

1.Создание локального репозитория в текущей папке.

```
Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС
$ cd Lab3

Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3
$ mkdir work

Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3
$ cd work

Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work
$ git init
Initialized empty Git repository in D:/Лабы/ПООИС/Lab3/work/.git/

Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (main)
$
```

2.Просмотреть статус текущего репозитория.

```
Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (main)
$ git status
On branch main

No commits yet

nothing to commit (create/copy files and use "git add" to track)
```

3. Что такое ветка и какая ветка является обычно основной.

Ветка — это параллельная версия репозитория. Она позволяет вам работать над новой функцией вашего проекта, не затрагивая основную версию проекта. После того, как вы закончите работу над новой функцией, вы можете объединить ветку обратно с основной версией проекта.

4.Добавить файл в контекст, который будет коммититься.

```
Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (main)
$ touch hello.html
```

5.Создать коммит на основе текущего контекста и указать для него комментарий.

```
Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (main)
$ git add hello.html

Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (main)
$ git commit -m "created hello.html"
[main (root-commit) 9ba9d2a] created hello.html
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 hello.html
```

6.Создать коммит, включающий изменения всех наблюдаемых файлов и указать для него комментарий.

```

Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (main)
$ echo "Hello, World!" >> hello.txt

Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (main)
$ echo "Hello, World!" >> hello1.txt

Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (main)
$ git add .
warning: in the working copy of 'hello.txt', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'hello1.txt', LF will be replaced by CRLF the next time Git touches it

Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (main)
$ git commit -m -a "Committed all files"
error: pathspec 'Committed all files' did not match any file(s) known to git

Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (main)
$ git commit -m "Committed all files"
[main 6bfa7a3] Committed all files
 2 files changed, 2 insertions(+)

```

7.Просмотреть протокол(лог) коммитов.

```

Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (main)
$ git log
commit 6bfa7a359011654b82831a2bfcf59b9d50d72520 (HEAD -> main)
Author: MaksimDedkov <maksimdedkov2006@gmail.com>
Date: Tue Oct 22 18:17:09 2024 +0300

    Committed all files

commit eeb3d4bbf933fbbda722d847d4a269064609bd28
Author: MaksimDedkov <maksimdedkov2006@gmail.com>
Date: Tue Oct 22 18:14:21 2024 +0300

    Committed all files

commit 9ba9d2a45134a09627ab48e74bf612f42cbe8444
Author: MaksimDedkov <maksimdedkov2006@gmail.com>
Date: Tue Oct 22 18:12:46 2024 +0300

    created hello.html

```

8.Посмотреть информацию о текущих настройках.

```

Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (main)
$ git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/etc/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
init.defaultbranch=main
core.autocrlf=true
core.safecrlf=warn
user.name=MaksimDedkov
user.email=maksimdedkov2006@gmail.com
core.autocrlf=true
core.safecrlf=warn
core.repositoryformatversion=0
core.filemode=false
core.bare=false
core.logallrefupdates=true
core.symlinks=false
core.ignorecase=true

```

9. Убрать файл из контекста.

```
Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (main)
$ git reset hello.txt
```

10. Посмотреть изменения в файле по сравнению с последним коммитом.

```
Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (main)
$ git show HEAD
commit 6bfa7a359011654b82831a2bfcf59b9d50d72520 (HEAD -> main)
Author: MaksimDedkov <maksimdedkov2006@gmail.com>
Date: Tue Oct 22 18:17:09 2024 +0300

    Committed all files

diff --git a/hello.txt b/hello.txt
index e69de29..8ab686e 100644
--- a/hello.txt
+++ b/hello.txt
@@ -0,0 +1 @@
+Hello, World!
diff --git a/hello1.txt b/hello1.txt
index e69de29..8ab686e 100644
--- a/hello1.txt
+++ b/hello1.txt
@@ -0,0 +1 @@
+Hello, World!
```

11. Убрать изменения относительно последнего коммита из файла.

```
Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (main)
$ git log --pretty=format:"%h %ad | %s%d [%an]" --date=short
6bfa7a3 2024-10-22 | Committed all files (HEAD -> main) [MaksimDedkov]
eeb3d4b 2024-10-22 | Committed all files [MaksimDedkov]
9ba9d2a 2024-10-22 | created hello.html [MaksimDedkov]

Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (main)
$ git reset --hard HEAD
HEAD is now at 6bfa7a3 Committed all files
```

12. Добавить в контекст коммита все измененные и созданные файлы

```
Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (main)
$ touch hell.txt

Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (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.txt

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

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

Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (main)
$ git add .
warning: in the working copy of 'hello.txt', LF will be replaced by CRLF the next time Git touches it

Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (main)
$ git commit -m "Created hell.txt and edited hello.txt"
[main 41f3889] Created hell.txt and edited hello.txt
 2 files changed, 1 insertion(+)
 create mode 100644 hell.txt
```

13. изменить глобальные/локальные настройки

```
Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (main)
$ git config user.name MaksimD

Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (main)
$ git config user.name
MaksimD
```

```
Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (main)
$ git config --global user.email maksimdedkov2006@gmail.com

Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (main)
$ git config --global user.email
maksimdedkov2006@gmail.com
```

14. переписать имя пользователя

```
Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (main)
$ git config user.name MaksimD

Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (main)
$ git config user.name
MaksimD
```

15. просмотреть существующие ветки

```
Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (main)
$ git branch
* main
```

16. создать новую ветку

```
Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (main)
$ git branch branch1
```

17. переключиться на другую ветку

```
Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (main)
$ git switch branch1
D      hello.txt
D      hello1.txt
Switched to branch 'branch1'
```

18. создать новую ветку и сразу же переключиться на неё

```
Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (branch1)
$ git switch -c branch2
Switched to a new branch 'branch2'
```

19. удалить ветку/удалить ветку, даже если она не примержена

```
Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (branch1)
$ git switch -c branch2
Switched to a new branch 'branch2'
```

20. примержить изменения из указанной ветки в текущую

```
Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (branch2)
$ git branch -d branch1
Deleted branch branch1 (was 41f3889).
```

21. в каком случае могут появиться конфликты? сделать конфликт

Ошибка может возникнуть, если в двух ветках репозитория была изменена одна и та же часть файла.

```

Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (main)
$ echo "text" >> hello.html

Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (main)
$ git add hello.html
warning: in the working copy of 'hello.html', LF will be replaced by CRLF the next time Git touches it

Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (main)
$ git commit -m "adding conflict"
[main 3fe9c54] adding conflict
1 file changed, 1 insertion(+)

Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (main)
$ git switch branch1
Switched to branch 'branch1'

Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (branch1)
$ echo "conflict" >> hello.html

Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (branch1)
$ git add hello.html
warning: in the working copy of 'hello.html', LF will be replaced by CRLF the next time Git touches it

Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (branch1)
$ git commit -m "conflict"
[branch1 af2e637] conflict
1 file changed, 1 insertion(+)

Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (branch1)
$ git merge main
Auto-merging hello.html
CONFLICT (content): Merge conflict in hello.html
Automatic merge failed; fix conflicts and then commit the result.

```

22. как посмотреть в каких файлах конфликты

```

Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (branch1|MERGING)
$ git status
On branch branch1
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add <file>..." to mark resolution)
    both modified:   hello.html

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

Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (branch1|MERGING)

```

```

hello.html  ▢ ✕
1          <<<<<< HEAD
2          conflict
3          =====
4          text
5          >>>>>> main
6

```

23. как устранить конфликты

```
hello.html  ▸ X
1  ✓  <html>
2    <head>
3    |
4    </head>
5  ✓  <body>
6    |
7    |   conflict
8    |   text
9    </body>
    </html>
```

```
Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (branch1|MERGING)
$ git add hello.html

Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (branch1|MERGING)
$ git commit -m "resolved conflict"
[branch1 cb5504a] resolved conflict

Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (branch1)
$ git status
On branch branch1
nothing to commit, working tree clean

Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (branch1)
$ git log --all --graph --pretty=format:"%h %ad | %s%d [%an]" --date=short
*   cb5504a 2024-10-23 | resolved conflict (HEAD -> branch1) [MaksimDedkov]
| \
|  * 3fe9c54 2024-10-23 | adding conflict (main) [MaksimDedkov]
|  * | af2e637 2024-10-23 | conflict [MaksimDedkov]
|  /
* 2aa39d5 2024-10-23 | added hello.html [MaksimDedkov]
```

24. как переключиться на указанный КОММИТ

```
Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (branch1)
$ git log --all --graph --pretty=format:"%h %ad | %s%d [%an]" --date=short
*   cb5504a 2024-10-23 | resolved conflict (HEAD -> branch1) [MaksimDedkov]
| \
|  * 3fe9c54 2024-10-23 | adding conflict (main) [MaksimDedkov]
|  * | af2e637 2024-10-23 | conflict [MaksimDedkov]
|  /
* 2aa39d5 2024-10-23 | added hello.html [MaksimDedkov]
```

```
Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (branch1)
```

```
$ git checkout 2aa39d5
Note: switching to '2aa39d5'.
```

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 2aa39d5 added hello.html

25. сделать ребазирование(rebase) текущей ветки

```
Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (branch1)
$ git add hello.html

Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (branch1)
$ git commit -m "rebased branch"
On branch branch1
nothing to commit, working tree clean

Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (branch1)
$ git switch main
Switched to branch 'main'

Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (main)
$ git rebase branch1
Successfully rebased and updated refs/heads/main.
```

26. устранение конфликтов во время ребазирования

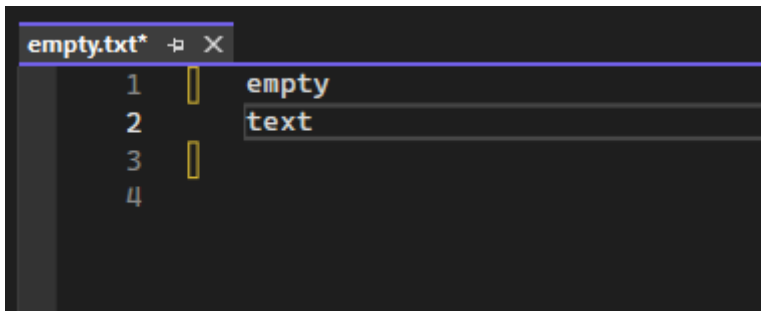
```
Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (branch1)
$ git rebase main
warning: skipped previously applied commit 5da7eb1
hint: use --reapply-cherry-picks to include skipped commits
hint: Disable this message with "git config advice.skippedCherryPicks false"
Auto-merging empty.txt
CONFLICT (content): Merge conflict in empty.txt
error: could not apply d84e10a... conflict
hint: Resolve all conflicts manually, mark them as resolved with
hint: "git add/rm <conflicted_files>", then run "git rebase --continue".
hint: You can instead skip this commit: run "git rebase --skip".
hint: To abort and get back to the state before "git rebase", run "git rebase --abort".
hint: Disable this message with "git config advice.mergeConflict false"
Could not apply d84e10a... conflict

Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (branch1|REBASE 1/1)
$ git status
interactive rebase in progress; onto 6fffb44
Last command done (1 command done):
  pick d84e10a conflict
No commands remaining.
You are currently rebasing branch 'branch1' on '6fffb44'.
  (fix conflicts and then run "git rebase --continue")
  (use "git rebase --skip" to skip this patch)
  (use "git rebase --abort" to check out the original branch)

Unmerged paths:
  (use "git restore --staged <file>..." to unstage)
  (use "git add <file>..." to mark resolution)
    both modified:   empty.txt

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

```
empty.txt  ▢ X
1          <<<<<< HEAD
2          empty
3          =====
4          text
5          >>>>>> d84e10a (conflict)
6
```

```
Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (branch1|REBASE 1/1)
$ git rebase --continue
Successfully rebased and updated refs/heads/branch1.

Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (branch1)
$ git commit -m "rebased txt"
On branch branch1
nothing to commit, working tree clean

Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (branch1)
$ git log --pretty=format:"%h %ad | %s%d [%an]" --date=short
df2671f 2024-10-23 | rebased (HEAD -> branch1) [MaksimDedkov]
6fffb44 2024-10-23 | added conflict (main) [MaksimDedkov]
0b7fb4b 2024-10-23 | added txt [MaksimDedkov]
c5c9560 2024-10-23 | adding conflict [MaksimDedkov]
621962c 2024-10-23 | conflict [MaksimDedkov]
2e02a62 2024-10-23 | adding conflict [MaksimDedkov]
cb5504a 2024-10-23 | resolved conflict [MaksimDedkov]
af2e637 2024-10-23 | conflict [MaksimDedkov]
3fe9c54 2024-10-23 | adding conflict [MaksimDedkov]
2aa39d5 2024-10-23 | added hello.html [MaksimDedkov]
```

27. отменить ребазирование во время конфликтов

```
Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (main)
$ git rebase branch1
Auto-merging hello.html
CONFLICT (content): Merge conflict in hello.html
error: could not apply f84692b... conflict
hint: Resolve all conflicts manually, mark them as resolved with
hint: "git add/rm <conflicted_files>", then run "git rebase --continue".
hint: You can instead skip this commit: run "git rebase --skip".
hint: To abort and get back to the state before "git rebase", run "git rebase --abort".
hint: Disable this message with "git config advice.mergeConflict false"
Could not apply f84692b... conflict

Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (main|REBASE 1/1)
$ git rebase --abort

Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (main)
$ git status
On branch main
nothing to commit, working tree clean
```

28. пропустить текущий конфликтный коммит и перейти к следующему


```

Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (branch1)
$ git rebase main
Auto-merging hello.html
CONFLICT (content): Merge conflict in hello.html
error: could not apply 96e696b... conflict
hint: Resolve all conflicts manually, mark them as resolved with
hint: "git add/rm <conflicted_files>", then run "git rebase --continue".
hint: You can instead skip this commit: run "git rebase --skip".
hint: To abort and get back to the state before "git rebase", run "git rebase --abort".
hint: Disable this message with "git config advice.mergeConflict false"
Could not apply 96e696b... conflict

Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (branch1|REBASE 2/2)
$ git rebase --skip
Successfully rebased and updated refs/heads/branch1.

```

29. отправить изменения из локального репозитория для указанной ветки удалённый(дистанционный) репозиторий

```

Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (main)
$ git remote add origin https://github.com/MaksimDedkov/MDrep.git

Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (main)
$ git push origin main
info: please complete authentication in your browser...
Enumerating objects: 30, done.
Counting objects: 100% (30/30), done.
Delta compression using up to 16 threads
Compressing objects: 100% (21/21), done.
Writing objects: 100% (30/30), 2.55 KiB | 1.28 MiB/s, done.
Total 30 (delta 5), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (5/5), done.
To https://github.com/MaksimDedkov/MDrep.git
 * [new branch]      main -> main

```

30. забрать изменения из репозитория, для которого были созданы удалённые ветки по умолчанию

```

Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (main)
$ git fetch origin

```

31. забрать изменения удалённой ветки из репозитория по умолчанию, основной ветки

```

Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3/work (main)
$ git fetch origin main
From https://github.com/MaksimDedkov/MDrep
 * branch          main      -> FETCH_HEAD

```

32. создание копии репозитория

```

Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3
$ ls
work/

Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3
$ git clone work work1
Cloning into 'work1'...
done.

Максим@КОМПЮТЕР MINGW64 /d/Лабы/ПООИС/Lab3
$ ls
work/  work1/

```

33. переименовать последний коммит

```
Максим@KOMPUTER MINGW64 /d/Лабы/ПООИС/Lab3/work (main)
$ git log --pretty=format:"%h %ad | %s%d [%an]" --date=short
c6ac0bf 2024-10-23 | new conflict (HEAD -> main) [MaksimDedkov]
840c67d 2024-10-23 | conf [MaksimDedkov]
6fffb44 2024-10-23 | added conflict [MaksimDedkov]
0b7fb4b 2024-10-23 | added txt [MaksimDedkov]
c5c9560 2024-10-23 | adding conflict [MaksimDedkov]
621962c 2024-10-23 | conflict [MaksimDedkov]
2e02a62 2024-10-23 | adding conflict [MaksimDedkov]
cb5504a 2024-10-23 | resolved conflict [MaksimDedkov]
af2e637 2024-10-23 | conflict [MaksimDedkov]
3fe9c54 2024-10-23 | adding conflict [MaksimDedkov]
2aa39d5 2024-10-23 | added hello.html [MaksimDedkov]
```

```
Максим@KOMPUTER MINGW64 /d/Лабы/ПООИС/Lab3/work (main)
$ git commit -m "last conflict" --amend
[main e4e110c] last conflict
Date: Wed Oct 23 22:27:02 2024 +0300
1 file changed, 1 insertion(+)
```

```
Максим@KOMPUTER MINGW64 /d/Лабы/ПООИС/Lab3/work (main)
$ git log --pretty=format:"%h %ad | %s%d [%an]" --date=short
e4e110c 2024-10-23 | last conflict (HEAD -> main) [MaksimDedkov]
840c67d 2024-10-23 | conf [MaksimDedkov]
6fffb44 2024-10-23 | added conflict [MaksimDedkov]
0b7fb4b 2024-10-23 | added txt [MaksimDedkov]
c5c9560 2024-10-23 | adding conflict [MaksimDedkov]
621962c 2024-10-23 | conflict [MaksimDedkov]
2e02a62 2024-10-23 | adding conflict [MaksimDedkov]
cb5504a 2024-10-23 | resolved conflict [MaksimDedkov]
af2e637 2024-10-23 | conflict [MaksimDedkov]
3fe9c54 2024-10-23 | adding conflict [MaksimDedkov]
2aa39d5 2024-10-23 | added hello.html [MaksimDedkov]
```

34. переименовать не последний коммит
git rebase -i HEAD~n

35. скрыть изменения по сравнению с последним коммитом

```
Максим@KOMPUTER MINGW64 /d/Лабы/ПООИС/Lab3/work (main)
$ git stash
No local changes to save
```