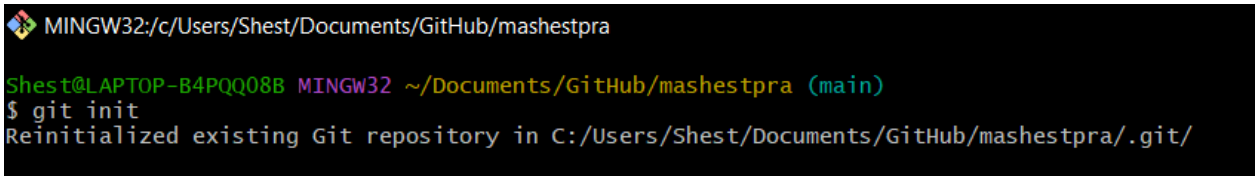
ВСР задание 2.1

Задание выполняется после проведения вебинара «Использование системы контроля версий Git».

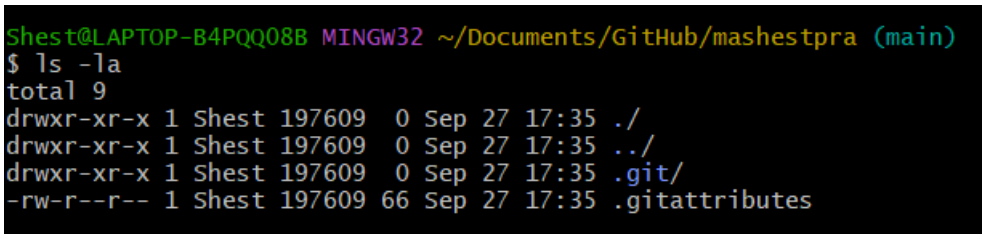
На примере основных действий, продемонстрированных на вебинаре по Git, освоить работу (и продемонстрировать выполнение показанных на вебинаре действий) с одним из визуальных клиентов для работы с Git на выбор:

Вариант 1. Git – Git Bash

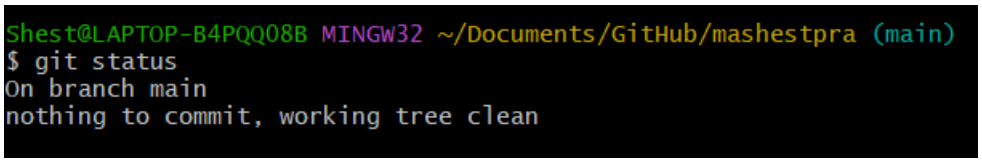
1. git init – команда, которая используется для создания пустого репозитория Git



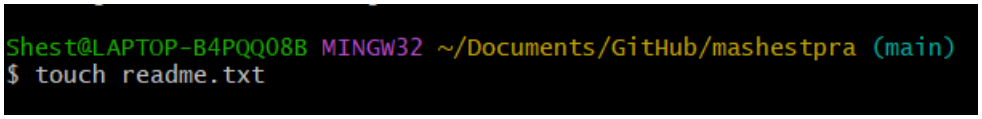
2. ls –la – команда, которая используется для того, чтобы убедится, что репозиторий создан.



3. git status – команда, которая помогает увидеть статус репозитория, также можно посмотреть на какой ветке находимся.



4. touch readme.txt (touch + название файла) – команда, которая помогает создать текстовый файл в репозитории.



5. nano readme.txt (nano + название файла) – команда, которая позволяет ввести или изменить какой-нибудь текст внутрь файла. После введения данной функции, появится окно, в которое мы записываем необходимый текст, чтобы его сохранить зажимаем клавиши Ctrl+S, чтобы выйти Ctrl+X.

```
Shest@LAPTOP-B4PQQ08B MINGW32 ~/Documents/GitHub/mashestpra (main)
$ nano readme.txt
```

Для фиксации изменений используем команду git status.

```
Shest@LAPTOP-B4PQQ08B MINGW32 ~/Documents/GitHub/mashestpra (main)
$ git status
On branch main
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        readme.txt

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

6. git add readme.txt (git add + название файла) – команда помогает отслеживать файл. В пункте 5 файл еще не отслеживается, поэтому вводим данную функцию.

```
Shest@LAPTOP-B4PQQ08B MINGW32 ~/Documents/GitHub/mashestpra (main)
$ git add readme.txt
warning: in the working copy of 'readme.txt', LF will be replaced by CRLF the next time Git touches it
```

После вводим команду git status, что проверить новый статус

```
Shest@LAPTOP-B4PQQ08B MINGW32 ~/Documents/GitHub/mashestpra (main)
$ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   readme.txt
```

7. git commit -m «First file» (-m «комментарий»)– команда, которая фиксирует статус. Так как у нас нет пока больше файлов, ничего фиксироваться не будет.

```
Shest@LAPTOP-B4PQQ08B MINGW32 ~/Documents/GitHub/mashestpra (main)
$ git commit -m "First file"
[main 4ccc813] First file
 1 file changed, 1 insertion(+)
 create mode 100644 readme.txt

Shest@LAPTOP-B4PQQ08B MINGW32 ~/Documents/GitHub/mashestpra (main)
$ git status
On branch main
nothing to commit, working tree clean
```

8. Далее еще раз выполняем команду nano, чтобы изменить текст в файле.

```
Shest@LAPTOP-B4PQQ08B MINGW32 ~/Documents/GitHub/mashestpra (main)
$ nano readme.txt
```

9. После проверки статуса, замечаем, что появились изменения и их выделяют красным цветом. Далее используем команду add.

```
Shest@LAPTOP-B4PQQ08B MINGW32 ~/Documents/GitHub/mashestpra (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:   readme.txt

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

Shest@LAPTOP-B4PQQ08B MINGW32 ~/Documents/GitHub/mashestpra (main)
$ git add readme.txt
warning: in the working copy of 'readme.txt', LF will be replaced by CRLF the next time Git touches it
```

10. Добавляем файлы, можно с помощью обычного копирования файла. Добавить их можно с помощью команды git add ‘*.txt’. По известной схеме проверяем статус файла.

```
Shest@LAPTOP-B4PQQ08B MINGW32 ~/Documents/GitHub/mashestpra (main)
$ git add '*.txt'
warning: in the working copy of 'readme after.txt', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'readme now.txt', LF will be replaced by CRLF the next time Git touches it

Shest@LAPTOP-B4PQQ08B MINGW32 ~/Documents/GitHub/mashestpra (main)
$ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   readme after.txt
        new file:   readme now.txt
        modified:   readme.txt
```

11. Фиксируем с помощью команды commit и добавляем комментарий

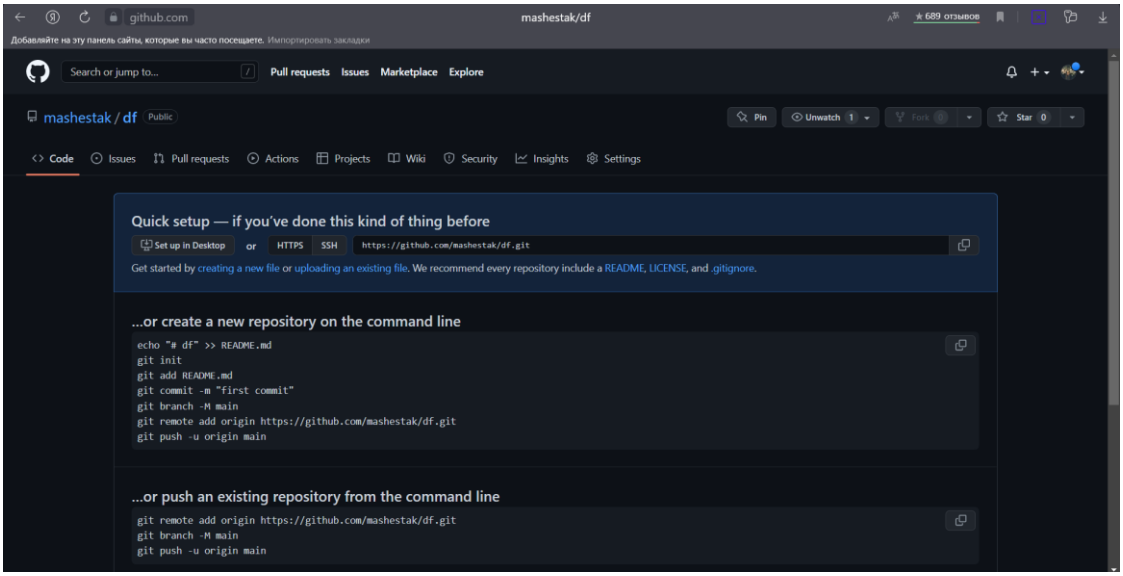
```
Shest@LAPTOP-B4PQQ08B MINGW32 ~/Documents/GitHub/mashestpra (main)
$ git commit -m "All files with txt extension were added"
[main c00f18d] All files with txt extension were added
 3 files changed, 3 insertions(+), 1 deletion(-)
 create mode 100644 readme after.txt
 create mode 100644 readme now.txt
```

12. git log (git log --summary) – git log команда, которая позволяет посмотреть историю всех фиксаций. git log --summary команда, которая помогает узнать более подробную информацию.

```
Shest@LAPTOP-B4PQQ08B MINGW32 ~/Documents/GitHub/mashestpra (main)
$ git log
commit c00f18d01fbf5ca5d5e4cd762c760414780075f2 (HEAD -> main)
Author: mashestak <98894528+mashestak@users.noreply.github.com>
Date:   Tue Sep 27 17:45:46 2022 +0300

    All files with txt extension were added

commit 4ccc8130a7026e578f704786c7cefd2938757498
Author: mashestak <98894528+mashestak@users.noreply.github.com>
Date:   Tue Sep 27 17:39:01 2022 +0300

    First file

commit 6c3a38274ed201908abd25a06c8737a54b8762fa
Author: mashestak <98894528+mashestak@users.noreply.github.com>
Date:   Tue Sep 27 17:35:13 2022 +0300

    Initial commit
```

```
Shest@LAPTOP-B4PQQ08B MINGW32 ~/Documents/GitHub/mashestpra (main)
$ git log --summary
commit c00f18d01fbf5ca5d5e4cd762c760414780075f2 (HEAD -> main)
Author: mashestak <98894528+mashestak@users.noreply.github.com>
Date:   Tue Sep 27 17:45:46 2022 +0300

    All files with txt extension were added

 create mode 100644 readme after.txt
 create mode 100644 readme now.txt

commit 4ccc8130a7026e578f704786c7cefd2938757498
Author: mashestak <98894528+mashestak@users.noreply.github.com>
Date:   Tue Sep 27 17:39:01 2022 +0300

    First file

 create mode 100644 readme.txt

commit 6c3a38274ed201908abd25a06c8737a54b8762fa
Author: mashestak <98894528+mashestak@users.noreply.github.com>
Date:   Tue Sep 27 17:35:13 2022 +0300

    Initial commit

 create mode 100644 .gitattributes
```

13. git remote add origin + ссылка на репозиторий – команда, которая позволяет добавлять удаленный репозиторий. Для этого необходимо зайти на сайт github, скопировать ссылку и вставить ее после рассматриваемой команды




```
Shest@LAPTOP-B4PQQ08B MINGW32 ~/Documents/GitHub/mashestpra (main)
$ git remote add origin https://github.com/mashestak/mashestpra.git
```

14. git push -u origin main (git push –u + имя репозитория + название базовой ветки) – команда позволяет выгружать файлы в удаленный репозиторий.

```
Shest@LAPTOP-B4PQQ08B MINGW32 ~/Documents/GitHub/mashestpra (main)
$ git push -u origin main
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (9/9), 836 bytes | 209.00 KiB/s, done.
Total 9 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/mashestak/mashestpra.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

Shest@LAPTOP-B4PQQ08B MINGW32 ~/Documents/GitHub/mashestpra (main)
$ git push -u origin main
Everything up-to-date
branch 'main' set up to track 'origin/main'.
```

15. git pull origin main – команда позволяет получить обратно файл, в котором были произведены изменения. В предпоследней строчке появилась информация о том, что прибавилось 2 строчки в репозитории.

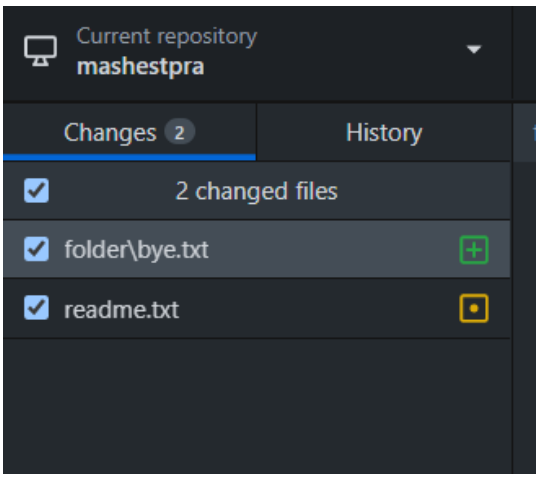
```
Shest@LAPTOP-B4PQQ08B MINGW32 ~/Documents/GitHub/mashestpra (main)
$ git pull origin main
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 647 bytes | 24.00 KiB/s, done.
From https://github.com/mashestak/mashestpra
 * branch            main       -> FETCH_HEAD
   c00f18d..2d27c0c  main       -> origin/main
Updating c00f18d..2d27c0c
Fast-forward
 readme.txt | 2 ++
 1 file changed, 2 insertions(+)
```

16. Сначала изменим текст в файле, используя уже известную команду nano. Далее чтобы посмотреть, что именно изменилось воспользуемся командой git diff HEAD и посмотрим на изменения с версией, которая была зафиксирована. Плюс данной версии, что изменения показаны цветом.

```
Shest@LAPTOP-B4PQQ08B MINGW32 ~/Documents/GitHub/mashestpra (main)
$ git diff HEAD

Shest@LAPTOP-B4PQQ08B MINGW32 ~/Documents/GitHub/mashestpra (main)
$ nano readme.txt

Shest@LAPTOP-B4PQQ08B MINGW32 ~/Documents/GitHub/mashestpra (main)
$ git diff HEAD
diff --git a/readme.txt b/readme.txt
index d4e9ff7..2201a1c 100644
--- a/readme.txt
+++ b/readme.txt
@@ -1,3 +1,3 @@
-Hello, world!
-Hello, world!
-Hello, world!
+Hello, world
+Hello, world
+Hello, world
```

17. Добавляя каталог, используем команду mkdir folder (mkdir + называние каталога). После добавиляем в каталоге файл через команду touch + название файла. Проверяем статус.

```
Shest@LAPTOP-B4PQQ08B MINGW32 ~/Documents/GitHub/mashestpra (main)
$ mkdir folder

Shest@LAPTOP-B4PQQ08B MINGW32 ~/Documents/GitHub/mashestpra (main)
$ touch folder/bye.txt

Shest@LAPTOP-B4PQQ08B MINGW32 ~/Documents/GitHub/mashestpra (main)
$ git status
On branch main
Your branch is up to date with 'origin/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:   readme.txt

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

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



18. Команда git diff --staged позволяет сравнить состояние репозитория с тем, который был до этого момента.

```
Shest@LAPTOP-B4PQQ08B MINGW32 ~/Documents/GitHub/mashestpra (main)
$ git diff --staged
diff --git a/folder/bye.txt b/folder/bye.txt
new file mode 100644
index 0000000..e69de29
```

19. git reset folder – команда позволяет отменить изменения. После выполнения команды опять набираем команду git diff –staged и видим, что изменений не было.

```
Shest@LAPTOP-B4PQQ08B MINGW32 ~/Documents/GitHub/mashestpra (main)
$ git reset folder/bye.txt
Unstaged changes after reset:
M       readme.txt

Shest@LAPTOP-B4PQQ08B MINGW32 ~/Documents/GitHub/mashestpra (main)
$ git diff --staged
```

20. Чтобы вернуть наш файл в первоначальный вид, то используем команду git checkout -- readme.txt (git checkout -- + название файла). Проверяем статус файла.

```
Shest@LAPTOP-B4PQQ08B MINGW32 ~/Documents/GitHub/mashestpra (main)
$ git checkout -- readme.txt

Shest@LAPTOP-B4PQQ08B MINGW32 ~/Documents/GitHub/mashestpra (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

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

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

А команда cat readme.txt (cat + название файла) помогут вывести изначальный текст в консоле.

```
Shest@LAPTOP-B4PQQ08B MINGW32 ~/Documents/GitHub/mashestpra (main)
$ cat readme.txt
Hello, world!
Hello, world!
Hello, world!
```

21. Для того чтобы создать ветку используем команду git branch + название ветки. Далее чтобы посмотреть на какой ветке мы работаем выбираем команду git branch.

```
Shest@LAPTOP-B4PQQ08B MINGW32 ~/Documents/GitHub/mashestpra (main)
$ git branch clean_up

Shest@LAPTOP-B4PQQ08B MINGW32 ~/Documents/GitHub/mashestpra (main)
$ git branch
  clean_up
* main
```

22. Чтобы переключится на другую ветку, набираем команду git checkout + название ветки и выводим название ветки на экран через git branch.

```
Shest@LAPTOP-B4PQQ08B MINGW32 ~/Documents/GitHub/mashestpra (main)
$ git checkout clean_up
Switched to branch 'clean_up'

Shest@LAPTOP-B4PQQ08B MINGW32 ~/Documents/GitHub/mashestpra (clean_up)
$ git branch
* clean_up
  main
```

23. Команда, позволяющая удалить каталог и файлы этой ветки, выглядит следующим образом git rm -r

```
Shest@LAPTOP-B4PQQ08B MINGW32 ~/Documents/GitHub/mashestpra (clean_up)
$ rm -r folder
```

Необходимо после этого фиксировать изменения.

```
Shest@LAPTOP-B4PQQ08B MINGW32 ~/Documents/GitHub/mashestpra (clean_up)
$ git commit -m 'Deleted folder and files'
[clean_up 55f2780] Deleted folder and files
 2 files changed, 2 deletions(-)
 delete mode 100644 readme after.txt
 delete mode 100644 readme now.txt
```

24. Чтобы выполнить команду слияния, вводим git merge + название ветки

```
Shest@LAPTOP-B4PQQ08B MINGW32 ~/Documents/GitHub/mashestpra (main)
$ git merge clean_up
Updating 2d27c0c..55f2780
Fast-forward
 readme after.txt | 1 -
 readme now.txt   | 1 -
 2 files changed, 2 deletions(-)
 delete mode 100644 readme after.txt
 delete mode 100644 readme now.txt
```

25. git branch -d clean_up – команда позволяет удалить ненужную ветку.

```
Shest@LAPTOP-B4PQQ08B MINGW32 ~/Documents/GitHub/mashestpra (main)
$ git branch -d clean_up
Deleted branch clean_up (was 55f2780).
```

26. git push – команда позволяет выгрузить все изменения в удаленный репозиторий.

```
Shest@LAPTOP-B4PQQ08B MINGW32 ~/Documents/GitHub/mashestpra (main)
$ git push
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 245 bytes | 245.00 KiB/s, done.
Total 2 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/mashestak/mashestpra.git
   2d27c0c..55f2780  main -> main

Shest@LAPTOP-B4PQQ08B MINGW32 ~/Documents/GitHub/mashestpra (main)
$ git remote add origin https://github.com/mashestak/mashestpra.git
error: remote origin already exists.
```