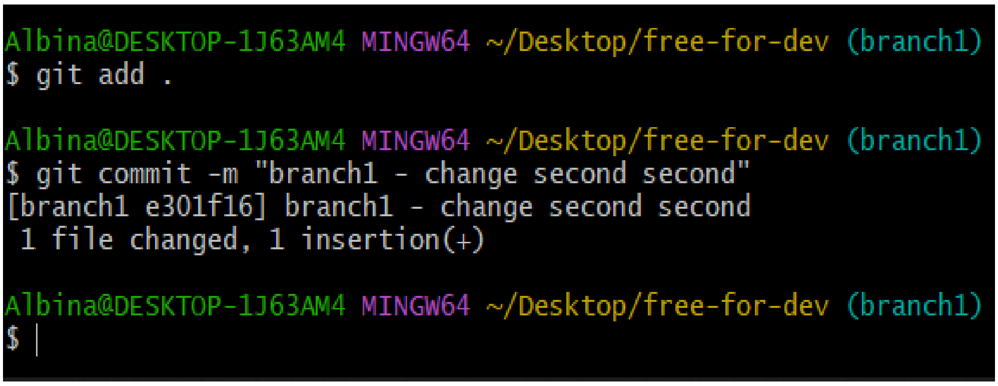


```
Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev (branch2)
$ git checkout branch1
Switched to branch 'branch1'

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev (branch1)
$ git add .

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev (branch1)
$ git commit -m "branch1 - change first first"
[branch1 9dc1708] branch1 - change first first
 1 file changed, 2 insertions(+)

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev (branch1)
$
```

Второй коммит, второе изменение, branch1:

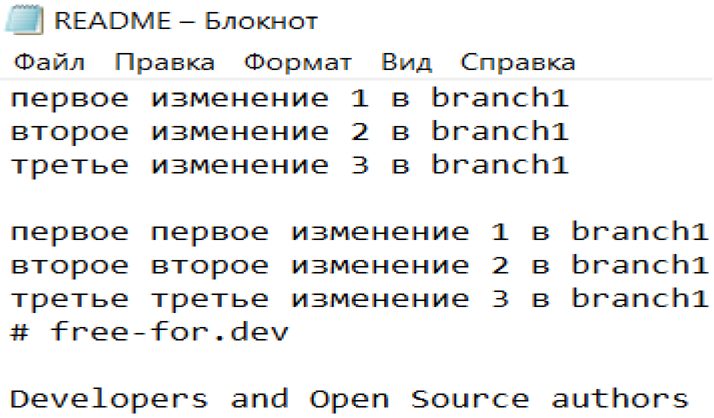


```
первое изменение 1 в branch1
второе изменение 2 в branch1
третье изменение 3 в branch1

первое первое изменение 1 в branch1
второе второе изменение 2 в branch1
# free-for.dev

Developers and Open Source authors
```

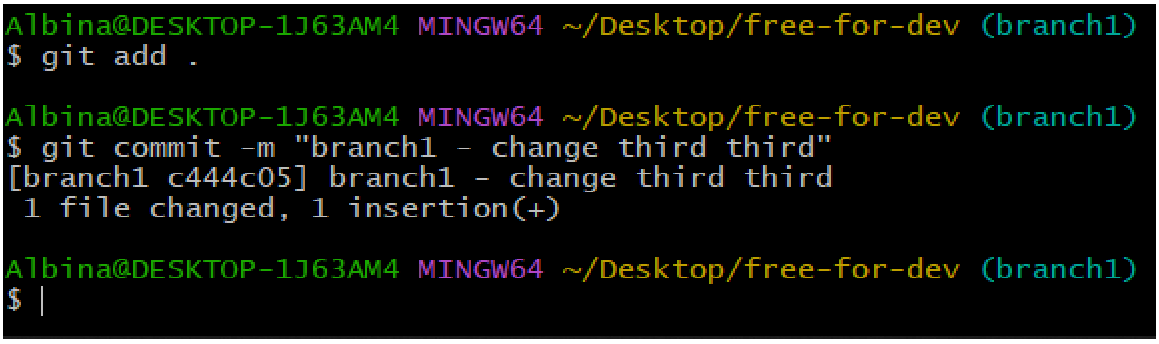


```
Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev (branch1)
$ git add .

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev (branch1)
$ git commit -m "branch1 - change second second"
[branch1 e301f16] branch1 - change second second
 1 file changed, 1 insertion(+)

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev (branch1)
$ |
```

Третий коммит, третье изменение, branch1:

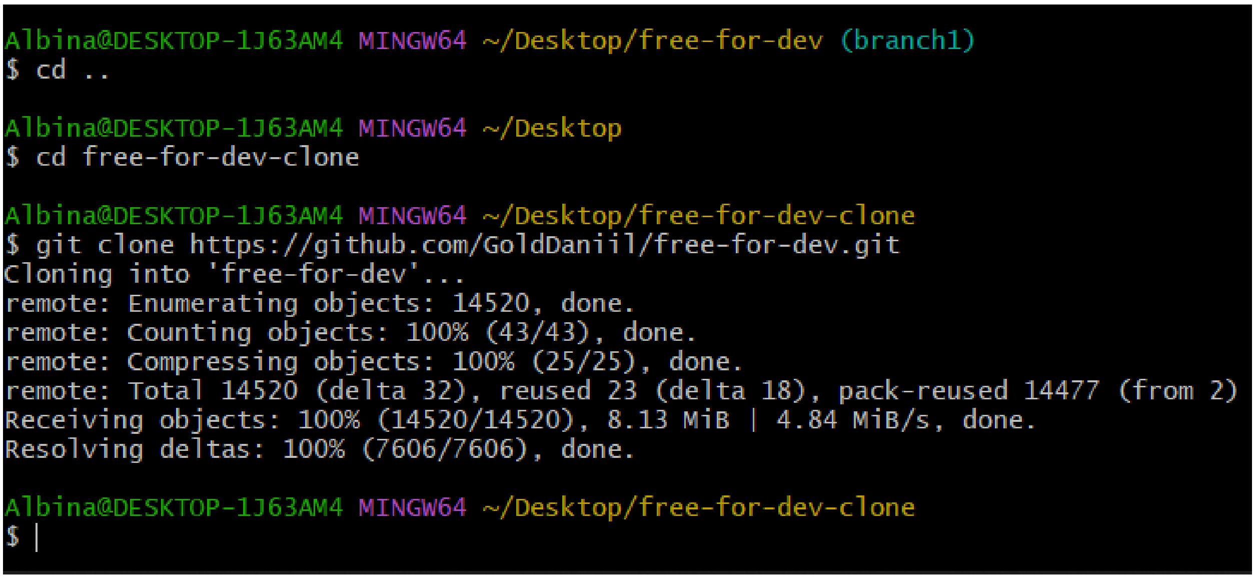


```
первое изменение 1 в branch1
второе изменение 2 в branch1
третье изменение 3 в branch1

первое первое изменение 1 в branch1
второе второе изменение 2 в branch1
третье третье изменение 3 в branch1
# free-for.dev

Developers and Open Source authors
```



```
Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev (branch1)
$ git add .

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev (branch1)
$ git commit -m "branch1 - change third third"
[branch1 c444c05] branch1 - change third third
 1 file changed, 1 insertion(+)

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev (branch1)
$ |
```

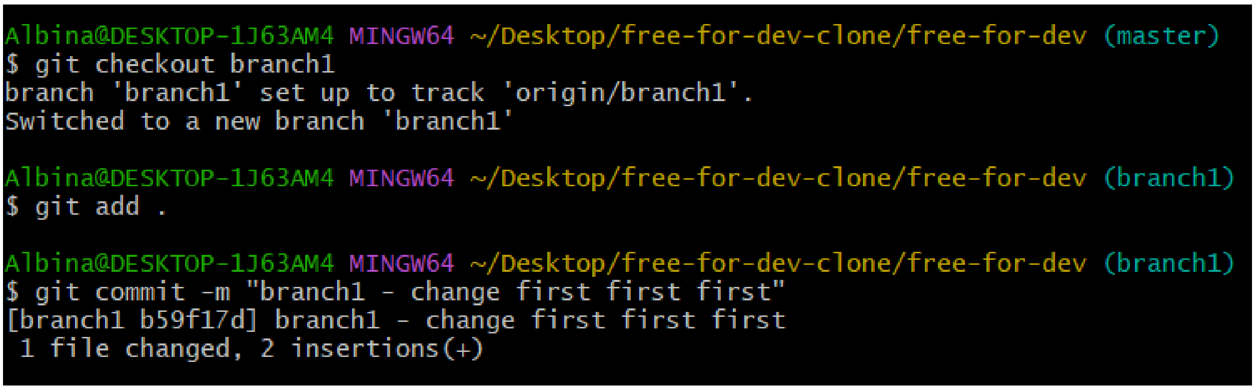
8. Склонируйте репозиторий еще раз в другую директорию



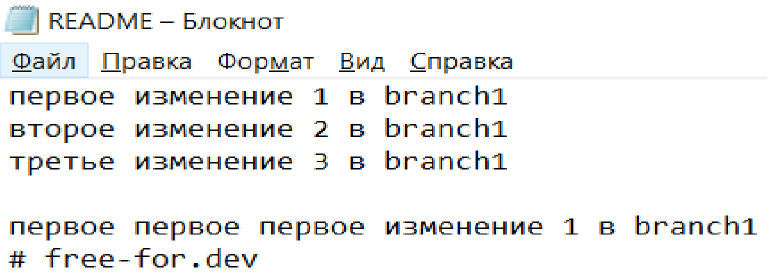
```
Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev (branch1)
$ cd ..

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop
$ cd free-for-dev-clone

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev-clone
$ git clone https://github.com/GoldDaniil/free-for-dev.git
Cloning into 'free-for-dev'...
remote: Enumerating objects: 14520, done.
remote: Counting objects: 100% (43/43), done.
remote: Compressing objects: 100% (25/25), done.
remote: Total 14520 (delta 32), reused 23 (delta 18), pack-reused 14477 (from 2)
Receiving objects: 100% (14520/14520), 8.13 MiB | 4.84 MiB/s, done.
Resolving deltas: 100% (7606/7606), done.

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev-clone
$ |
```

9. В новом клоне репозитории сделайте 3 коммита в ветку branch1

Первый коммит, первое изменение:

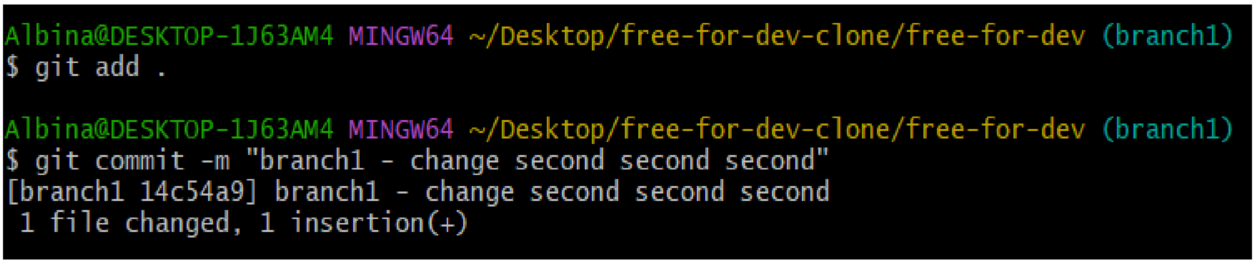


```
Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev-clone/free-for-dev (master)
$ git checkout branch1
branch 'branch1' set up to track 'origin/branch1'.
Switched to a new branch 'branch1'

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev-clone/free-for-dev (branch1)
$ git add .

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev-clone/free-for-dev (branch1)
$ git commit -m "branch1 - change first first first"
[branch1 b59f17d] branch1 - change first first first
 1 file changed, 2 insertions(+)
```

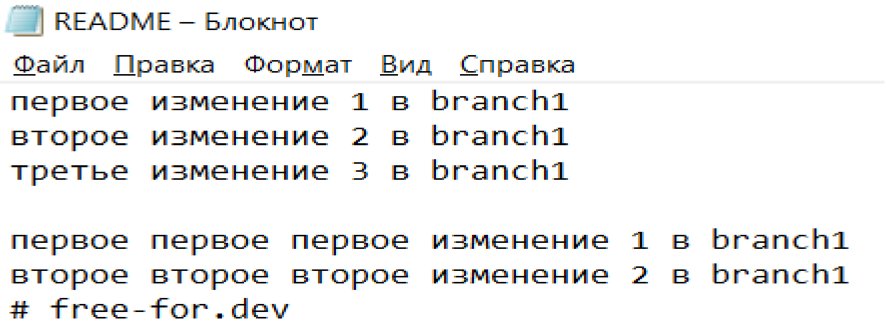


```
первое изменение 1 в branch1
второе изменение 2 в branch1
третье изменение 3 в branch1

первое первое первое изменение 1 в branch1
# free-for.dev
```

Второй коммит, второе изменение:

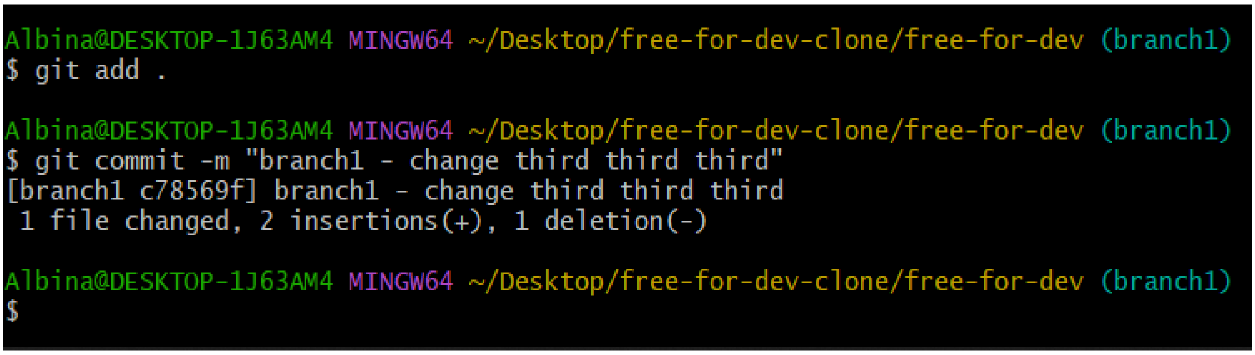


```
Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev-clone/free-for-dev (branch1)
$ git add .

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev-clone/free-for-dev (branch1)
$ git commit -m "branch1 - change second second second"
[branch1 14c54a9] branch1 - change second second second
 1 file changed, 1 insertion(+)
```

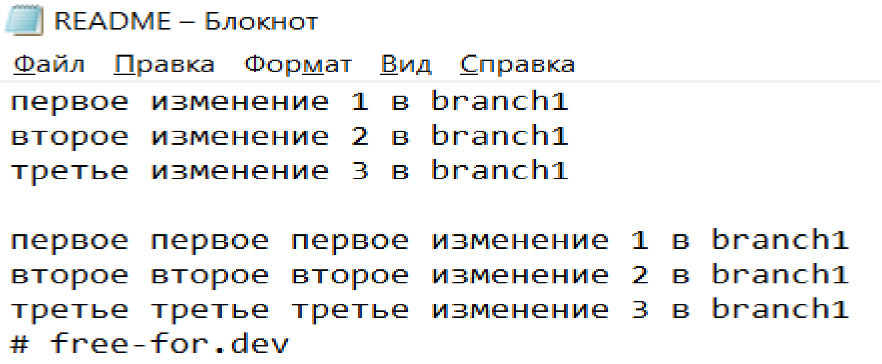


```
первое изменение 1 в branch1
второе изменение 2 в branch1
третье изменение 3 в branch1

первое первое первое изменение 1 в branch1
второе второе второе изменение 2 в branch1
# free-for.dev
```

Третий коммит, третье изменение:

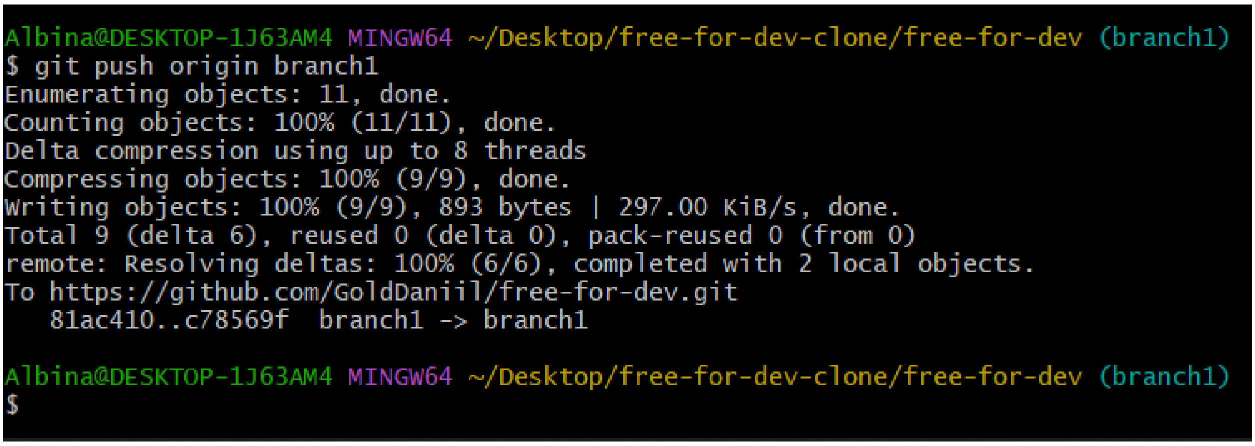


```
Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev-clone/free-for-dev (branch1)
$ git add .

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev-clone/free-for-dev (branch1)
$ git commit -m "branch1 - change third third third"
[branch1 c78569f] branch1 - change third third third
 1 file changed, 2 insertions(+), 1 deletion(-)

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev-clone/free-for-dev (branch1)
$
```



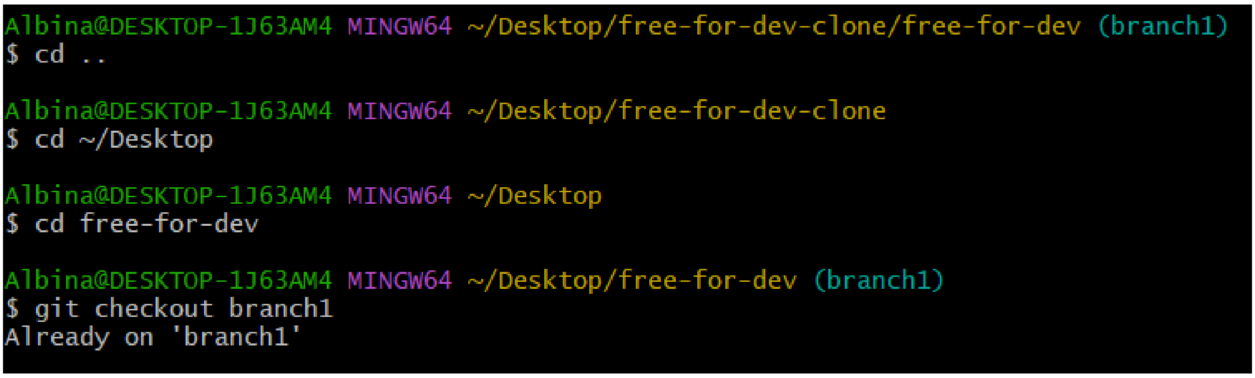
```
первое изменение 1 в branch1
второе изменение 2 в branch1
третье изменение 3 в branch1

первое первое первое изменение 1 в branch1
второе второе второе изменение 2 в branch1
третье третье третье изменение 3 в branch1
# free-for.dev
```

10.Выгрузите все изменения из нового репозитория в удаленный репозиторий

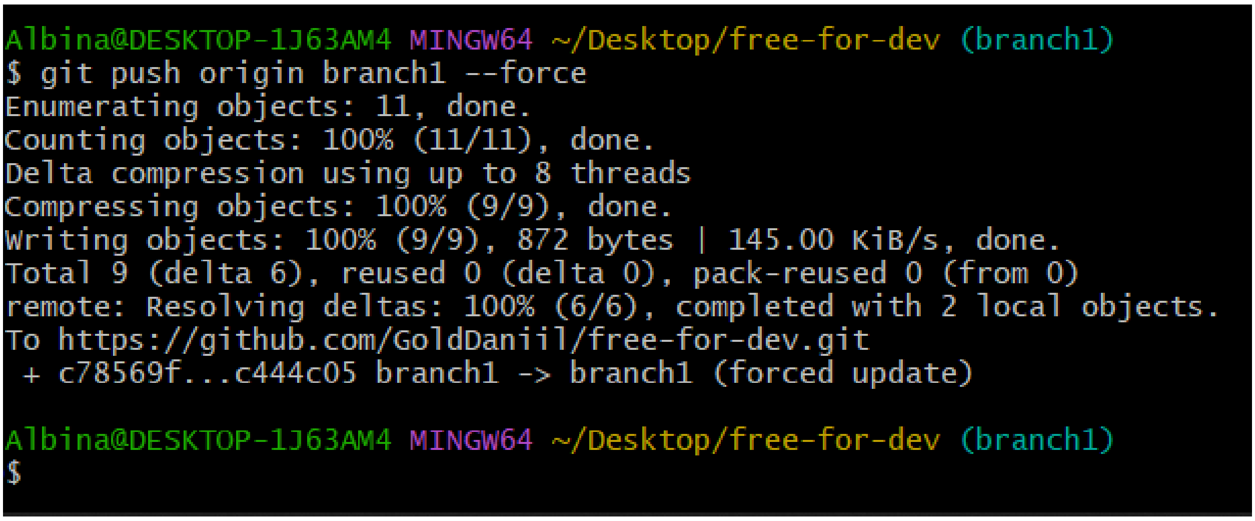


```
Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev-clone/free-for-dev (branch1)
$ git push origin branch1
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 8 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (9/9), 893 bytes | 297.00 KiB/s, done.
Total 9 (delta 6), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (6/6), completed with 2 local objects.
To https://github.com/GoldDaniil/free-for-dev.git
   81ac410..c78569f  branch1 -> branch1

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev-clone/free-for-dev (branch1)
$
```

11. Вернитесь в старый клон с репозиторием, выгрузите изменения с опцией--force

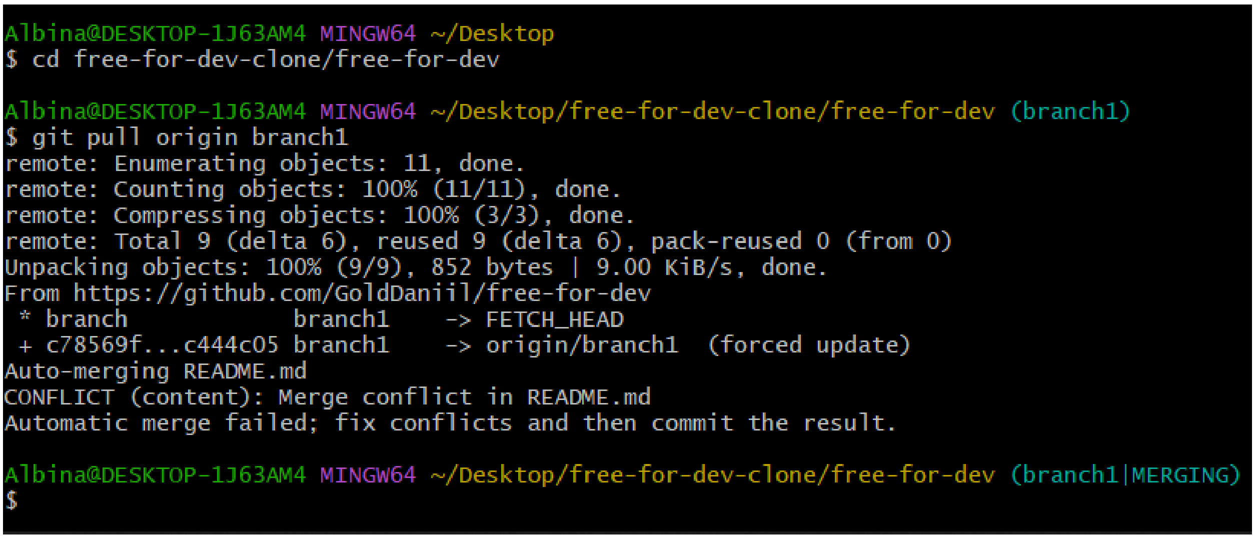


```
Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev-clone/free-for-dev (branch1)
$ cd ..

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev-clone
$ cd ~/Desktop

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop
$ cd free-for-dev

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev (branch1)
$ git checkout branch1
Already on 'branch1'
```



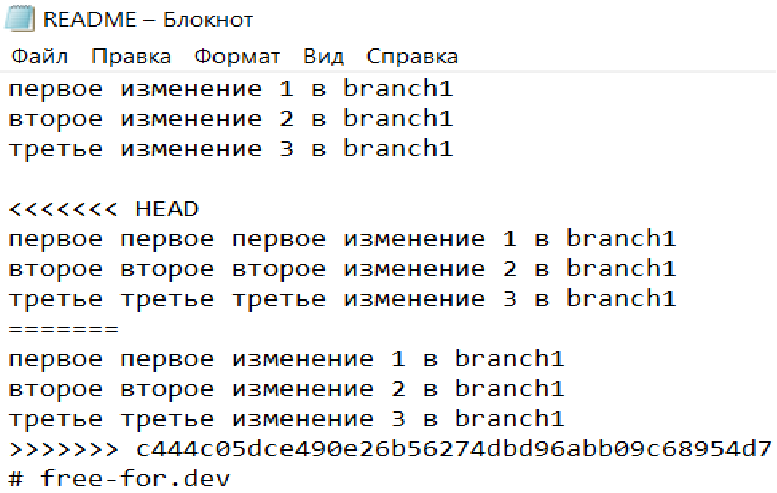
```
Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev (branch1)
$ git push origin branch1 --force
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 8 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (9/9), 872 bytes | 145.00 KiB/s, done.
Total 9 (delta 6), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (6/6), completed with 2 local objects.
To https://github.com/GoldDaniil/free-for-dev.git
 + c78569f...c444c05 branch1 -> branch1 (forced update)

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev (branch1)
$
```

12. Получите все изменения в новом репозитории



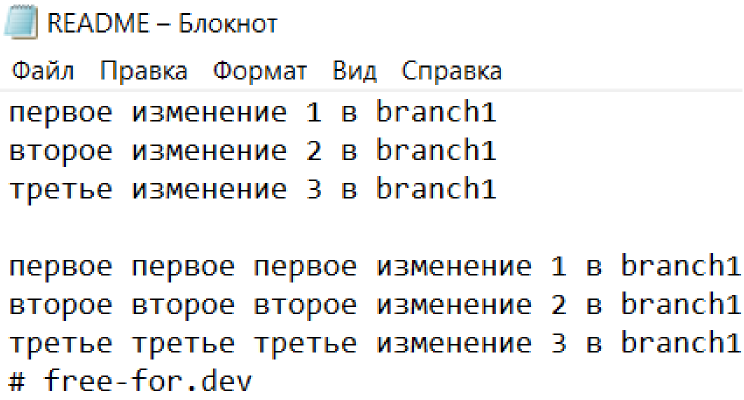
```
Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop
$ cd free-for-dev-clone/free-for-dev

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev-clone/free-for-dev (branch1)
$ git pull origin branch1
remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 9 (delta 6), reused 9 (delta 6), pack-reused 0 (from 0)
Unpacking objects: 100% (9/9), 852 bytes | 9.00 KiB/s, done.
From https://github.com/GoldDaniil/free-for-dev
 * branch            branch1    -> FETCH_HEAD
 + c78569f...c444c05 branch1    -> origin/branch1  (forced update)
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev-clone/free-for-dev (branch1|MERGING)
$
```

Конфликт:

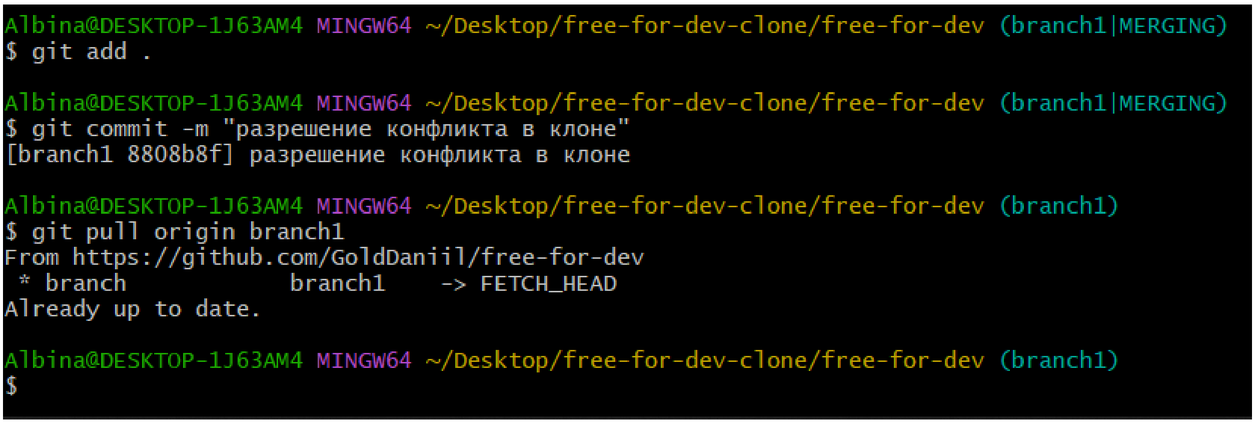


```
первое изменение 1 в branch1
второе изменение 2 в branch1
третье изменение 3 в branch1

<<<<<<< HEAD
первое первое первое изменение 1 в branch1
второе второе второе изменение 2 в branch1
третье третье третье изменение 3 в branch1
=======
первое первое изменение 1 в branch1
второе второе изменение 2 в branch1
третье третье изменение 3 в branch1
>>>>>>> c444c05dce490e26b56274dbd96abb09c68954d7
# free-for.dev
```

Разрешение конфликта:



```
первое изменение 1 в branch1
второе изменение 2 в branch1
третье изменение 3 в branch1

первое первое первое изменение 1 в branch1
второе второе второе изменение 2 в branch1
третье третье третье изменение 3 в branch1
# free-for.dev
```



```
Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev-clone/free-for-dev (branch1|MERGING)
$ git add .

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev-clone/free-for-dev (branch1|MERGING)
$ git commit -m "разрешение конфликта в клоне"
[branch1 8808b8f] разрешение конфликта в клоне

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev-clone/free-for-dev (branch1)
$ git pull origin branch1
From https://github.com/GoldDaniil/free-for-dev
 * branch            branch1    -> FETCH_HEAD
Already up to date.

Albina@DESKTOP-1J63AM4 MINGW64 ~/Desktop/free-for-dev-clone/free-for-dev (branch1)
$
```

КОНТРОЛЬНЫЕ ВОПРОСЫ

2. К какому типу систем контроля версий относится Git?

Git относится к распределительной системе контроля версий, в таких системах разработчик имеет локальную копию репозитория со всеми комитами

8. Что такое конфликт в Git? Как его решить и почему они бывают?

Конфликт это состояние репозитория, когда изменения в двух ветках затрагивают одни и те же строки в файле, и Git не может автоматически определить, какое изменение должно быть применено. Конфликт можно решить вручную, отредактировав файл и выбрав нужный вариант.

9. Как отменить слияние веток, если произошел конфликт?

Чтобы отменить слияние веток нужно воспользоваться командой git merge –abort, это вернёт всё в состояния до начала слияния.

11. Что делает команда git status?

Команда git status показывает текущее состояние рабочего каталога и индекса, это помогает понять какие файлы были изменены и какие файлы нужно проиндексировать.

15. Что делает команда git show?

Показывает информацию об объекте(коммите, дереве или теге) для которого мы вызываем эту команду, выводит изменения введённые в последнем коммите и его автора, дату.

19. Чем отличаются команды "git push" и "git pull"?

git push отправляет изменения локального репозитория в удалённый, а git pull наоборот, забирает изменения из удалённого репозитория и объединяет с локальной веткой.

ВЫВОД

В ходе выполнения данной практической работы были изучены основные принципы работы с системой контроля версий Git, включая базовые команды, управление локальными и удалёнными репозиториями, работу с ветками и разрешение конфликтов.

Были рассмотрены такие важные аспекты, как создание и настройка локального репозитория, выполнение коммитов, откат изменений, работа с тегами, а также взаимодействие с удалёнными репозиториями на GitHub. Отдельное внимание было уделено управлению ветвлением, слиянию изменений и разрешению возможных конфликтов.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. GIT: Создание прочной основы для эффективной разработки. Ч. 1. : учебное пособие / Д. В. Жматов .— Москва : РТУ МИРЭА , 2024

2. Турнецкая Е. Л., Аграновский А. В.Программная инженерия. Интеграционный подход к разработке» (Турнецкая, Е. Л. Программная инженерия. Интеграционный подход к разработке / Е. Л. Турнецкая, А. В. Аграновский. — Санкт-Петербург : Лань, 2023.