



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ (ФИЛИАЛ)
ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО
ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
В Г. ТАГАНРОГЕ РОСТОВСКОЙ ОБЛАСТИ
ПИ (филиал) ДГТУ в г. Таганроге

ОТЧЕТ

По дисциплине (модулю) **Перспективные информационные технологии**
(наименование учебной дисциплины (модуля))

на тему: «Инструменты моделирования ИОТ системы: распределённая система контроля версий Git»

Выполнил обучающийся: _____ Кононенко Д. И.
(Ф.И.О.)

Направление:

09.03.02

Информационные системы и технологии (наименование)

Обозначение отчета 2253218 Группа ВО ИСиТ-4122
номер зачетки

Проверил Доцент
должность Орда-Жигулина М.В.
(Ф.И.О.)

Отчет защищён _____ **дата** _____ **оценка** _____ **подпись** _____

Таганрог
2025

Лабораторная работа №4.1

«Инструменты моделирования ИОТ системы: распределённая система контроля версий Git»

Цель лабораторной работы:

Познакомиться с инструментом Git, создать свой репозиторий на GitHub, где будут храниться все выполненные лабораторные работы, и освоить базовые операции: создание репозитория, загрузку файлов, клонирование, внесение изменений и синхронизацию с удалённым репозиторием.

1. Создание репозитория и работа с файлами через веб-интерфейс github
 - 1.1. Создание репозитория
 1. Перешел на сайт GitHub и вошел в свой аккаунт.
 2. Нажал кнопку "Create new..." в левом верхнем углу для создания нового репозитория.

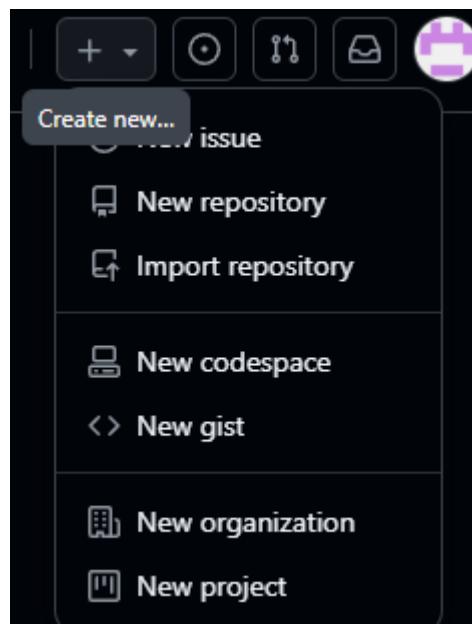


Рисунок 1 – Создание нового...

Изм.	Лист	№ докум.	Подпись	Дат	ВО ИСиТ-4122				
Разраб.	Кононенко Д. И				Перспективные информационные технологии	Лим.	Лист	Листов	
Провер.	Орда-Жигулina							2	
Н. Контр.	Орда-Жигулina							10	
Утвёрд.	Орда-Жигулina					ПИ (филиал) ДГТУ в г. Таганроге			

3. Заполнил основные поля:

- Repository name: STUDY_FOG_group_Ivanov_I_I. В названии на GitHub не отображается кириллица, поэтому записал название на латинице.
- Description: "Репозиторий для отчётов лабораторных работ группа ФИО по проекту STUDY_FOG".
- Выбрал видимость репозитория: Public.

Публичный (Public) – любой пользователь в интернете может просматривать этот репозиторий. Вы выбираете, кто может вносить изменения.

Приватный (Private) – Вы выбираете, кто может просматривать и вносить изменения в этот репозиторий.
- Поставил галочку "Add a README file".

4. Нажал "Create repository".

The screenshot shows the GitHub 'Create repository' interface. The 'General' tab is selected, indicated by a blue circle with the number '1'. The 'Owner' dropdown is set to 'Sinrain1337'. The 'Repository name' field contains 'STUDY_FOG_group_Ivanov_I_I', with a note below it stating 'STUDY_FOG_group_Ivanov_I_I is available.' A link to 'supreme-giggle?' is provided as a suggestion. The 'Description' field contains the text 'Репозиторий для отчётов лабораторных работ группы ФИО по проекту STUDY_FOG'. Below the description, it says '74 / 350 characters'. The 'Configuration' tab is partially visible, indicated by a blue circle with the number '2'. It includes sections for 'Choose visibility' (set to 'Public'), 'Add README' (switched 'On'), 'Add .gitignore' (set to 'No .gitignore'), and 'Add license' (set to 'No license'). At the bottom right is a large green 'Create repository' button.

Рисунок 2 – Создание репозитория.

Изм.	Лист	№ докум.	Подпись	Дата

УП. 180000.000

Лист

3

1.2. Выгрузка материалов через веб-интерфейс

- В созданном репозитории нажал кнопку "Add file" → "Upload files".

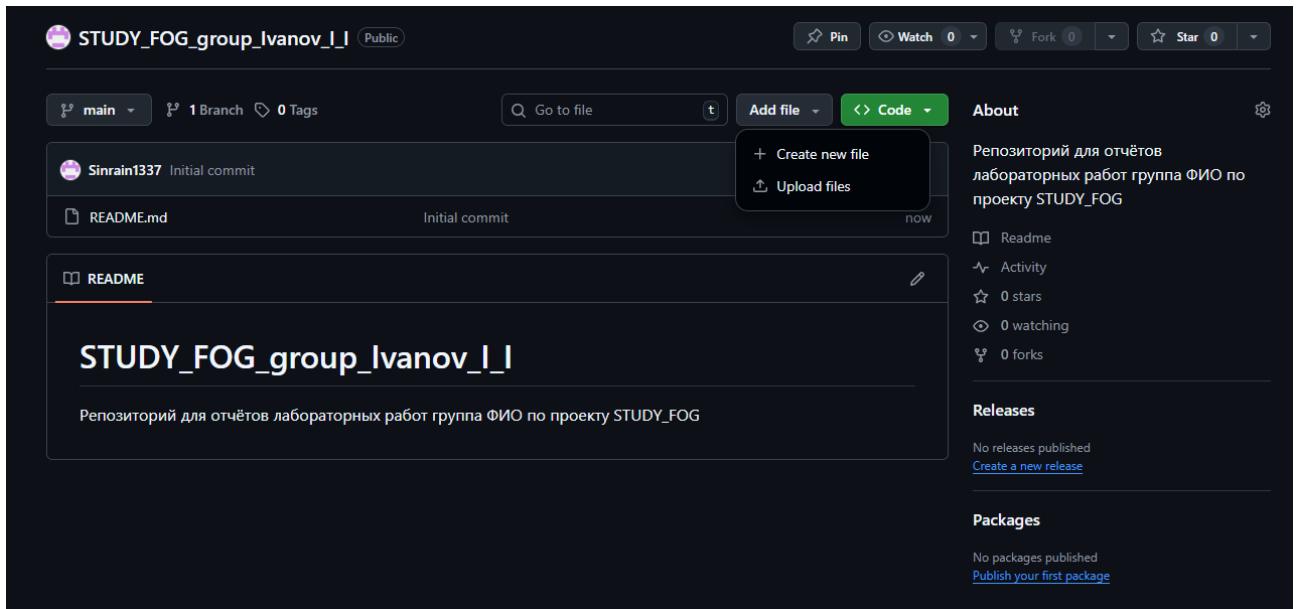


Рисунок 3 – Загрузка файлов.

- Перетащил папки (директории) для загрузки:

- Для каждого отчета и кода в соответствии с выполненными лабораторными работами создал и добавил директории с названиями Lab_3_1, Lab_3_2, Lab_3_3
- Отчёты по лабораторным работам (в формате .docx или .pdf).
- Файлы с кодом (fog_standard.py, viz_cloud_fog_edge_pipeline.py).
- Важно: Сначала загрузил исходные (неизменённые) файлы .ру из репозитория SA9Z/STUDY_FOG.

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

УП. 180000.000

Лист

4

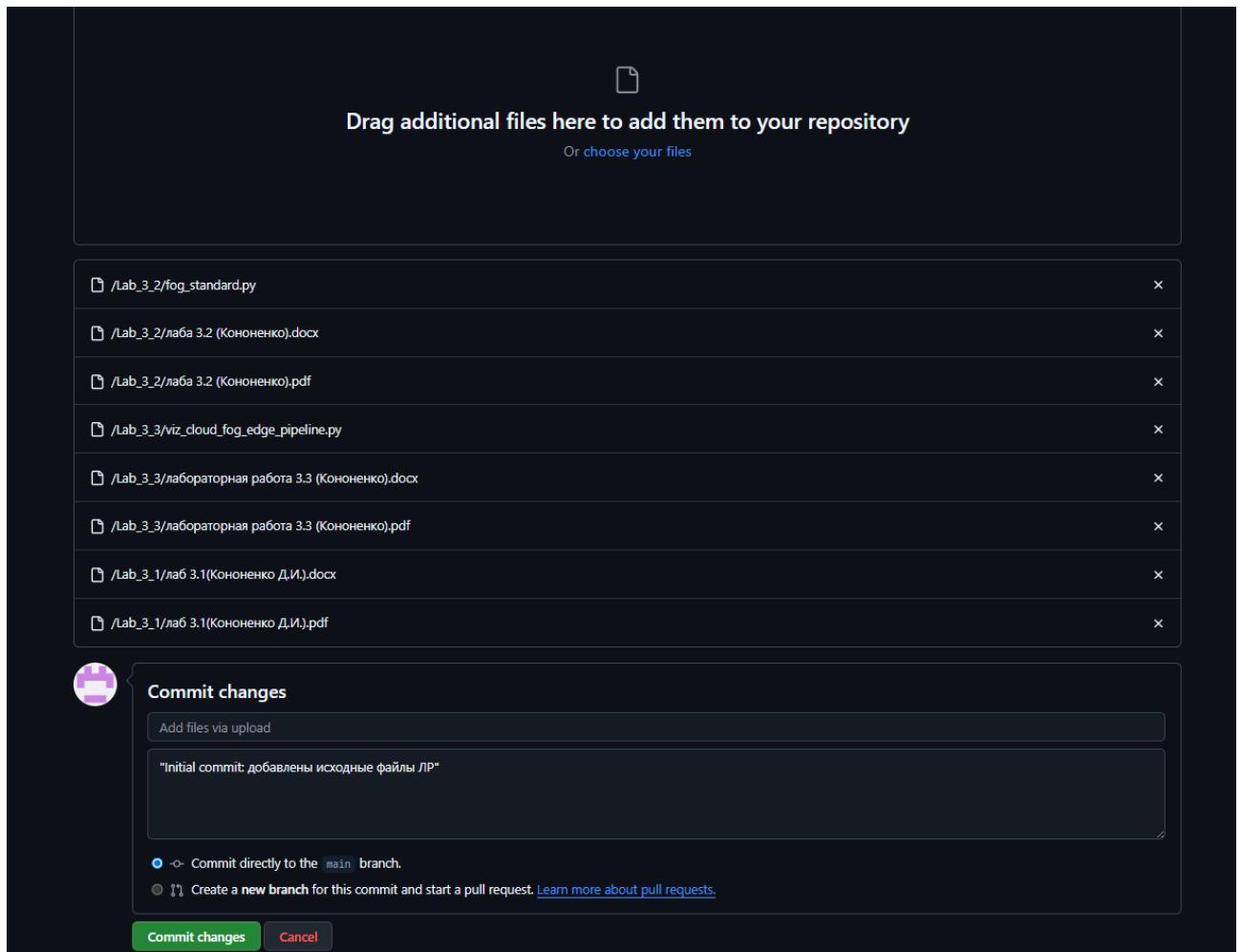


Рисунок 4 – Добавление файлов.

3. Добавил комментарий к коммиту (например, "Initial commit: добавлены исходные файлы ЛР").
4. Нажал "Commit changes".
 - 1.3. Изменение файлов через веб-интерфейс
 1. Перешел в файл fog_standard.py
 2. Нажал на иконку карандаша ("Edit this file").

Изм.	Лист	№ докум.	Подпись	Дата

УП. 180000.000

Лист
5

```

1 """
2 viz_phone_pipeline_bilingual.py
3
4 Бианглийная визуализация конвейера: датчик + Fog + Курьер + Телефон.
5 Bilingual visualization of the pipeline: Sensor + Fog + Courier + Phone
6
7 Показываем / не показываем:
8     • Сквозную задержку "от датчика до телефона" для каждой задачи
9     • End-to-end latency "sensor + phone" per task.
10    • Размер буфера телефона по мере прибывания сообщений.
11    Phone buffer size as messages arrive.
12    • В консоли печатаются метрики на русском и английском.
13    Console prints metrics in Russian and English.
14 """
15 import random, statistics
16 import matplotlib.pyplot as plt
17
18 PIPELINE_RU = "датчик + Fog + Курьер + Телефон"
19 PIPELINE_EN = "Sensor + Fog + Courier + Phone"
20
21 def simulate(n_tasks=30, seed=7):
22     random.seed(seed)
23     # Processing times (ms) per stage / Время обработки (мс) на этапе
24     sensor = [random.randint(20, 60) for _ in range(n_tasks)]
25     fog = [random.randint(30, 80) for _ in range(n_tasks)]
26     courier = [random.randint(10, 40) for _ in range(n_tasks)]
27
28     # End-to-end latency per task is the sum of stage times!
29     # Сквозная задержка на задачу - это сумма времени этапов!

```

Commit changes

Commit message

Update fog_standard.py

Extended description

Add an optional extended description...

Commit directly to the main branch
 Create a new branch for this commit and start a pull request [Learn more about pull requests](#)

Commit changes

Рисунок 5 – Изменение файла.

3. Заменил содержимое файла на измененную версию согласно лабораторной работе №2 (уже выполненную).

4. Нажал кнопку «Commit changes...» и введите комментарий к коммиту.

Укажите каким изменениям подвергся файл, а также выберете создание функции New Branch (Новая Ветка).

Изм.	Лист	№ докум.	Подпись	Дата

УП. 180000.000

Лист

6

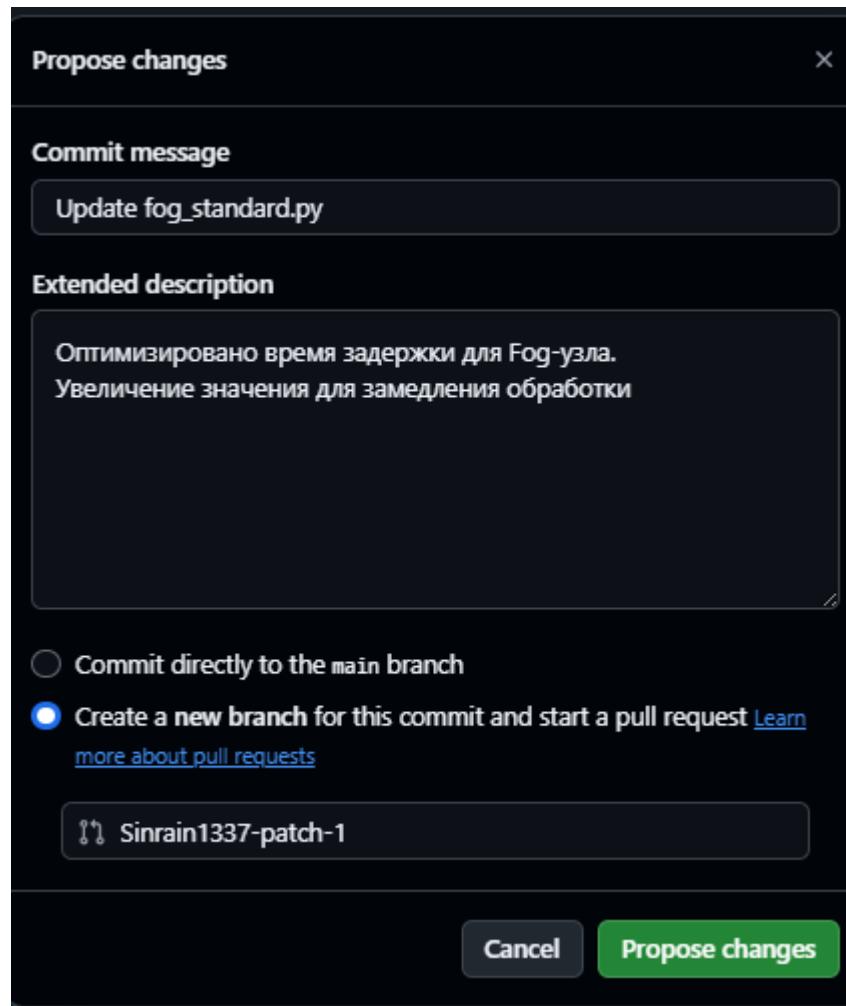


Рисунок 6 – Коммит измененного файла.

5. Нажал "Propose changed".

Теперь работа сохранена в изолированной среде, и можно в любой момент вернуться к предыдущим состояниям файла. Вот скриншот с изменениями в файле:

```

Showing 1 changed file with 1 addition and 1 deletion.

Lab_3_2/fog_standard.py
@@ -22,7 +22,7 @@ def simulate(n_tasks=30, seed=7):
    random.seed(seed)
    # Processing times (ms) per stage / времена обработки (мс) на каждом этапе:
    sensor = [random.randint(20, 60) for _ in range(n_tasks)] # Sensor / датчик
-    fog = [random.randint(30, 80) for _ in range(n_tasks)] # Fog node / Fog-узел
+    fog = [random.randint(3, 5) for _ in range(n_tasks)] # Fog node / Fog-узел
    courier = [random.randint(10, 40) for _ in range(n_tasks)] # Courier / Курьер
    # End-to-end latency per task is the sum of stage times:

```

Рисунок 7 – Измененный файл.

Изм.	Лист	№ докум.	Подпись	Дата

УП. 180000.000

Лист

7

6. Создал «Pull request» и выполните функцию «Merge pull request». Теперь ваша основная ветка main изменена. Откройте код на странице CODE в вашем репозитории и проверьте изменения.

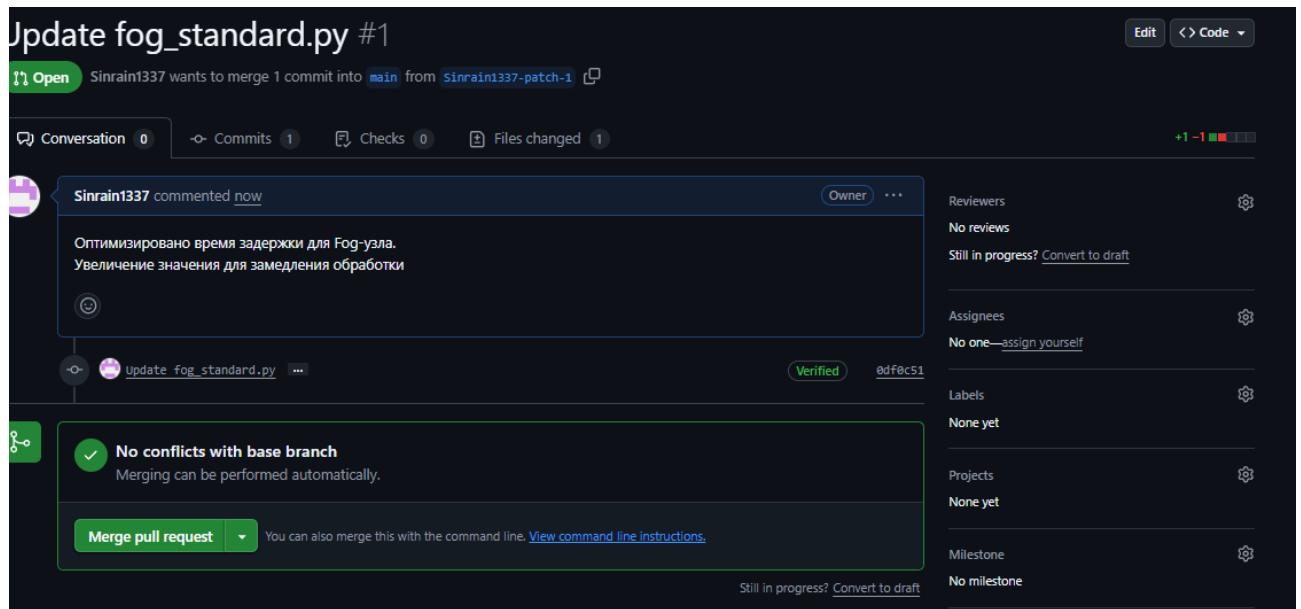


Рисунок 8 – Pull request.

2. РАБОТА С РЕПОЗИТОРИЕМ ЧЕРЕЗ КОМАНДНУЮ СТРОКУ

2.1. Клонирование репозитория

1. Установил Git bash и VS Code, если они ещё не установлены.
2. Открыл терминал в VS Code.
3. Выполнил команду для клонирования репозитория:

```
>> git clone https://github.com/SA9Z/STUDY_FOG_group_Ivanov_I_I.git
```

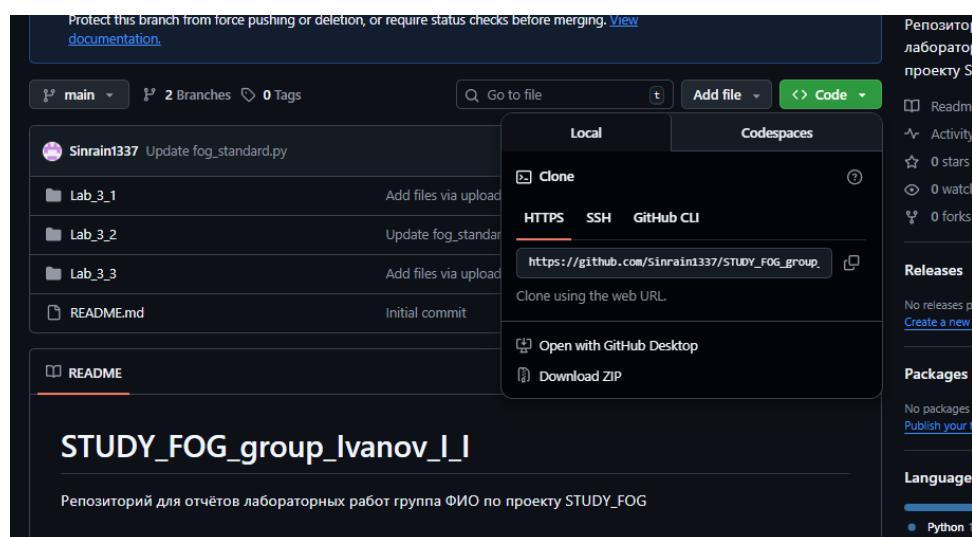


Рисунок 9 – Копирование ссылки проекта.

Изм.	Лист	№ докум.	Подпись	Дата

УП. 180000.000

Лист

8

4. Перешел в папку проекта:

```
>> cd STUDY_FOG_group_Ivanov_I_I
```

```
PS C:\Users\verbe\Desktop\пары\курс 4\Семестр 7\Перспективные информационные технологии> Git clone https://github.com/Sinrain1337/STUDY_FOG_group_Ivanov_I_I.git
Cloning into 'STUDY_FOG_group_Ivanov_I_I'...
remote: Enumerating objects: 23, done.
remote: Counting objects: 100% (23/23), done.
remote: Compressing objects: 100% (22/22), done.
remote: Total 23 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (23/23), 5.63 MiB | 3.09 MiB/s, done.
Resolving deltas: 100% (2/2), done.
PS C:\Users\verbe\Desktop\пары\курс 4\Семестр 7\Перспективные информационные технологии> cd .\STUDY_FOG_group_Ivanov_I_I\>
PS C:\Users\verbe\Desktop\пары\курс 4\Семестр 7\Перспективные информационные технологии\STUDY_FOG_group_Ivanov_I_I>
```

Рисунок 10 – Клонирование репозитория.

2.2. Внесение изменений и отправка на GitHub

1. Открыл файл viz_cloud_fog_edge_pipeline.py в VS Code и заменил его содержимое на готовую версию в соответствии с вариантом.
2. Проверил статус репозитория:

```
>> git status
```

Можно увидеть список изменённых файлов:

```
PS C:\Users\verbe\Desktop\пары\курс 4\Семестр 7\Перспективные информационные технологии\STUDY_FOG_group_Ivanov_I_I\Lab_3_3> 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:   viz_cloud_fog_edge_pipeline.py

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

Рисунок 11 – Список измененных файлов.

3. Добавил изменённый файл в индекс:

```
>> git add .viz_cloud_fog_edge_pipeline.py
```

```
_3> git add .\viz_cloud_fog_edge_pipeline.py
```

Рисунок 12 – Добавил измененный файл.

Изм.	Лист	№ докум.	Подпись	Дата

УП. 180000.000

Лист

9

4. Создал коммит с комментарием:

```
>> git commit -m "Измененный файл по варианту 7"
```

```
PS C:\Users\verbe\Desktop\пары\курс 4\Семестр 7\Перспективные информационные технологии\STUDY_FOG_group_Ivanov_I_I\Lab_3_3> git commit -m "изменил файл по варианту 7"
[main 850d456] изменил файл по варианту 7
 1 file changed, 3 insertions(+), 3 deletions(-)
```

Рисунок 13 – Зафиксировал изменение.

5. Отправил изменения на GitHub:

```
>> git push origin main
```

```
PS C:\Users\verbe\Desktop\пары\курс 4\Семестр 7\Перспективные информационные технологии\STUDY_FOG_group_Ivanov_I_I\Lab_3_3> git push origin main
info: please complete authentication in your browser...
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 491 bytes | 491.00 KiB/s, done.
Total 4 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/Sirrain1337/STUDY_FOG_group_Ivanov_I_I.git
 2a2392a..850d456 main -> main
```

Рисунок 13 – Отправка изменений.

Почему работа через командную строку эффективнее?

- Автоматизация: Скрипты и пакетные операции выполняются быстрее.
- Гибкость: Возможность использовать сложные сценарии (например, слияние веток, откат изменений).
- Локальная работа: Все операции, кроме отправки и загрузки, выполняются offline.
- Полный контроль: Вы сами управляете каждым этапом, что полезно для понимания процесса.
- После выполнения git push обновите страницу репозитория на GitHub — вы увидите уведомление о новом коммите и изменённых файлах.

Вывод: В ходе лабораторной работы я познакомился с инструментом Git, создал свой репозиторий на GitHub, где будут храниться все выполненные лабораторные работы, и освоил базовые операции: создание репозитория, загрузку файлов, клонирование, внесение изменений и синхронизацию с удалённым репозиторием.

Изм.	Лист	№ докум.	Подпись	Дата

УП. 180000.000

Лист

10