

6 Вариант: Использовать в работе 3 разных файла, и количество версий каждого файла не должно быть меньше 3.

## Задание 1

Создание проекта и репозитория (Создаём сразу три файла: first.py , second.py , third.py)

```
1111@erroremelya MINGW64 /d/na6 3
$ mkdir eay

1111@erroremelya MINGW64 /d/na6 3
$ cd eay

1111@erroremelya MINGW64 /d/na6 3/eay
$ touch first.py

1111@erroremelya MINGW64 /d/na6 3/eay
$ touch second.py

1111@erroremelya MINGW64 /d/na6 3/eay
$ touch third.py

1111@erroremelya MINGW64 /d/na6 3/eay
$ git init
Initialized empty Git repository in D:/лаб 3/eay/.git/
```

Проверка состояния репозитория

```
1111@erroremelya MINGW64 /d/na6 3/eay (master)
$ git status
on branch master
untracked files:
  (use "git add <file>..." to include in what will be committed)
        second.py
        third.py

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

Индексация изменений

```
1111@erroremelya MINGW64 /d/na6 3/eay (master)
$ git add first.py
```

История версий

```
1111@erroremelya MINGW64 /d/na6 3/eay (master)
$ git log
commit 70a1baabfe82119712c6772ea838f423c953d4c3 (HEAD -> master)
Author: Arte111 <artem.emelianov.2004@gmail.com>
Date: Sat Sep 3 15:44:17 2022 +0700

    new version of first

commit 5ca285e972896a365788a5f68d4bfaceb4080336
Author: Arte111 <artem.emelianov.2004@gmail.com>
Date: Sat Sep 3 15:42:07 2022 +0700

    Version with line of code

commit b701d1666f4467c8b02b1dea3adb0aaafaa57802
Author: Arte111 <artem.emelianov.2004@gmail.com>
Date: Sat Sep 3 15:38:36 2022 +0700

    other files

commit 0e156ca9225dd35728d225605458e37696cdd193
Author: Arte111 <artem.emelianov.2004@gmail.com>
Date: Sat Sep 3 15:36:18 2022 +0700

    first empty

1111@erroremelya MINGW64 /d/na6 3/eay (master)
$ git log --pretty=oneline
70a1baabfe82119712c6772ea838f423c953d4c3 (HEAD -> master) new version of first
5ca285e972896a365788a5f68d4bfaceb4080336 Version with line of code
b701d1666f4467c8b02b1dea3adb0aaafaa57802 other files
0e156ca9225dd35728d225605458e37696cdd193 first empty

1111@erroremelya MINGW64 /d/na6 3/eay (master)
$ git log --pretty=format:"%h %ad | %s%d [%an]" --graph --date=short
* 70a1baa 2022-09-03 | new version of first (HEAD -> master) [Arte111]
* 5ca285e 2022-09-03 | Version with line of code [Arte111]
* b701d16 2022-09-03 | other files [Arte111]
* 0e156ca 2022-09-03 | first empty [Arte111]

1111@erroremelya MINGW64 /d/na6 3/eay (master)
$ |
```

Создание алиасов. Создан алиас **st** для команды **status**, и **hist** для **log** с изменённым Выводом

```
1111@erroremelya MINGW64 /d/na6 3/eay (master)
$ git config --global alias.st status

1111@erroremelya MINGW64 /d/na6 3/eay (master)
$ git st
On branch master
nothing to commit, working tree clean

1111@erroremelya MINGW64 /d/na6 3/eay (master)
$ git config --global alias.hist 'log --pretty=format:"%h %ad | %s%d [%an]" --graph --date=short'

1111@erroremelya MINGW64 /d/na6 3/eay (master)
$ git hist
* 70a1baa 2022-09-03 | new version of first (HEAD -> master) [Arte111]
* 5ca285e 2022-09-03 | version with line of code [Arte111]
* b701d16 2022-09-03 | other files [Arte111]
* 0e156ca 2022-09-03 | first empty [Arte111]

1111@erroremelya MINGW64 /d/na6 3/eay (master)
$ |
```

Перемещение по версиям

```
1111@erroremelya MINGW64 /d/na6 3/eay (master)
$ git hist
* e396ba4 2022-09-03 | version with third line (HEAD -> master) [Arte111]
* 70a1baa 2022-09-03 | new version of first [Arte111]
* 5ca285e 2022-09-03 | version with line of code [Arte111]
* b701d16 2022-09-03 | other files [Arte111]
* 0e156ca 2022-09-03 | first empty [Arte111]

1111@erroremelya MINGW64 /d/na6 3/eay (master)
$ git checkout 5ca285e
Note: switching to '5ca285e'.

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 5ca285e Version with line of code

1111@erroremelya MINGW64 /d/na6 3/eay ((5ca285e...))
$ git checkout master
Previous HEAD position was 5ca285e Version with line of code
Switched to branch 'master'

1111@erroremelya MINGW64 /d/na6 3/eay (master)
$ cat first.py
first line

second line

third line
1111@erroremelya MINGW64 /d/na6 3/eay (master)
$ |
```

Тегирование

```
1111@erroremelya MINGW64 /d/na6 3/eay (master)
$ git hist
* e396ba4 2022-09-03 | version with third line (HEAD -> master) [Arte111]
* 70a1baa 2022-09-03 | new version of first [Arte111]
* 5ca285e 2022-09-03 | version with line of code [Arte111]
* b701d16 2022-09-03 | other files [Arte111]
* 0e156ca 2022-09-03 | first empty [Arte111]

1111@erroremelya MINGW64 /d/na6 3/eay (master)
$ git tag v0.3

1111@erroremelya MINGW64 /d/na6 3/eay (master)
$ git hist
* e396ba4 2022-09-03 | version with third line (HEAD -> master, tag: v0.3) [Arte111]
* 70a1baa 2022-09-03 | new version of first [Arte111]
* 5ca285e 2022-09-03 | version with line of code [Arte111]
* b701d16 2022-09-03 | other files [Arte111]
* 0e156ca 2022-09-03 | first empty [Arte111]

1111@erroremelya MINGW64 /d/na6 3/eay (master)
$ |
```

## Клонирование репозитория

```
1111@erroremelya MINGW64 /d/Ла6 3/test
$ git clone https://github.com/Arte111/Lab3
Cloning into 'Lab3'...
remote: Enumerating objects: 38, done.
remote: Counting objects: 100% (38/38), done.
remote: Compressing objects: 100% (24/24), done.
remote: Total 38 (delta 5), reused 34 (delta 4), pack-reused 0
Receiving objects: 100% (38/38), 759.32 KiB | 1.76 MiB/s, done.
Resolving deltas: 100% (5/5), done.
1111@erroremelya MINGW64 /d/Ла6 3/test
$ |
```

### Отмена ошибочных изменений:

#### 1. Отмена локальных изменений

```
1111@erroremelya MINGW64 /d/Ла6 3/eay (master)
$ cat first.py
first line

second line

third line

mistake
1111@erroremelya MINGW64 /d/Ла6 3/eay (master)
$ git st
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:   first.py

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

1111@erroremelya MINGW64 /d/Ла6 3/eay (master)
$ git checkout first.py
Updated 1 path from the index

1111@erroremelya MINGW64 /d/Ла6 3/eay (master)
$ git st
On branch master
nothing to commit, working tree clean

1111@erroremelya MINGW64 /d/Ла6 3/eay (master)
$ cat first.py
first line

second line

third line
1111@erroremelya MINGW64 /d/Ла6 3/eay (master)
$ |
```

#### 2. Отмена проиндексированных изменений

```
1111@erroremelya MINGW64 /d/Ла6 3/eay (master)
$ cat first.py
first line

second line

third line

mistake
1111@erroremelya MINGW64 /d/Ла6 3/eay (master)
$ git add first.py

1111@erroremelya MINGW64 /d/Ла6 3/eay (master)
$ git st
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   first.py

1111@erroremelya MINGW64 /d/Ла6 3/eay (master)
$ git reset HEAD first.py
Unstaged changes after reset:
M   first.py

1111@erroremelya MINGW64 /d/Ла6 3/eay (master)
$ git checkout first.py
Updated 1 path from the index

1111@erroremelya MINGW64 /d/Ла6 3/eay (master)
$ cat first.py
first line

second line

third line
1111@erroremelya MINGW64 /d/Ла6 3/eay (master)
$ |
```

### 3. Отмена неправильного коммита

```
1111@erroremelya MINGW64 /d/паб 3/eay (master)
$ cat first.py
first line

second line

third line

mistake
1111@erroremelya MINGW64 /d/паб 3/eay (master)
$ git add first.py

1111@erroremelya MINGW64 /d/паб 3/eay (master)
$ git st
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   first.py

1111@erroremelya MINGW64 /d/паб 3/eay (master)
$ git commit -m "with mistake"
[master 82857dc] with mistake
1 file changed, 3 insertions(+), 1 deletion(-)

1111@erroremelya MINGW64 /d/паб 3/eay (master)
$ git hist
* 82857dc 2022-09-03 | with mistake (HEAD -> master) [Arte111]
* 67da1fd 2022-09-03 | Revert "mistake in code" [Arte111]
* 7a6aaee 2022-09-03 | mistake in code [Arte111]
* e396ba4 2022-09-03 | Version with third line (tag: v0.3) [Arte111]
* 70a1baa 2022-09-03 | new version of first [Arte111]
* 5ca285e 2022-09-03 | Version with line of code [Arte111]
* b701d16 2022-09-03 | other files [Arte111]
* 0e156ca 2022-09-03 | first empty [Arte111]

1111@erroremelya MINGW64 /d/паб 3/eay (master)
$ git revert HEAD --no-edit
[master 5d7dc0c] Revert "with mistake"
Date: Sat Sep 3 16:22:51 2022 +0700
1 file changed, 1 insertion(+), 3 deletions(-)

1111@erroremelya MINGW64 /d/паб 3/eay (master)
$ git hist
* 5d7dc0c 2022-09-03 | Revert "with mistake" (HEAD -> master) [Arte111]
* 82857dc 2022-09-03 | with mistake [Arte111]
* 67da1fd 2022-09-03 | Revert "mistake in code" [Arte111]
* 7a6aaee 2022-09-03 | mistake in code [Arte111]
* e396ba4 2022-09-03 | Version with third line (tag: v0.3) [Arte111]
* 70a1baa 2022-09-03 | new version of first [Arte111]
* 5ca285e 2022-09-03 | Version with line of code [Arte111]
* b701d16 2022-09-03 | other files [Arte111]
* 0e156ca 2022-09-03 | first empty [Arte111]

1111@erroremelya MINGW64 /d/паб 3/eay (master)
$ |
```

### Ветвление

```
1111@erroremelya MINGW64 /d/паб 3/eay (master)
$ git checkout -b the_first_branch
Switched to a new branch 'the_first_branch'

1111@erroremelya MINGW64 /d/паб 3/eay (the_first_branch)
$ git st
On branch the_first_branch
nothing to commit, working tree clean

1111@erroremelya MINGW64 /d/паб 3/eay (the_first_branch)
$ |
```

## Слияние веток

```
1111@erroremelya MINGW64 /d/Ла6 3/eay (the_first_branch|MERGING)
$ git hist --all
* b01f55a 2022-09-03 | README (master) [Arte111]
* 8888bdd 2022-09-03 | with cfg file (HEAD -> the_first_branch) [Arte111]
* 75cf7cb 2022-09-03 | with cfg file [Arte111]
/
* 5d7dc0c 2022-09-03 | Revert "with mistake" [Arte111]
* 82857dc 2022-09-03 | with mistake [Arte111]
* 67da1fd 2022-09-03 | Revert "mistake in code" [Arte111]
* 7a6aaee 2022-09-03 | mistake in code [Arte111]
* e396ba4 2022-09-03 | Version with third line (tag: v0.3) [Arte111]
* 70a1baa 2022-09-03 | new version of first [Arte111]
* 5ca285e 2022-09-03 | Version with line of code [Arte111]
* b701d16 2022-09-03 | other files [Arte111]
* 0e156ca 2022-09-03 | first empty [Arte111]

1111@erroremelya MINGW64 /d/Ла6 3/eay (the_first_branch|MERGING)
$ git merge master
fatal: You have not concluded your merge (MERGE_HEAD exists).
Please, commit your changes before you merge.

1111@erroremelya MINGW64 /d/Ла6 3/eay (the_first_branch|MERGING)
$ git st
On branch the_first_branch
All conflicts fixed but you are still merging.
(use "git commit" to conclude merge)

Changes to be committed:
  new file:   README

1111@erroremelya MINGW64 /d/Ла6 3/eay (the_first_branch|MERGING)
$ git commit -m "README too"
[the_first_branch d8cd58d] README too

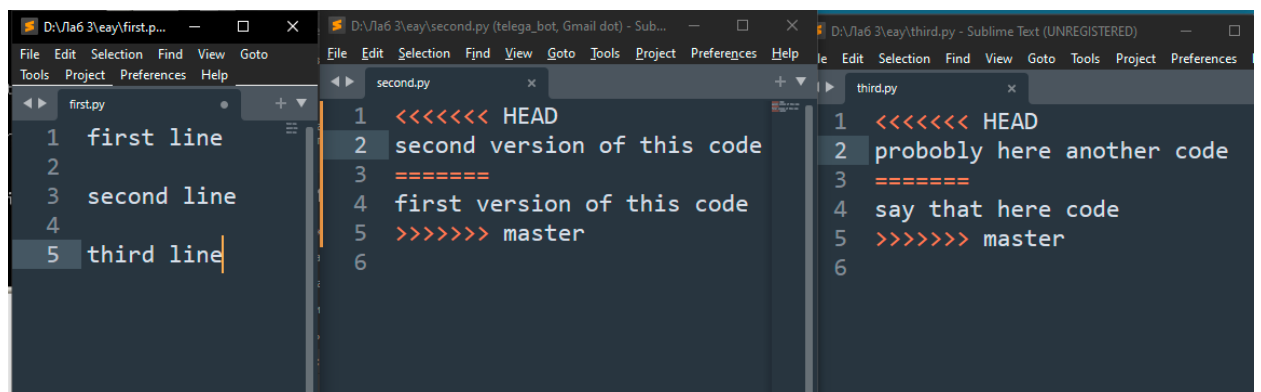
1111@erroremelya MINGW64 /d/Ла6 3/eay (the_first_branch)
$ git hist --all
* d8cd58d 2022-09-03 | README too (HEAD -> the_first_branch) [Arte111]
/
* b01f55a 2022-09-03 | README (master) [Arte111]
* 8888bdd 2022-09-03 | with cfg file [Arte111]
* 75cf7cb 2022-09-03 | with cfg file [Arte111]
/
* 5d7dc0c 2022-09-03 | Revert "with mistake" [Arte111]
* 82857dc 2022-09-03 | with mistake [Arte111]
* 67da1fd 2022-09-03 | Revert "mistake in code" [Arte111]
* 7a6aaee 2022-09-03 | mistake in code [Arte111]
* e396ba4 2022-09-03 | Version with third line (tag: v0.3) [Arte111]
* 70a1baa 2022-09-03 | new version of first [Arte111]
* 5ca285e 2022-09-03 | Version with line of code [Arte111]
* b701d16 2022-09-03 | other files [Arte111]
* 0e156ca 2022-09-03 | first empty [Arte111]

1111@erroremelya MINGW64 /d/Ла6 3/eay (the_first_branch)
$
```

Конфликт при слиянии веток происходит, если отличается содержимое одного файла в разных ветках

```
1111@erroremelya MINGW64 /d/Ла6 3/eay (the_second_branch)
$ git merge master
Auto-merging second.py
CONFLICT (content): Merge conflict in second.py
Auto-merging third.py
CONFLICT (content): Merge conflict in third.py
Automatic merge failed; fix conflicts and then commit the result.

1111@erroremelya MINGW64 /d/Ла6 3/eay (the_second_branch|MERGING)
$
```



Необходимо вручную привести файл к версии одной из веток

```
1111@erroremelya MINGW64 /d/лаб 3/eay (the_second_branch)
$ git merge master
Auto-merging second.py
CONFLICT (content): Merge conflict in second.py
Auto-merging third.py
CONFLICT (content): Merge conflict in third.py
Automatic merge failed; fix conflicts and then commit the result.

1111@erroremelya MINGW64 /d/лаб 3/eay (the_second_branch|MERGING)
$ git add .

1111@erroremelya MINGW64 /d/лаб 3/eay (the_second_branch|MERGING)
$ git commit -m "fixed conflict"
[the_second_branch 82c9151] fixed conflict

1111@erroremelya MINGW64 /d/лаб 3/eay (the_second_branch)
$ git hist --all
* 82c9151 2022-09-03 | fixed conflict (HEAD -> the_second_branch) [Arte111]
|\
| * 805fd75 2022-09-03 | add third file (master) [Arte111]
| * 63d0cb2 2022-09-03 | first version [Arte111]
| * 126d844 2022-09-03 | change third file [Arte111]
| * 6ec1c86 2022-09-03 | second version [Arte111]
```

Создание чистого репозитория

```
1111@erroremelya MINGW64 /d/лаб 3
$ cd eay

1111@erroremelya MINGW64 /d/лаб 3/eay (master)
$ cd ..

1111@erroremelya MINGW64 /d/лаб 3
$ git clone --bare eay clean_repo.git
Cloning into bare repository 'clean_repo.git'...
done.

1111@erroremelya MINGW64 /d/лаб 3
$ ls clean_repo.git
HEAD config description hooks/ info/ objects/ packed-refs refs/

1111@erroremelya MINGW64 /d/лаб 3
$ cd eay

1111@erroremelya MINGW64 /d/лаб 3/eay (master)
$ git remote add shared ../clean_repo.git
error: remote shared already exists.

1111@erroremelya MINGW64 /d/лаб 3/eay (master)
$ git add README

1111@erroremelya MINGW64 /d/лаб 3/eay (master)
$ git commit -m "added shared comment to README"
On branch master
nothing to commit, working tree clean

1111@erroremelya MINGW64 /d/лаб 3/eay (master)
$ git push shared master
Everything up-to-date

1111@erroremelya MINGW64 /d/лаб 3/eay (master)
$ |
```

Вывод: В результате выполнения лабораторной работы освоил команды для работы с системой контроля версий GIT.

## Задание 2. Контрольные вопросы

- 1.Что представляет собой система GIT? GIT – система контроля версий.
- 2.Перечислите возможности систем управления версиями. Хранение нескольких версий проекта, фиксация изменений.
- 3.Что такое коммит и когда он выполняется? Коммит – отправка последнего изменения исходного кода в репозиторий.
- 4.Как посмотреть историю коммитов? Чтобы посмотреть историю коммитов, необходимо запустить команду **git log**.

5.Как создать пустой GIT репозиторий? Чтобы создать пустой репозиторий, требуется запустить команду **git init**, после чего в текущем каталоге будет создан новый подкаталог с именем **.git**.

6.Какую архитектуру имеет система контроля версий GIT? GIT имеет архитектуру 3 деревьев.

7.Как создать новую ветку в GIT и соединить ветки? Для создания новой ветки необходимо запустить команду **git checkout -b Название ветки**. Для соединения веток – **git merge название ветки, с которой сливаемся** (файлы с одинаковым названием, должны иметь одинаковое содержание)

8.Поясните понятие «staging area». Staging area – дополнительный слой (дерево) между репозиторием и рабочими копиями. Stage необходим для того, чтобы можно было выбирать какие файлы включать в коммит.

9.Как сравнить версии файла в GIT? Для сравнения версий файла, нужно выполнить команду **git diff**.

Дополнительное задание.

Ссылка на github <https://github.com/Arte111/Lab3>