

# Лабораторная работа №1

## Основы Git и Github

### Цель работы:

Выполнение лабораторной работы направлено на изучение:

1. наиболее распространенных практик в области контроля версий программного обеспечения, его использования в командной разработке ПО и *DevOps*;
2. концепции *Git*, основанной на понятиях репозитория и ветвления версий ПО;
3. порядка использования *GitHub* и его базовых операций.

### Порядок работы:

1. Зарегистрировался в *GitHub*, как показано на рисунке 1.

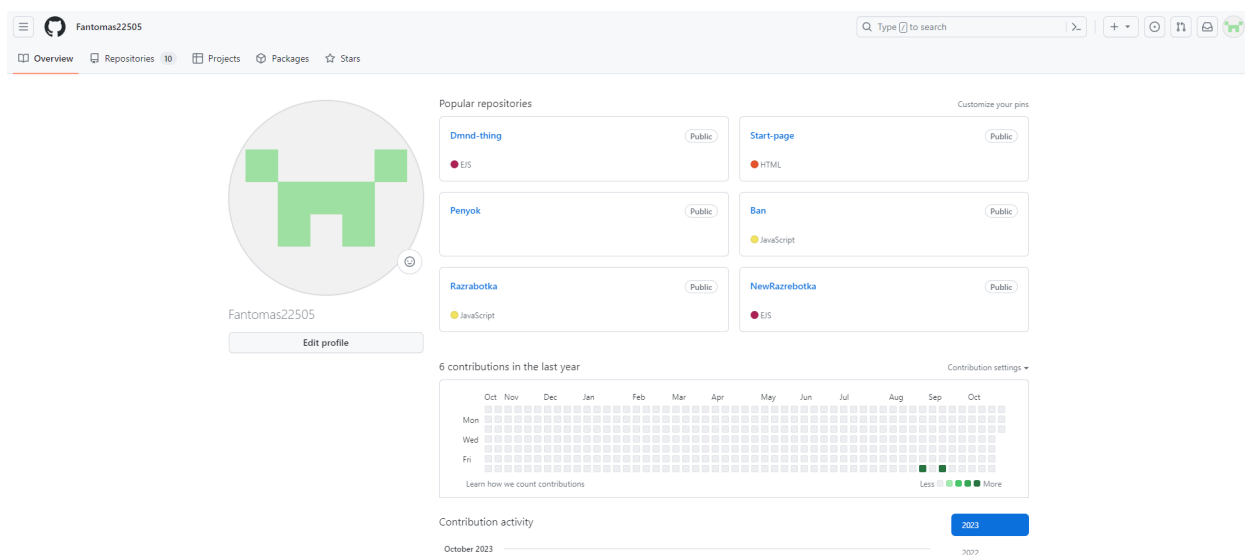


Рисунок 1 - Аккаунт *GitHub*

2. Создал новый репозиторий: задал имя репозитория, добавил описание, выбрал видимость репозитория «публичный», выбрал опцию «*Initialize this repository with a README*», затем отредактировал файл *README*, как показано на рисунке 2.

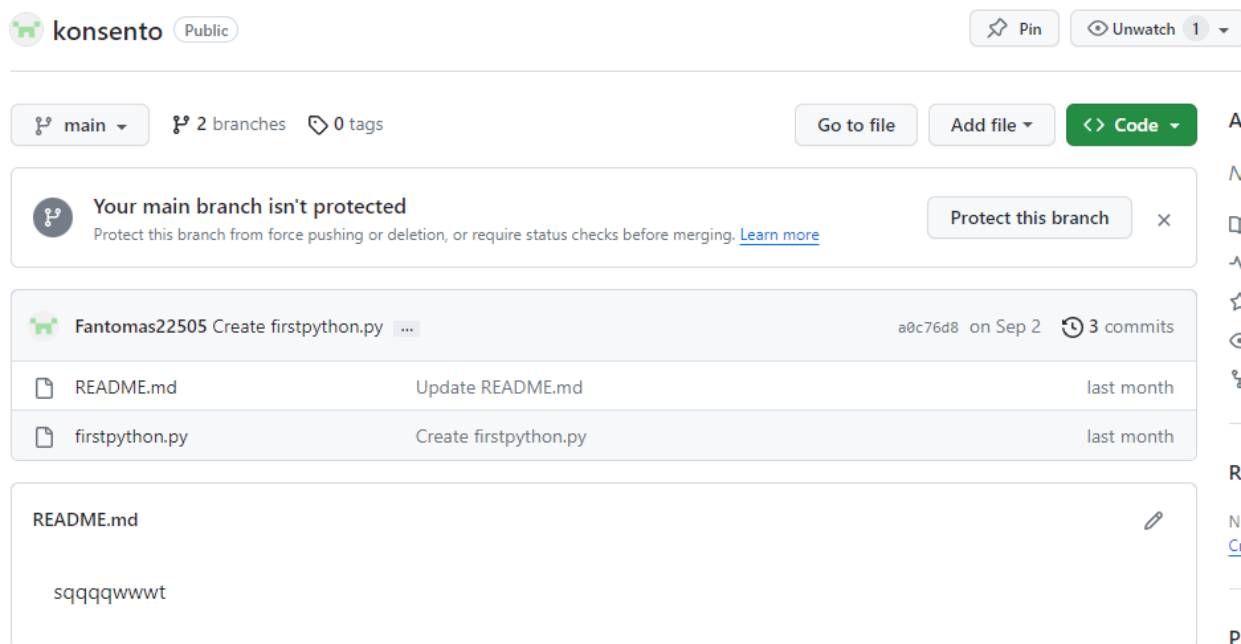


Рисунок 2 - Новый репозиторий

Создал файл *firstpython.py* с помощью встроенного веб-редактора *GitHub*. Зафиксировал изменения в репозитории, как показано на рисунке 3.

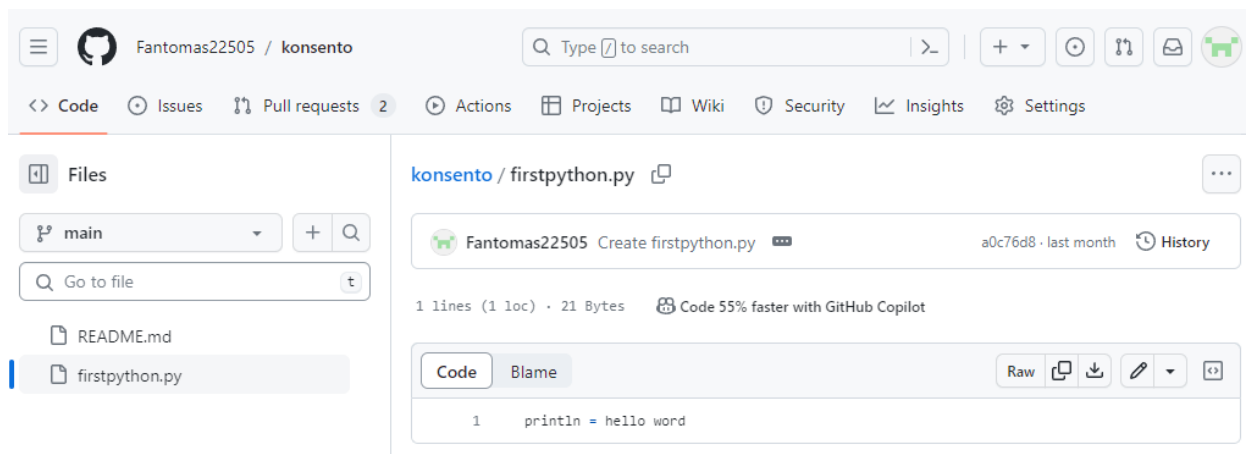


Рисунок 3 - Создание файла

3. Создал новую ветку «*Number2*», создал новый файл. Убедился, что файл, добавленный в дочернюю ветку, не добавляется автоматически в основную ветку, как показано на рисунке 4.

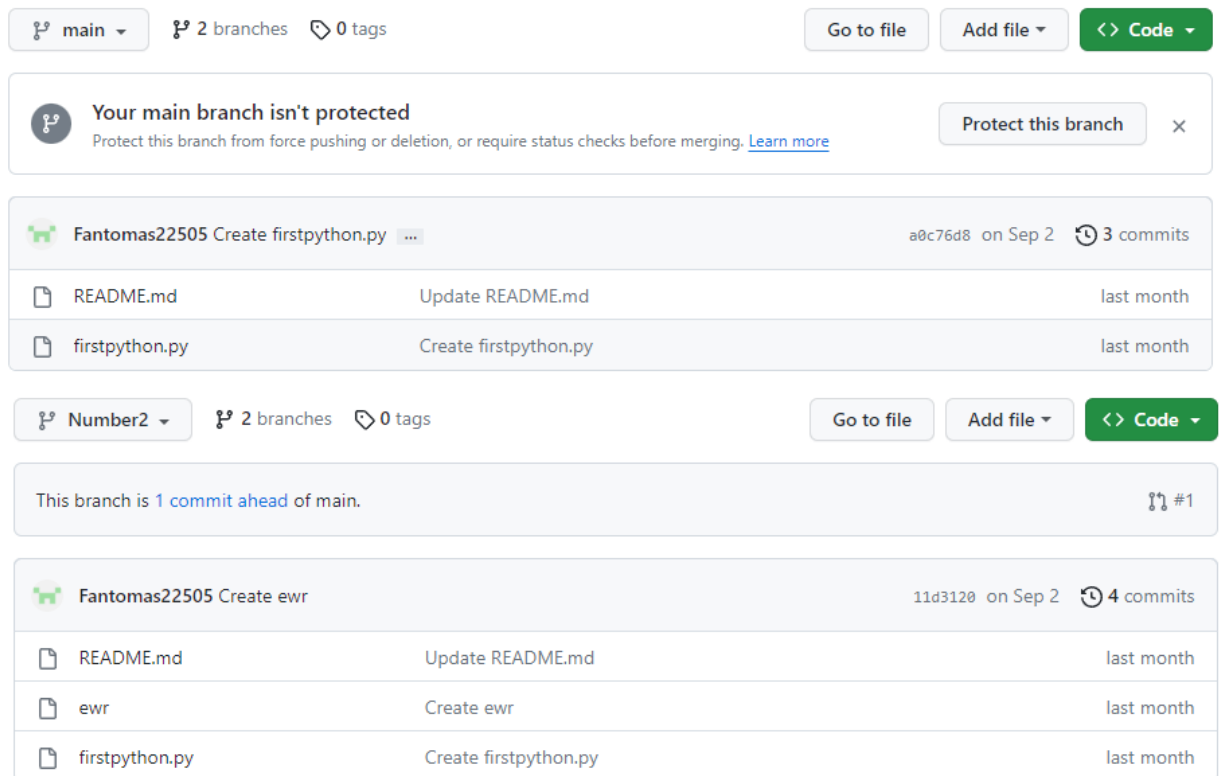


Рисунок 4 - Создание новой ветки

В *Child\_Branch* нажал кнопку «*Compare & pull request*», убедился, что в списке указан 1 *changed file*, как показано на рисунке 5.

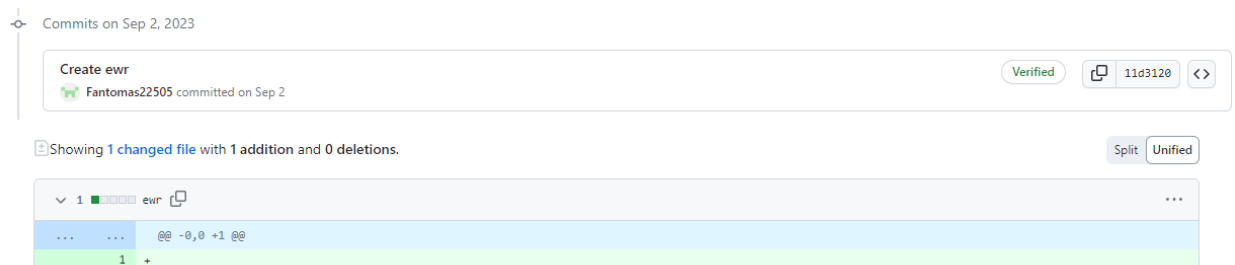


Рисунок 5 - Список измененных файлов

4. Чтобы объединить ветки по запросу *pull request* в проекте, открыл вкладку «*Pull requests*». Отображается список ожидающих запросов на включение. Перешел на нужный *pull request* и нажал «*Merge pull request*», чтобы принять запрос на включение и объединить обновления, как показано на рисунке 6.

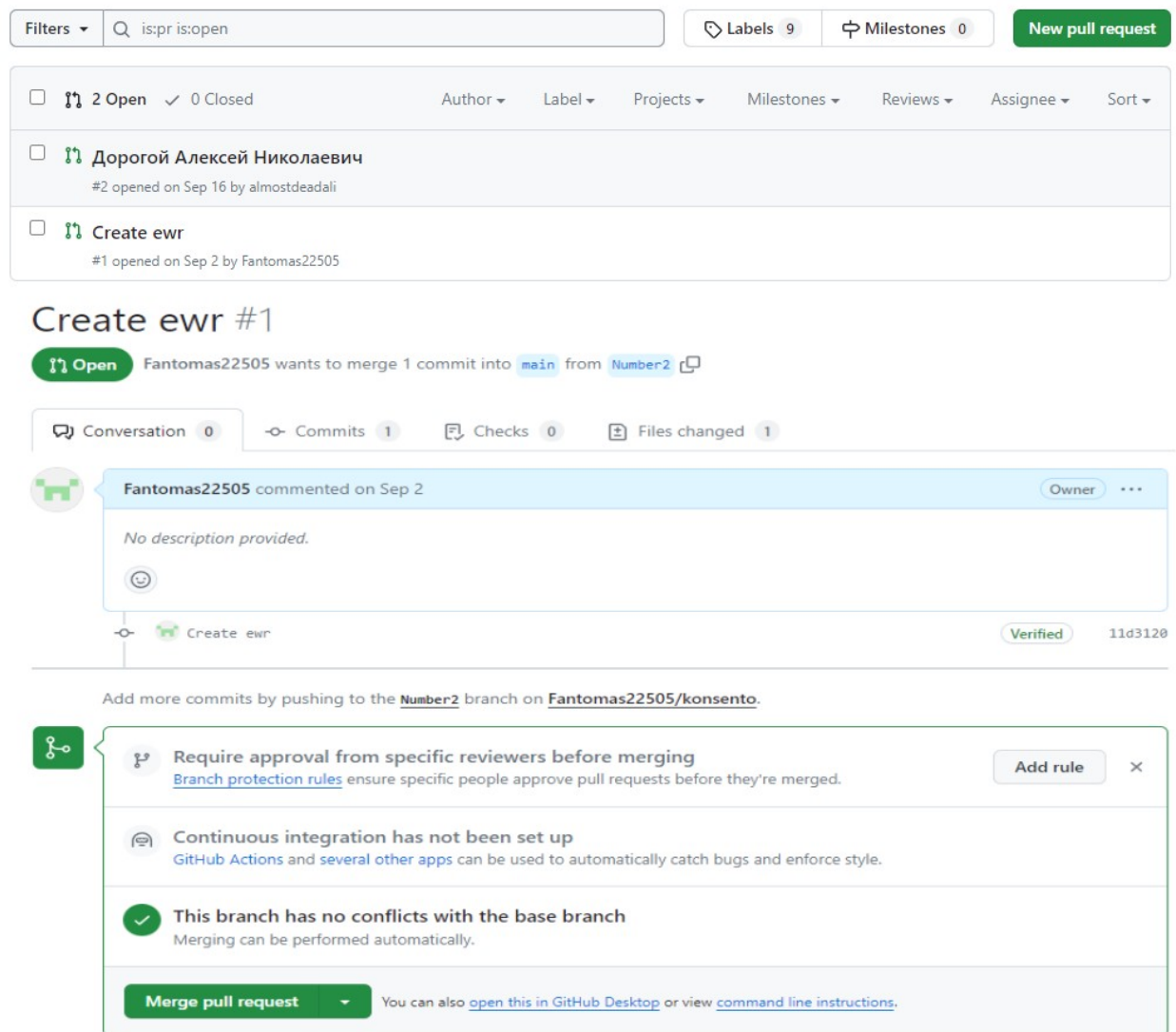


Рисунок 6 - Запрос на включение

5. Произвел работу с локальным репозиторием посредством командной строки *Windows PowerShell* через специальное приложение *Git*.

Создал каталог *муреро* и перешел в созданный каталог, затем создал локальный репозиторий, вывел на экран содержимое подкаталога *.git*, в котором находится локальный репозиторий, как показано на рисунке 7.

```
Windows PowerShell

Попробуйте новую кроссплатформенную оболочку PowerShell (https://aka.ms/pscore6)

PS C:\Users\Gera_Of_Livia> cd documents
>> mkdir myrepo
>> cd myrepo
>> git init
>> ls .git

Каталог: C:\Users\Gera_Of_Livia\documents

Mode                LastWriteTime         Length Name
----                -
d-----          24.10.2023    17:43             myrepo
Initialized empty Git repository in C:/Users/Gera_Of_Livia/Documents/myrepo/.git/

Каталог: C:\Users\Gera_Of_Livia\documents\myrepo\.git

Mode                LastWriteTime         Length Name
----                -
d-----          24.10.2023    17:43             hooks
d-----          24.10.2023    17:43             info
d-----          24.10.2023    17:43             objects
d-----          24.10.2023    17:43             refs
-a-----          24.10.2023    17:43         130 config
-a-----          24.10.2023    17:43         73 description
-a-----          24.10.2023    17:43         23 HEAD
```

Рисунок 7 - Новый каталог

Создал пустой файл *newfile*, добавил его в репозиторий. Прежде чем зафиксировать изменения, сообщил *git* информацию пользователя. Появившийся в репозитории *newfile* зафиксировал с добавлением сообщения «*added new file*», как показано на рисунке 8.

```
Windows PowerShell

PS C:\Users\Gera_Of_Livia\documents\myrepo> ni newfile
>> git add newfile
>> git config --global user.email "you@example.com"
>> git config --global user.name "Your Name"

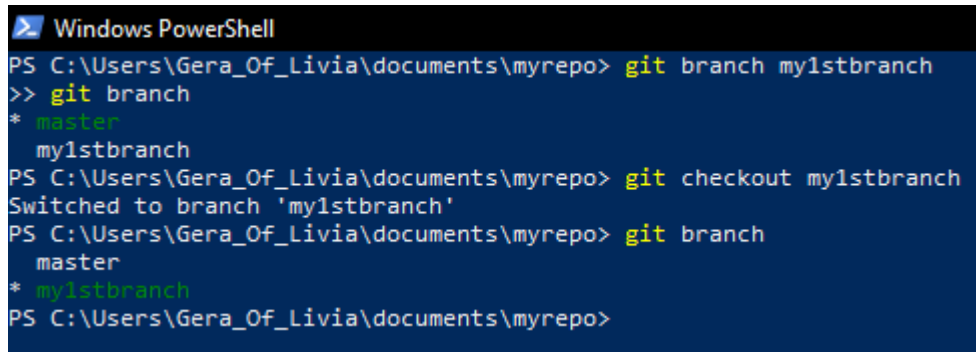
Каталог: C:\Users\Gera_Of_Livia\documents\myrepo

Mode                LastWriteTime         Length Name
----                -
-a-----          24.10.2023    17:50             0 newfile

PS C:\Users\Gera_Of_Livia\documents\myrepo> git commit -m "added newfile"
[master (root-commit) f512e8f] added newfile
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 newfile
PS C:\Users\Gera_Of_Livia\documents\myrepo>
```

Рисунок 8 - Новый файл

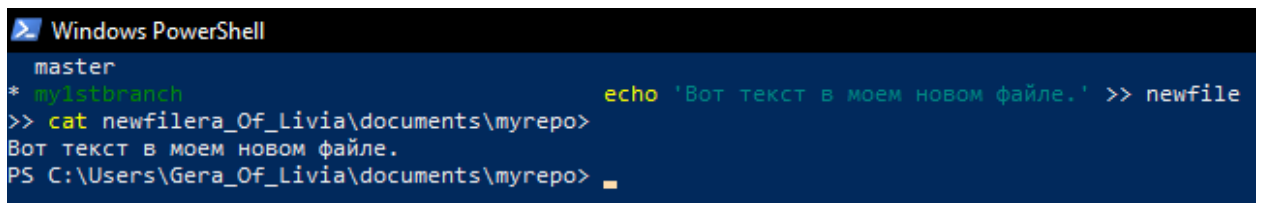
Создал новую ветку под названием *my1stbranch*, убедился в наличии двух веток в репозитории и переключился с ветки *master* на ветку *my1stbranch*, как показано на рисунке 9.



```
Windows PowerShell
PS C:\Users\Gera_Of_Livia\documents\myrepo> git branch my1stbranch
>> git branch
* master
  my1stbranch
PS C:\Users\Gera_Of_Livia\documents\myrepo> git checkout my1stbranch
Switched to branch 'my1stbranch'
PS C:\Users\Gera_Of_Livia\documents\myrepo> git branch
  master
* my1stbranch
PS C:\Users\Gera_Of_Livia\documents\myrepo>
```

Рисунок 9 - Новая ветка

Внес изменения в *newfile* путем добавления текста, убедился что текст добавлен, как показано на рисунке 10.



```
Windows PowerShell
master
* my1stbranch
>> echo 'Вот текст в моем новом файле.' >> newfile
>> cat newfile
Вот текст в моем новом файле.
PS C:\Users\Gera_Of_Livia\documents\myrepo>
```

Рисунок 10 - Запись и чтение из файла

Создал новый файл *readme.md* и добавил в репозиторий, проверил изменения в текущей ветке *my1stbranch*, затем добавил *newfile* явно, как показано на рисунке 11.

```
Windows PowerShell
PS C:\Users\Gera_Of_Livia\documents\myrepo> ni readme.md
>> git add readme.md
>> git status

Каталог: C:\Users\Gera_Of_Livia\documents\myrepo

Mode                LastWriteTime         Length Name
----                -
-a-----          24.10.2023      18:29             0 readme.md
On branch my1stbranch
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   readme.md

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:   newfile

PS C:\Users\Gera_Of_Livia\documents\myrepo> git add *
>> git status
On branch my1stbranch
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   newfile
        new file:   readme.md
```

Рисунок 11 - Добавление файла

Сохранил изменения в ветку, прикрепив сообщение «*added readme.md modified newfile*», далее получил историю последних коммитов — последний коммит в *my1stbranch*, а также предыдущий коммит в *master*, как показано на рисунке 12.

```
> git commit -m "added readme.md modified newfile"
[my1stbranch c9f556f] added readme.md modified newfile
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 readme.md
PS C:\Users\Gera_Of_Livia\documents\myrepo> git log
commit c9f556f7c886e4694cfd8565842b08c2f70cd7eb (HEAD -> my1stbranch)
Author: Your Name <you@example.com>
Date:   Tue Oct 24 18:39:14 2023 +0300

    added readme.md modified newfile

commit f512e8f728079462e55350bbb1a2ba38395a9bde (master)
Author: Your Name <you@example.com>
Date:   Tue Oct 24 17:52:42 2023 +0300
```

Рисунок 12 - Сохранение изменений

Произвел отмену изменений, используя ярлык *HEAD* для отката последнего коммита, как показано на рисунке 13.

```
Windows PowerShell
PS C:\Users\Gera_Of_Livia\documents\myrepo> git revert HEAD --no-edit
[my1stbranch dfe0c9d] Revert "added readme.md modified newfile"
Date: Tue Oct 24 18:43:56 2023 +0300
2 files changed, 0 insertions(+), 0 deletions(-)
delete mode 100644 readme.md
```

Рисунок 13 - Отмена изменений

Создал новый *goodfile* и убедился, что файл зафиксирован в *my1stbranch*, как показано на рисунке 14.

```
Windows PowerShell
PS C:\Users\Gera_Of_Livia\documents\myrepo> ni goodfile
>> git add goodfile
>> git commit -m "added goodfile"
>> git log

Каталог: C:\Users\Gera_Of_Livia\documents\myrepo

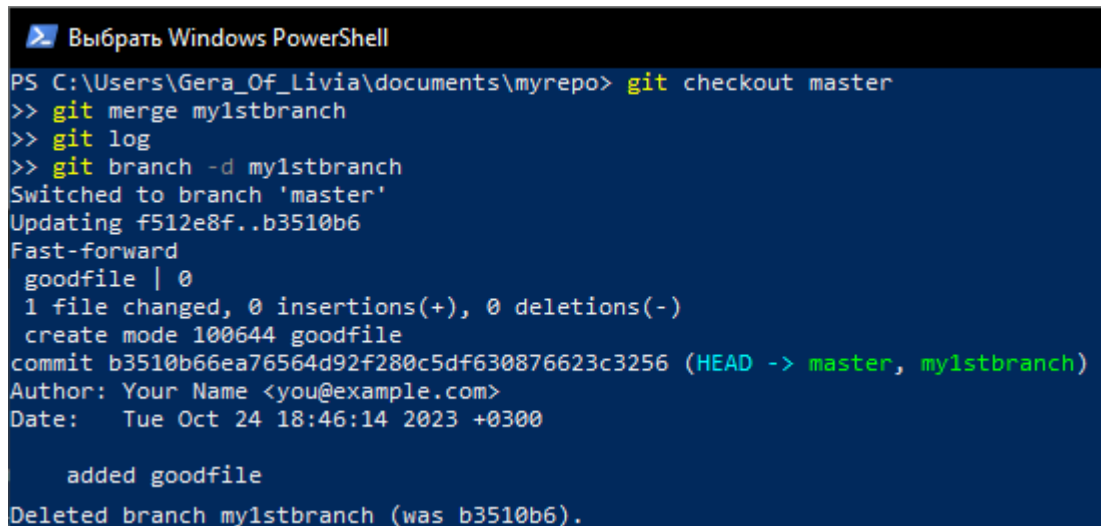
Mode                LastWriteTime         Length Name
----                -
-a-----         24.10.2023      18:46             0 goodfile
[my1stbranch b3510b6] added goodfile
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 goodfile
commit b3510b66ea76564d92f280c5df630876623c3256 (HEAD -> my1stbranch)
Author: Your Name <you@example.com>
Date: Tue Oct 24 18:46:14 2023 +0300

    added goodfile
```

Рисунок 14 - Новый файл

Далее объединил содержимое *my1stbranch* с основной веткой, для этого сначала сделал ветку *master* активной. Произвел слияние веток и вывел на экран журнал. После того, как слияние успешно завершилось, удалил ветку *my1stbranch*, как показано на рисунке 15.





```
Выбрать Windows PowerShell
PS C:\Users\Gera_Of_Livia\documents\myrepo> git checkout master
>> git merge my1stbranch
>> git log
>> git branch -d my1stbranch
Switched to branch 'master'
Updating f512e8f..b3510b6
Fast-forward
 goodfile | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 goodfile
commit b3510b66ea76564d92f280c5df630876623c3256 (HEAD -> master, my1stbranch)
Author: Your Name <you@example.com>
Date: Tue Oct 24 18:46:14 2023 +0300

    added goodfile
Deleted branch my1stbranch (was b3510b6).
```

Рисунок 15 - Слияние веток

### Задание №1:

```
git checkout -b newbranch //Создал новую ветку newbranch
ni newbranchfile //Создал пустой файл newbranchfile
git add newbranchfile //Добавил файл в свою ветку
git commit -m "добавлен newbranchfile" //Зафиксировал
изменения в новой ветке

git revert HEAD --no-edit //Отменил последние зафиксированные
изменения

ni newgoodfile //Создал новый файл с именем newgoodfile
git add newgoodfile //Добавил последний файл в новую ветку
git commit -m "добавлен newgoodfile" //Зафиксировал
изменения

git checkout master //Переключился на основную ветку
git merge newbranch //Объединил изменения в новой ветке с
основной
```

Результат выполнения представлен на рисунке 16.

```
Windows PowerShell
PS C:\Users\Gera_Of_Livia\documents\myrepo> git checkout -b newbranch
>> ni newbranchfile
>> git add newbranchfile
>> git commit -m "добавлен newbranchfile"
>> git revert HEAD --no-edit
>> ni newgoodfile
>> git add newgoodfile
>> git commit -m "добавлен newgoodfile"
>> git checkout master
>> git merge newbranch
Switched to a new branch 'newbranch'

Каталог: C:\Users\Gera_Of_Livia\documents\myrepo

Mode                LastWriteTime         Length Name
----                -
-a-----         24.10.2023         22:17             0 newbranchfile
[newbranch 6c04553] добавлен newbranchfile
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 newbranchfile
[newbranch bec1d7d] Revert "добавлен newbranchfile"
 Date: Tue Oct 24 22:17:58 2023 +0300
 1 file changed, 0 insertions(+), 0 deletions(-)
 delete mode 100644 newbranchfile
-a-----         24.10.2023         22:17             0 newgoodfile
[newbranch 22985d1] добавлен newgoodfile
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 newgoodfile
Switched to branch 'master'
Updating b3510b6..22985d1
Fast-forward
 newgoodfile | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 newgoodfile
```

Рисунок 16 - Процесс выполнения задания

## Задание №2:

Выполнил *fork* проекта второго студента, как показано на рисунке 17.

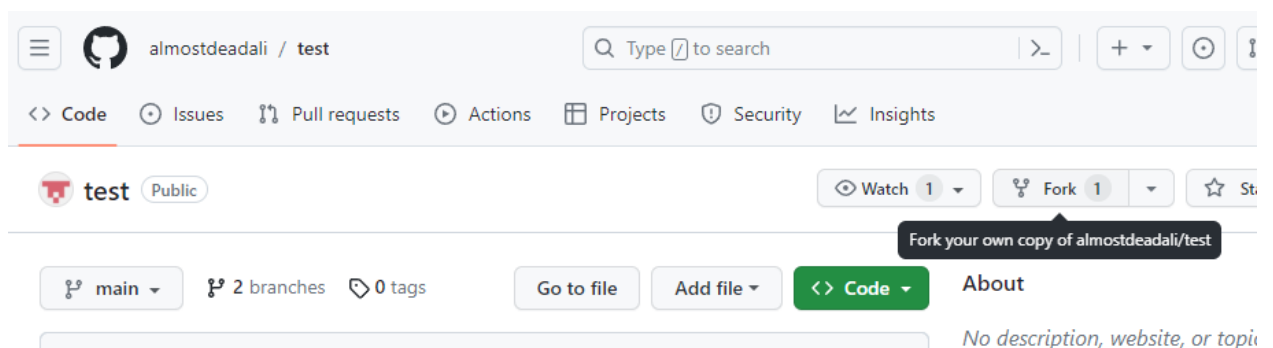
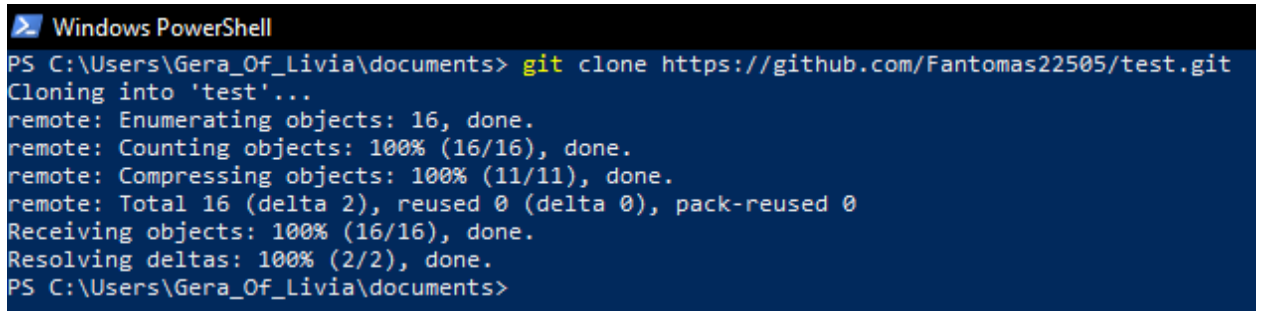


Рисунок 17 - Fork проекта второго студента

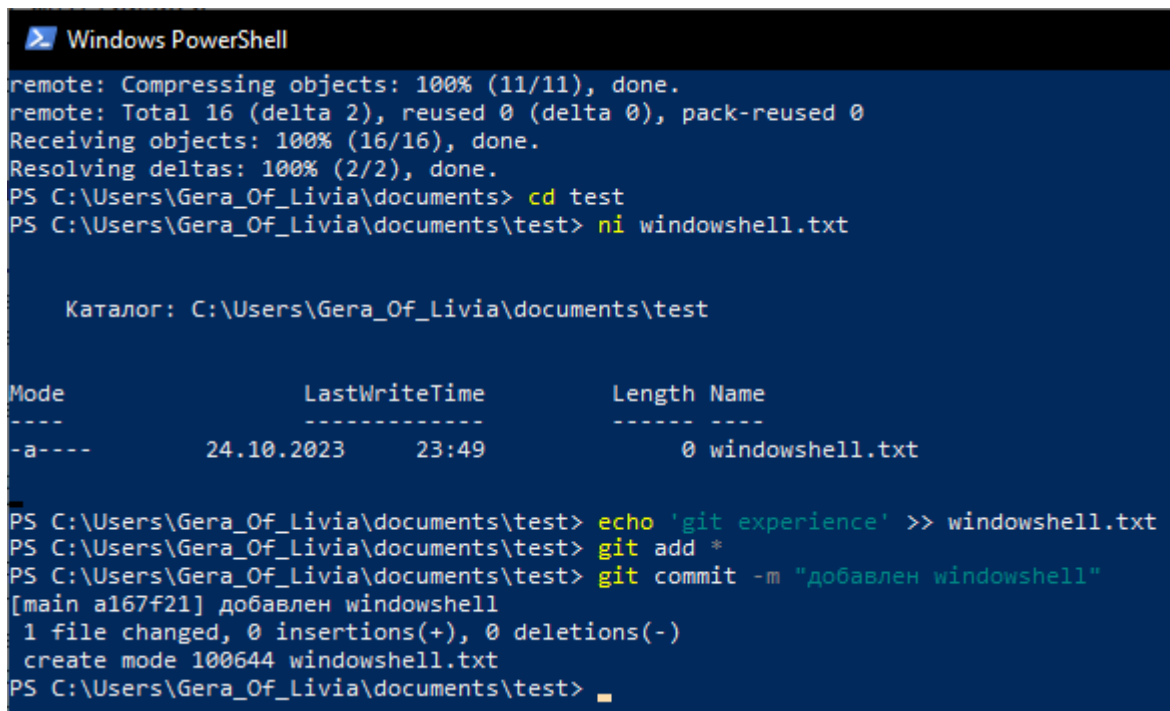
Выполнил *clone* проекта в локальный репозиторий с помощью *Windows PowerShell*, как показано на рисунке 18.



```
Windows PowerShell
PS C:\Users\Gera_Of_Livia\documents> git clone https://github.com/Fantomas22505/test.git
Cloning into 'test'...
remote: Enumerating objects: 16, done.
remote: Counting objects: 100% (16/16), done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 16 (delta 2), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (16/16), done.
Resolving deltas: 100% (2/2), done.
PS C:\Users\Gera_Of_Livia\documents>
```

Рисунок 18 - Clone проекта

Добавил новый файл в репозиторий, зафиксировал изменения, как показано на рисунке 19.



```
Windows PowerShell
remote: Compressing objects: 100% (11/11), done.
remote: Total 16 (delta 2), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (16/16), done.
Resolving deltas: 100% (2/2), done.
PS C:\Users\Gera_Of_Livia\documents> cd test
PS C:\Users\Gera_Of_Livia\documents\test> ni windowshell.txt

Каталог: C:\Users\Gera_Of_Livia\documents\test

Mode                LastWriteTime         Length Name
----                -
-a-----         24.10.2023   23:49             0 windowshell.txt

PS C:\Users\Gera_Of_Livia\documents\test> echo 'git experience' >> windowshell.txt
PS C:\Users\Gera_Of_Livia\documents\test> git add *
PS C:\Users\Gera_Of_Livia\documents\test> git commit -m "добавлен windowshell"
[main a167f21] добавлен windowshell
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 windowshell.txt
PS C:\Users\Gera_Of_Livia\documents\test>
```

Рисунок 19 - Новый файл

Выполнил синхронизацию с *fork*-репозиторием, как показано на рисунке 20.

```
Выбрать Windows PowerShell
PS C:\Users\Gera_Of_Livia\documents\test> git push origin main
info: please complete authentication in your browser...
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 327 bytes | 327.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Fantomas22505/test.git
 095ecf1..a167f21  main -> main
PS C:\Users\Gera_Of_Livia\documents\test>
```

Рисунок 20 - Синхронизация с *fork*-репозиторием

Сформировал *pull request* к *origin* проекту на прием данного изменения, как показано на рисунке 21.

## Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#). [Learn more about diff comp](#)

base repository: almostdeadali/test

base: main

head repository: Fantomas22505/test

compare: main

✓ Able to merge. These br

добавлен windowshell

Write

Preview

H B I

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

☒ Allow edits by maintainers

Create pull request

Рисунок 21 - Формирование *pull request*

**Вывод:** изучили концепцию *Git*, основанную на понятиях репозитория и ветвления версий ПО, изучили порядок использования *GitHub* и его базовых операций, научились работать с онлайн-хостингом GitHub.