



МИНОБРНАУКИ РОССИИ
федеральное государственное автономное образовательное учреждение
высшего образования
«Московский государственный технологический университет «СТАНКИН»
(ФГАОУ ВО «МГТУ «СТАНКИН»)

Институт цифровых
интеллектуальных систем

Кафедра
компьютерных систем управления

Дисциплина «Основы системного программного обеспечения»

Отчет по лабораторной работе №1

Выполнил
студент гр. АДБ-22-06:

(дата)

(подпись)

Сидоров А.В.

Проверил
к.т.н., доцент

(дата)

(подпись)

Ковалев И.А.

Москва 2025 г.

Оглавление

1. Цель и задачи работы	3
2. Ход работы	3
2.1 Регистрация и первичная настройка GitHub.....	3
2.2 Создание удалённого репозитория	3
2.3 Установка и глобальная настройка Git.....	4
2.4 Создание папки и файла.....	5
2.5 Инициализация. Добавление и фиксация изменений	6
2.6 Связывание локального и удалённого репозитория.....	7
2.7 Работа с ветками	9
2.8 Откаты изменений	13
2.9 Игнорирование файлов	15
2.10 Совместная работа (Fork → Clone → Pull Request)	17
4. Результаты	17
5. Выводы.....	17

1. Цель и задачи работы

1. Закрепить навыки базовой работы с Git в локальном и удалённом репозитории.
2. Освоить типовой рабочий процесс: add → commit → push и pull → merge.
3. Научиться создавать ветки, решать конфликты, откатываться к предыдущим версиям.
4. Освоить совместную работу через Fork/Clone и файл .gitignore для исключения лишних данных.

2. Ход работы

2.1 Регистрация и первичная настройка GitHub

1. Перешёл на <https://github.com> и создал бесплатный публичный аккаунт.
2. В Settings → Repositories → Default branch сменил название основной ветки с main на master.

2.2 Создание удалённого репозитория

1. Нажал New repository → ввёл имя firstApp, описание и оставил тип Public.
2. Поставил галочку «Add README». В результате GitHub создал репозиторий.

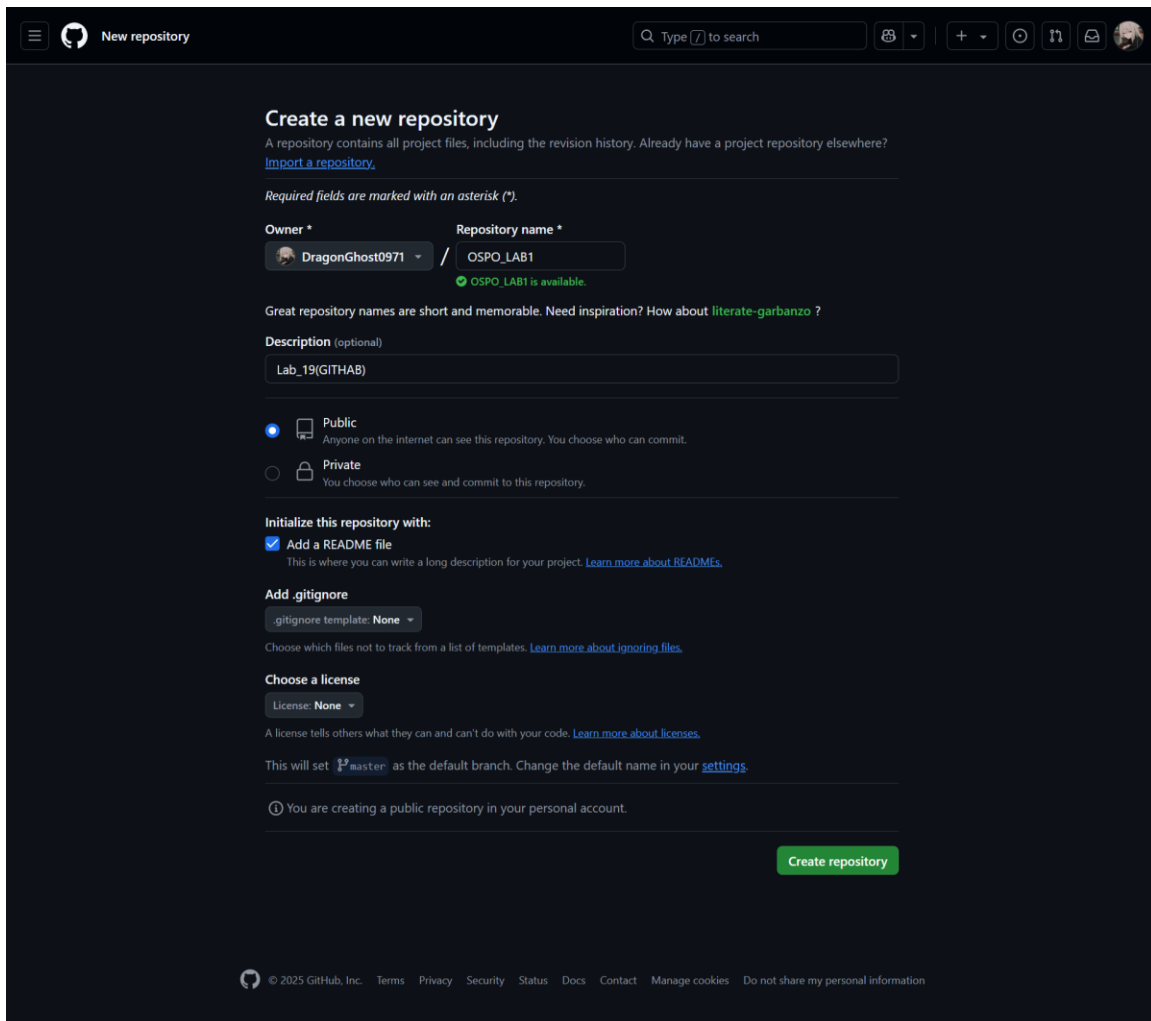
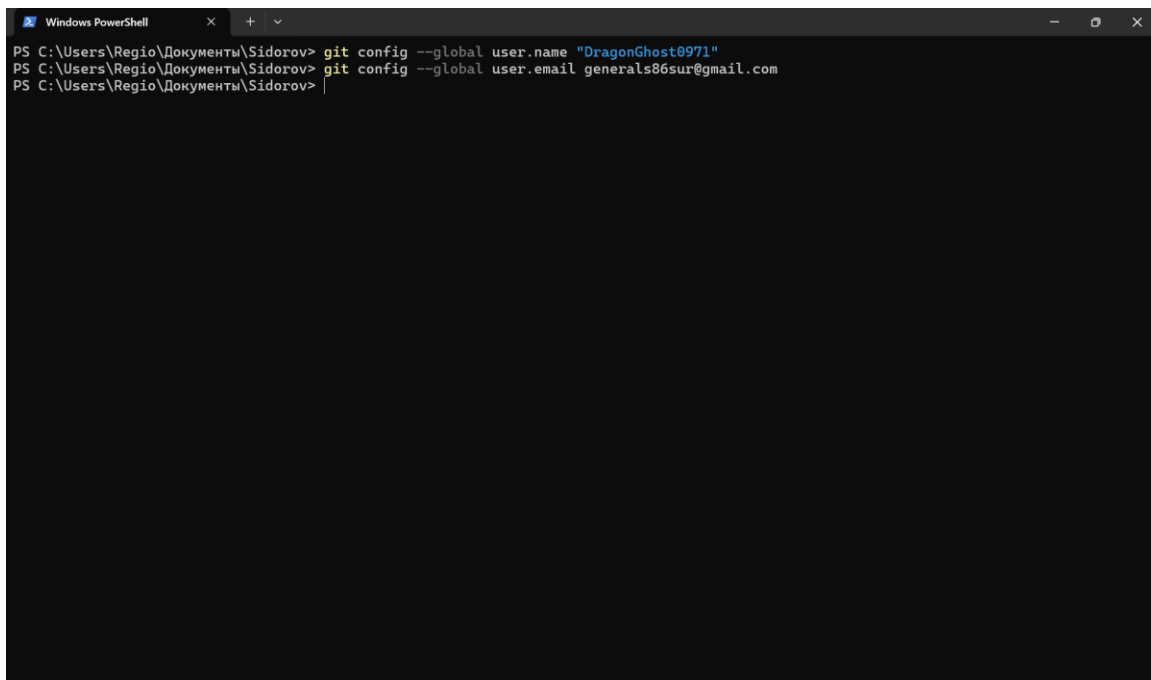


Рис. 1 – Создание репозитория

2.3 Установка и глобальная настройка Git

```
git --version
git config --global user.name "Name"
git config --global user.email "mail@mail.ru"
```

После ввода команд данные сохраняются в `~/.gitconfig`; каждое дальнейшее изменение будет подписано правильным именем и почтой.



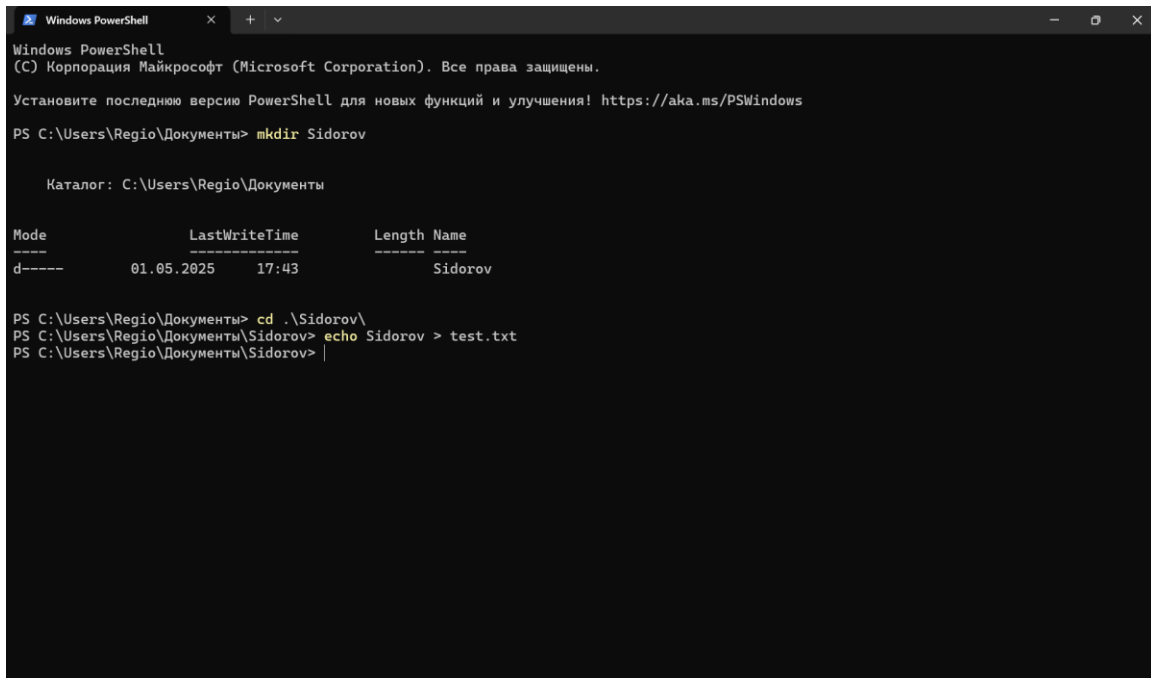
```
Windows PowerShell
PS C:\Users\Regio\Документы\Sidorov> git config --global user.name "DragonGhost0971"
PS C:\Users\Regio\Документы\Sidorov> git config --global user.email generals86sur@gmail.com
PS C:\Users\Regio\Документы\Sidorov> |
```

Рис. 2 - Глобальная настройка git

2.4 Создание папки и файла

Создал папку со своей фамилией и файла с помощью команд:

```
Mkdir Имя_папки
Echo > имя_файла.txt
```



```
Windows PowerShell
(C) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

Установите последнюю версию PowerShell для новых функций и улучшения! https://aka.ms/PSWindows

PS C:\Users\Regio\Документы> mkdir Sidorov

Каталог: C:\Users\Regio\Документы

Mode                LastWriteTime         Length Name
----                -
d-----          01.05.2025   17:43             Sidorov

PS C:\Users\Regio\Документы> cd .\Sidorov\
PS C:\Users\Regio\Документы\Sidorov> echo Sidorov > test.txt
PS C:\Users\Regio\Документы\Sidorov> |
```

Рис. 3 - Создание папки и текстового файла

2.5 Инициализация. Добавление и фиксация изменений

```
git init
git add .
git commit -m "first commit"
```

Git создал снимок (commit) с уникальным SHA-1.

```
Windows PowerShell
PS C:\Users\Regio\Документы\Sidorov> git init
Initialized empty Git repository in C:/Users/Regio/Документы/Sidorov/.git/
PS C:\Users\Regio\Документы\Sidorov> git status
On branch master

No commits yet

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

nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\Regio\Документы\Sidorov> git add .
PS C:\Users\Regio\Документы\Sidorov> git status
On branch master

No commits yet

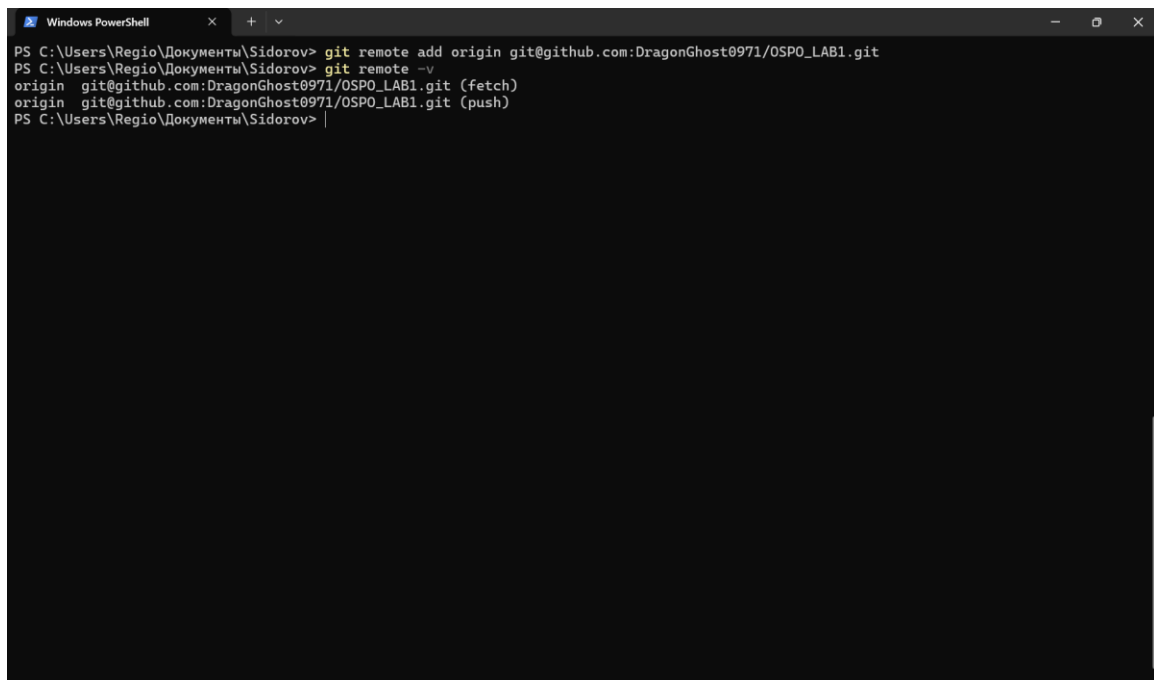
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   test.txt

PS C:\Users\Regio\Документы\Sidorov> git commit -m "First commit"
[master (root-commit) 089d2d5] First commit
1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 test.txt
PS C:\Users\Regio\Документы\Sidorov> git status
On branch master
nothing to commit, working tree clean
PS C:\Users\Regio\Документы\Sidorov> |
```

Рис. 4 - Инициализация. Добавление и фиксация изменений

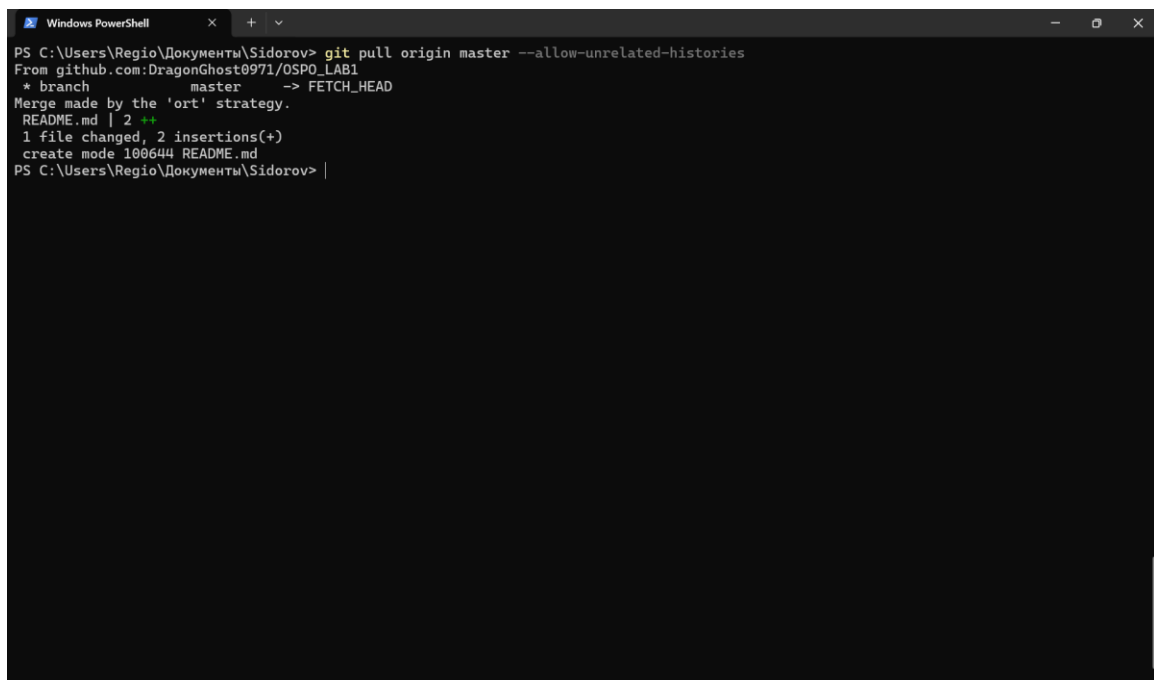
2.6 Связывание локального и удалённого репозитория

```
git remote add origin url
git pull origin master --allow-unrelated-histories
```



```
Windows PowerShell
PS C:\Users\Regio\Документы\Sidorov> git remote add origin git@github.com:DragonGhost0971/OSP0_LAB1.git
PS C:\Users\Regio\Документы\Sidorov> git remote -v
origin  git@github.com:DragonGhost0971/OSP0_LAB1.git (fetch)
origin  git@github.com:DragonGhost0971/OSP0_LAB1.git (push)
PS C:\Users\Regio\Документы\Sidorov> |
```

Рис. 5 - связывание локального с удалённым репозиторием



```
Windows PowerShell
PS C:\Users\Regio\Документы\Sidorov> git pull origin master --allow-unrelated-histories
From github.com:DragonGhost0971/OSP0_LAB1
 * branch          master      -> FETCH_HEAD
Merge made by the 'ort' strategy.
 README.md | 2 ++
1 file changed, 2 insertions(+)
create mode 100644 README.md
PS C:\Users\Regio\Документы\Sidorov> |
```

Рис. 6 - Подтягивание удалённого репозитория к локальному

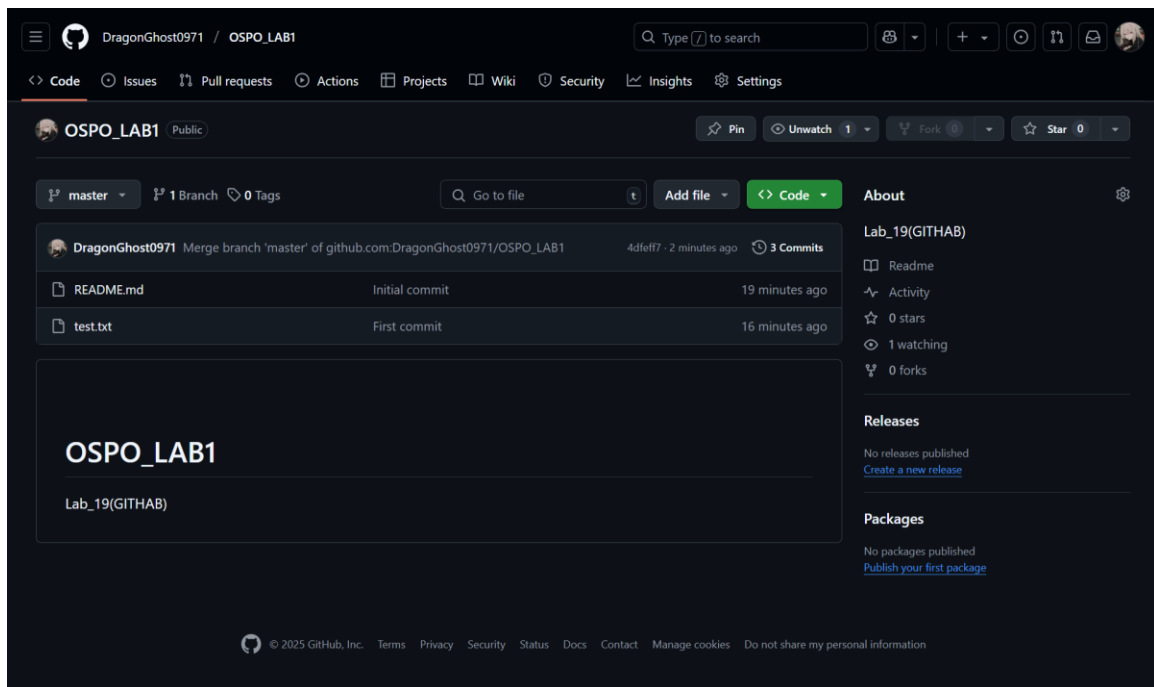


Рис. 7 - Удалённый репозиторий

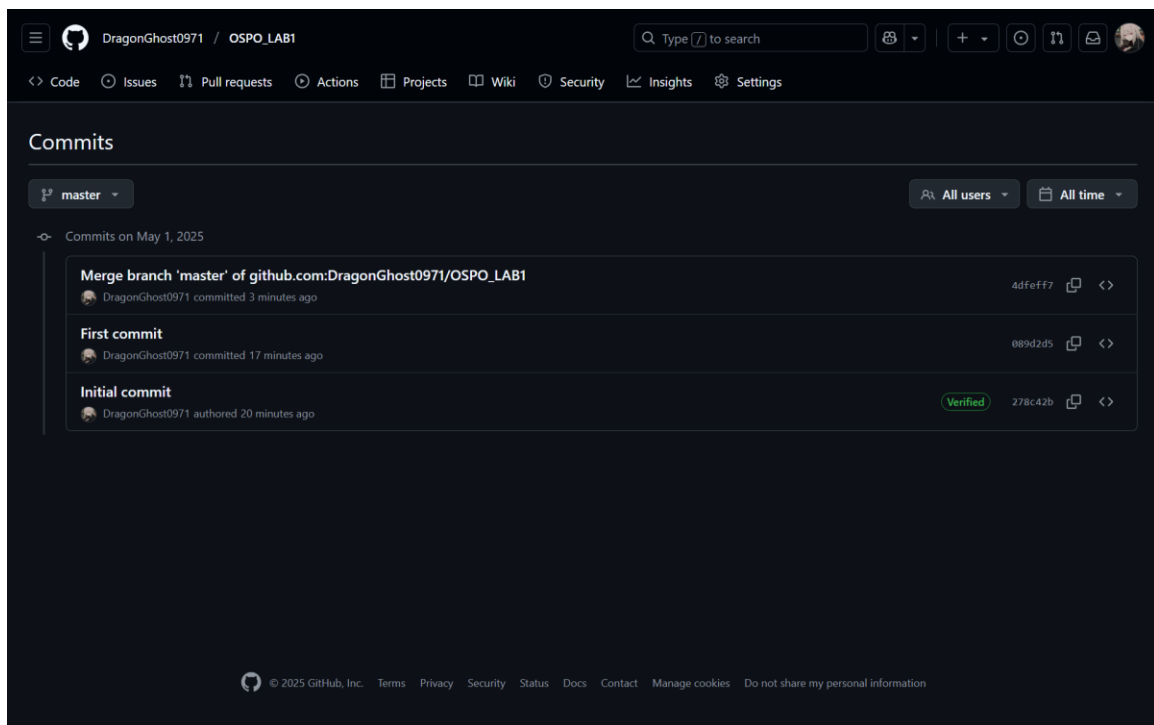
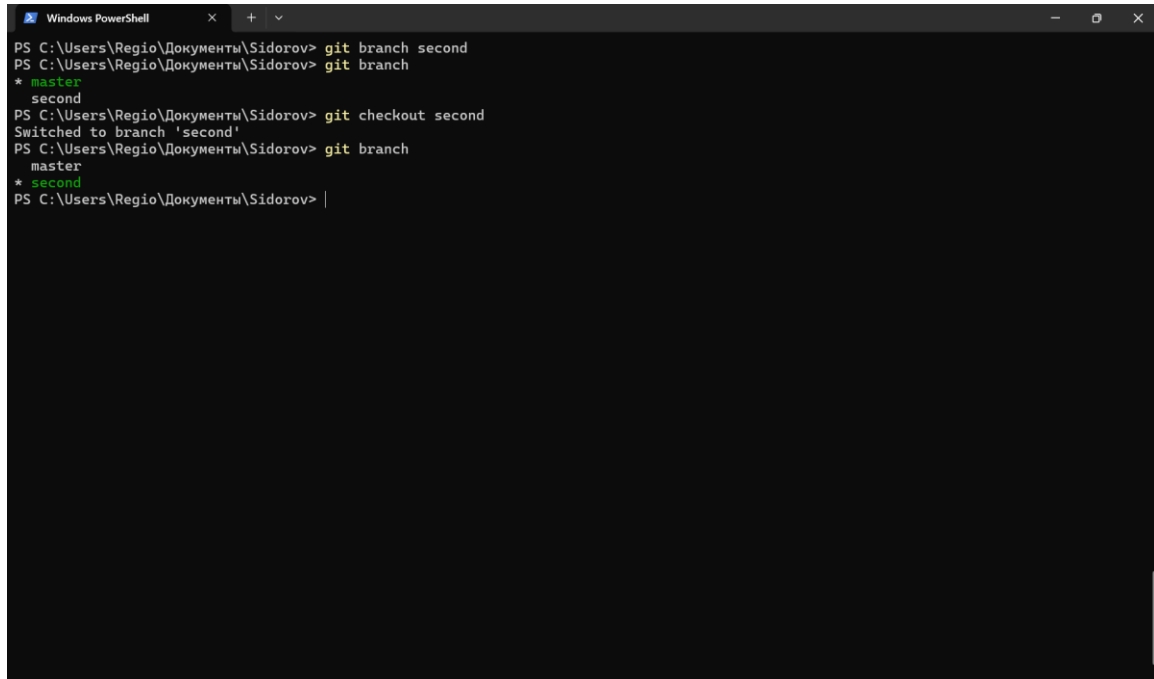


Рис. 8 - Коммиты удалённого репозитория

2.7 Работа с ветками

Создание новой ветки:

```
git branch second
git checkout second
git add .
git commit -m "add test_new.txt"
git push -u origin second
```



```
Windows PowerShell
PS C:\Users\Regio\Документы\Sidorov> git branch second
PS C:\Users\Regio\Документы\Sidorov> git branch
* master
  second
PS C:\Users\Regio\Документы\Sidorov> git checkout second
Switched to branch 'second'
PS C:\Users\Regio\Документы\Sidorov> git branch
  master
* second
PS C:\Users\Regio\Документы\Sidorov> |
```

Рис. 9- Создание ветки second

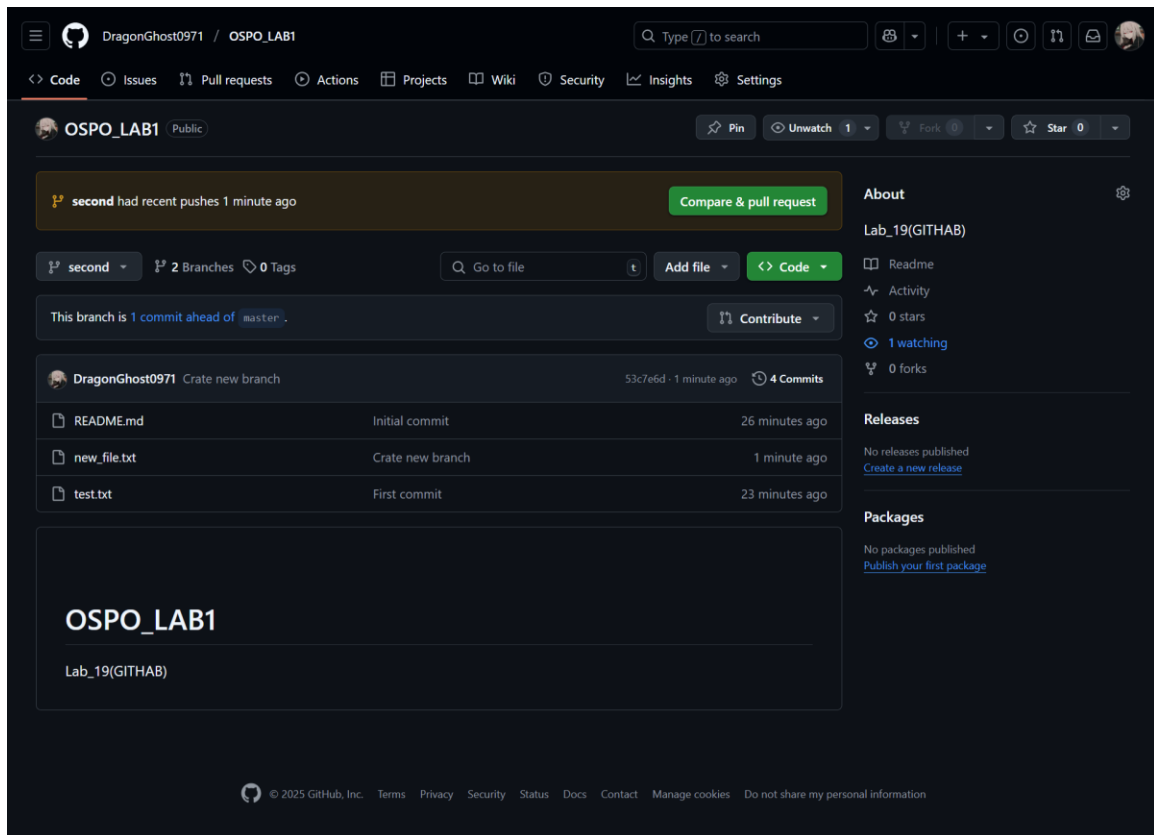


Рис. 10 - ветка second на удалённом репозитории

Слияние и удаление ветки:

```
git checkout master
git merge second
git push origin master
git branch -d second
git push origin --delete second
```

```
Windows PowerShell
PS C:\Users\Regio\Документы\Sidorov> git merge second
Updating 4dfeff7..53c7e6d
Fast-forward
 new_file.txt | Bin 0 -> 20 bytes
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 new_file.txt
PS C:\Users\Regio\Документы\Sidorov> dir

Каталог: C:\Users\Regio\Документы\Sidorov

Mode                LastWriteTime         Length Name
----                -
-a----            01.05.2025   18:17             20 new_file.txt
-a----            01.05.2025   18:05             29 README.md
-a----            01.05.2025   17:43             20 test.txt

PS C:\Users\Regio\Документы\Sidorov> |
```

Рис. 11 - Слияние second с master веткой

```
Windows PowerShell
PS C:\Users\Regio\Документы\Sidorov> git branch -d second
Deleted branch second (was 53c7e6d).
PS C:\Users\Regio\Документы\Sidorov> git branch
* master
PS C:\Users\Regio\Документы\Sidorov> |
```

Рис. 12 - Удаление second ветки

Конфликт с веткой newdev и его разрешение:

```
<<<<<< HEAD
Ivanovichii
=====
```

Ivano

>>>>>> newdev

```
Windows PowerShell
PS C:\Users\Regio\Документы\Sidorov> git merge newdev
Auto-merging newfile.txt
CONFLICT (content): Merge conflict in newfile.txt
Automatic merge failed; fix conflicts and then commit the result.
PS C:\Users\Regio\Документы\Sidorov> |
```

```
Windows PowerShell
PS C:\Users\Regio\Документы\Sidorov> git diff
diff --cc newfile.txt
index f6f973b,f2c0cf4..0000000
--- a/newfile.txt
+++ b/newfile.txt
@@@ -1,1 -1,1 +1,5 @@@
- Vladimirovichass
-Vladimirov
+<<<<<<< HEAD
++Vladimirovichass
+=====
++Vladimirov
+>>>>>>> newdev
PS C:\Users\Regio\Документы\Sidorov> |
```

2.8 Откаты изменений

git log

git checkout hash

git checkout master

git reset HEAD

```
Windows PowerShell
commit ccbba6d7a85b408872d3e3efccf4e6597065b5e4 (HEAD -> master)
Author: DragonGhost0971 <generals86sur@gmail.com>
Date: Thu May 1 21:11:28 2025 +0300

AA

commit b17f25aee251f883044091090bbc651448be803c
Merge: bfb5965 53c7e6d
Author: DragonGhost0971 <generals86sur@gmail.com>
Date: Thu May 1 21:08:31 2025 +0300

Merge branch 'second' of github.com:DragonGhost0971/OSPO_LAB1

commit bfb5965f219a307ab6c03005449891b57e451aa3 (origin/master, origin/HEAD)
Merge: 0f26112 de08b1d
Author: DragonGhost0971 <generals86sur@gmail.com>
Date: Thu May 1 21:04:44 2025 +0300

Finsh

commit 0f2611243bb2b64720978d625062be60c226254d
Author: DragonGhost0971 <generals86sur@gmail.com>
Date: Thu May 1 20:59:01 2025 +0300

Add word

commit de08b1d5879f09a76fca6f1cdca76c5ac91796a0
Author: DragonGhost0971 <generals86sur@gmail.com>
Date: Thu May 1 20:57:59 2025 +0300

Delete word

commit 855949f7a8bd47e478f0becc093096dcae85c09f
Author: DragonGhost0971 <generals86sur@gmail.com>
Date: Thu May 1 20:56:55 2025 +0300

:|
```

Рис. 13 - Просмотр коммитов

```
Windows PowerShell
PS C:\Users\Regio\Документы\Sidorov> git checkout de08b1d5879f09a76fca6f1cdca76c5ac91796a0
Note: switching to 'de08b1d5879f09a76fca6f1cdca76c5ac91796a0'.

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 de08b1d Delete word
PS C:\Users\Regio\Документы\Sidorov> git branch
* (HEAD detached at de08b1d)
  master
PS C:\Users\Regio\Документы\Sidorov> |
```

Рис. 14 - Перейти к коммиту по hash

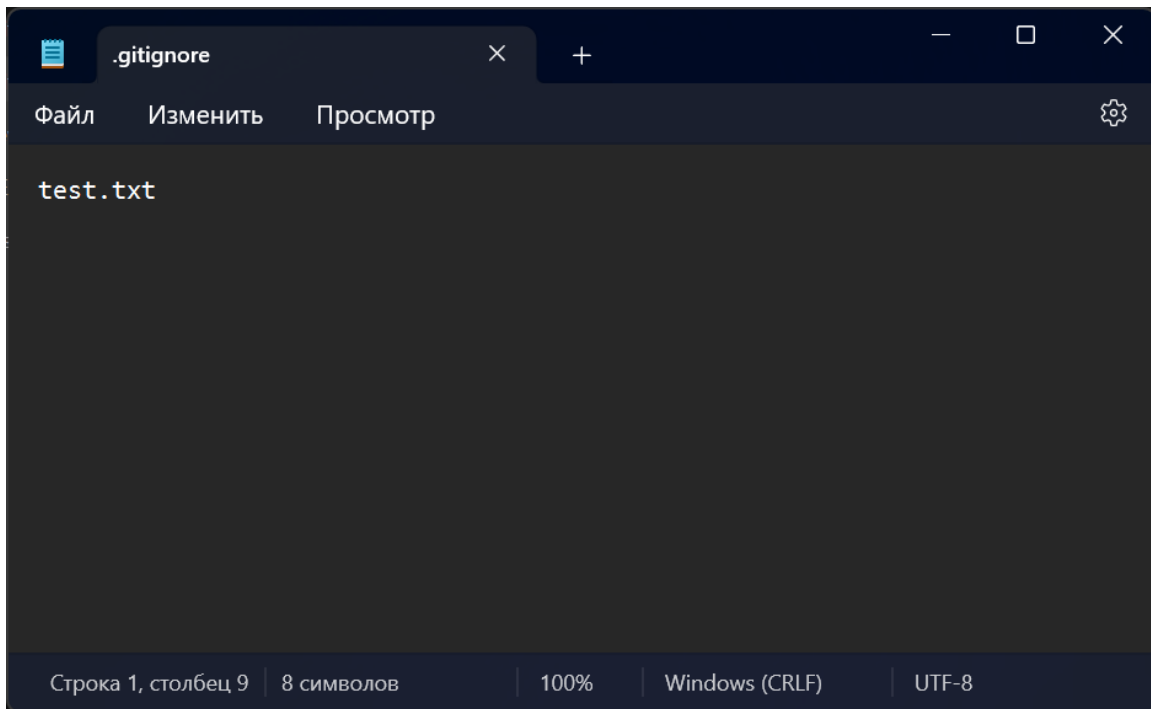


Рис. 17 - Файлы которые необходимо игнорировать

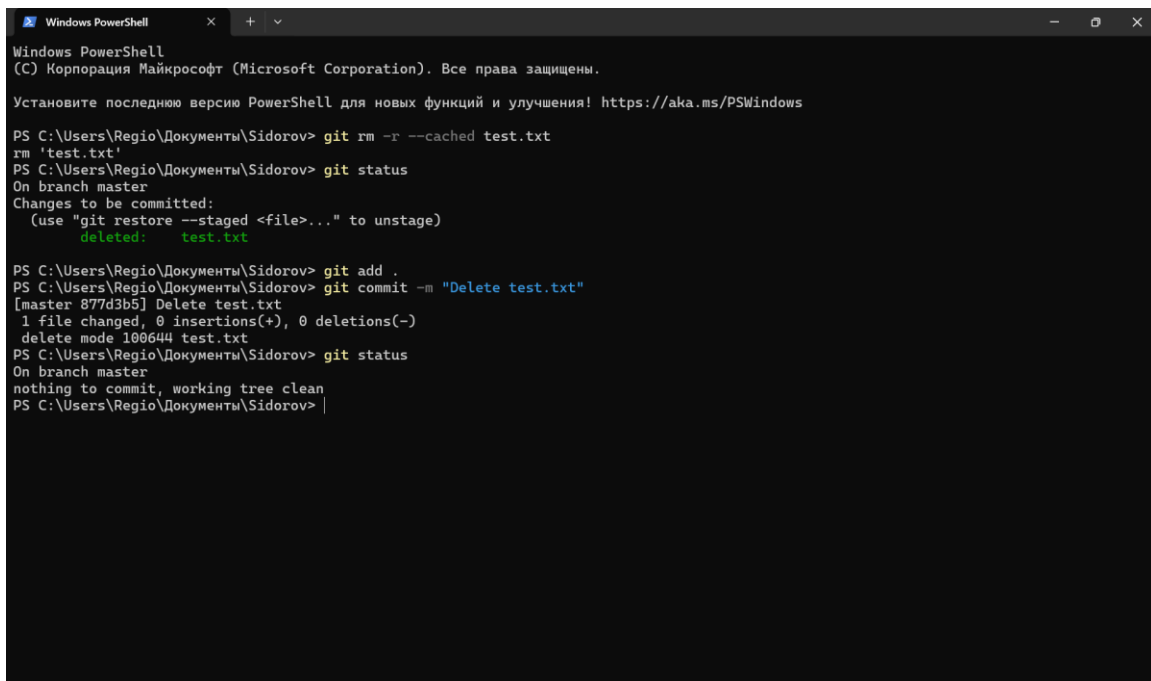


Рис. 18 - Предварительно нужно удалить этот файл из отслеживаемых

```
git add .
git commit -m "add .gitignore"
git push
```


2.10 Совместная работа (Fork → Clone → Pull Request)

1. Напарник сделал Fork моего репозитория.
2. Я выдал ему права Write в Settings → Collaborators.
3. Каждый из нас выполнил `git clone` своего URL и внёс изменения.
4. После `git push` напарник оформил Pull Request; я провёл ревью и выполнил Merge.

4. Результаты

- Создано и синхронизировано два репозитория (локальный + GitHub).
- Освоены базовые и продвинутые команды Git (`init`, `add`, `commit`, `push`, `pull`, `branch`, `merge`, `revert`, `reset`, `checkout`).
- Отработаны типичные случаи конфликтов и их ручное разрешение.
- Организована совместная разработка с использованием `fork`, `clone` и `pull request`.
- Реализовано исключение лишних файлов через `.gitignore`.

5. Выводы

В ходе лабораторной работы были получены практические навыки полного жизненного цикла проекта под управлением Git. Освоены как индивидуальные операции (фиксация, откат, ветвление), так и элементы командной разработки (`fork`, `merge-conflict resolution`).