

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 2

дисциплина: *Архитектура компьютера*

Студент: Довашеев Дмитрий

Группа: НКАбд-07-25

МОСКВА

2025 г.

Оглавление

1 Цель работы.....	3
3 Теоретическое введение	5
4 Выполнение лабораторной работы.....	7
4.1 Техническое обеспечение	7
4.2 Базовая настройка Git	7
4.3 Создание рабочего пространства и репозитория курса на основе шаблона.....	10
4.4 Создание репозитория курса на основе шаблона.	10
4.5 Настройка каталога курса.....	13
5 Задания для самостоятельной работы.....	14
6 Выводы.....	15
Список литературы	16

1 Цель работы

Целью работы является изучить идеологию и применение системы контроля версий. Приобрести практические навыки по работе с системой git.

2 Задание

На основе методических указаний провести работу с базовыми командами системы контроля версий git, выучить применение команд для разных случаев использования, настроить GitHub.

3 Теоретическое введение

Системы контроля версий (Version Control System, VCS) применяются при работе нескольких человек над одним проектом. Обычно основное дерево проекта хранится в локальном или удалённом репозитории, к которому настроен доступ для участников проекта. При внесении изменений в содержание проекта система контроля версий позволяет их фиксировать, совмещать изменения, произведённые разными участниками проекта, производить откат к любой более ранней версии проекта, если это требуется. В классических системах контроля версий используется централизованная модель, предполагающая наличие единого репозитория для хранения файлов. Выполнение большинства функций по управлению версиями осуществляется специальным сервером. Участник проекта (пользователь) перед началом работы посредством определённых команд получает нужную ему версию файлов. После внесения изменений, пользователь размещает новую версию в хранилище. При этом предыдущие версии не удаляются из центрального хранилища и к ним можно вернуться в любой момент. Сервер может сохранять не полную версию изменённых файлов, а производить так называемую дельта-компрессию — сохранять только изменения между последовательными версиями, что позволяет уменьшить объём хранимых данных. Системы контроля версий поддерживают возможность отслеживания и разрешения конфликтов, которые могут возникнуть при работе нескольких человек над одним файлом. Можно объединить (слить) изменения, сделанные разными участниками (автоматически или вручную), вручную выбрать нужную версию, отменить изменения вовсе или заблокировать файлы для изменения. В зависимости от настроек блокировка не позволяет другим пользователям получить рабочую копию или препятствует изменению рабочей копии файла средствами файловой системы ОС, обеспечивая таким образом, привилегированный доступ только одному пользователю, работающему с файлом.

<code>git commit -am</code>	сохранить все добавленные изменения и все изменённые файлы 'Описание коммита'
<code>git checkout -b</code>	создание новой ветки, базирующейся на текущей имя_ветки
<code>git checkout</code>	переключение на некоторую ветку (при переключении на ветку, которой ещё нет в локальном репозитории, она будет создана и связана с удалённой)
<code>git push origin</code>	отправка изменений конкретной ветки в центральный репозиторий имя_ветки
<code>git merge</code>	слияние ветки с текущим деревом
<code>--no-ff</code>	
<code>имя_ветки</code>	
<code>git branch -d</code>	удаление локальной уже слитой с основным деревом ветки имя_ветки
<code>git branch -D</code>	принудительное удаление локальной ветки имя_ветки
<code>git push origin</code>	удаление ветки с центрального репозитория :имя_ветки

Таблица 3.1 Описание некоторых команд системы контроля версий Git.

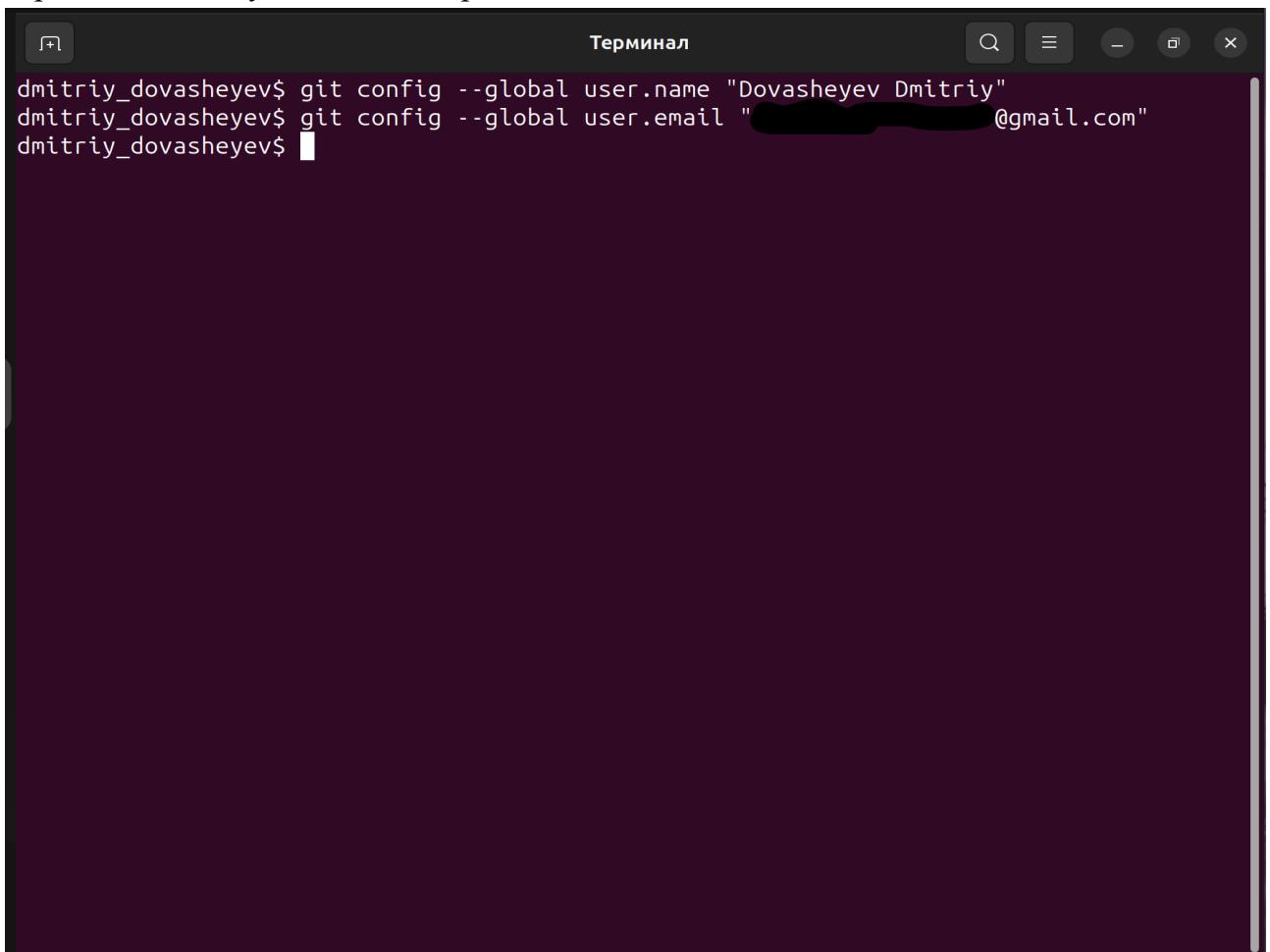
4 Выполнение лабораторной работы

4.1 Техническое обеспечение

Лабораторная работа была выполнена на домашнем компьютере под управлением операционной системы Ubuntu 24.04.3 LTS.

4.2 Базовая настройка Git

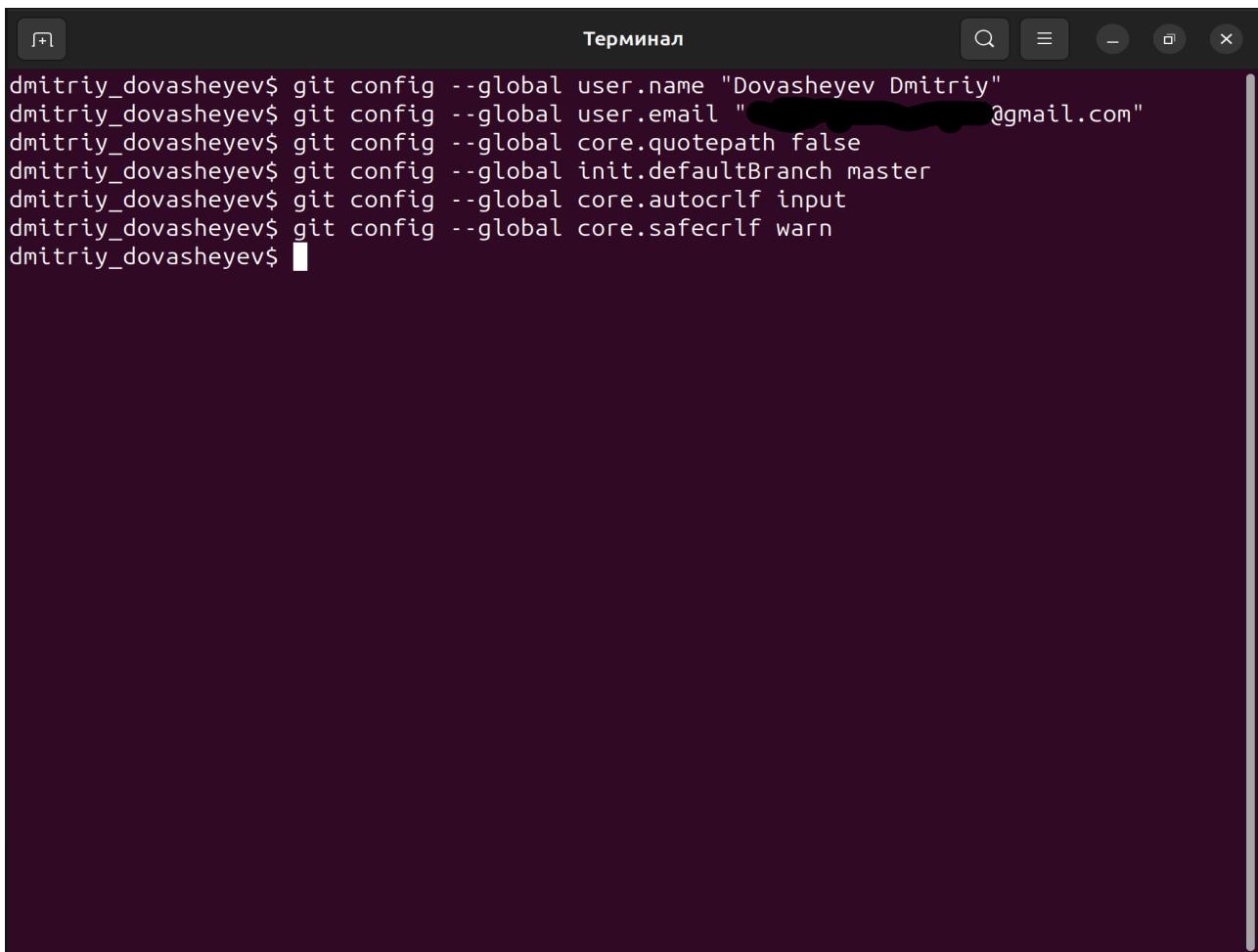
Для начала я проведу предварительную конфигурацию Git, для этого открываю терминал и ввожу команды на (рис. 4.2.1)



```
dmitriy_dovasheyev$ git config --global user.name "Dovasheyev Dmitriy"
dmitriy_dovasheyev$ git config --global user.email "████████@gmail.com"
dmitriy_dovasheyev$
```

Рис. 4.2.1 Предварительная конфигурация Git.

Далее настраиваю параметры utf-8, имя начальной ветки, autocrlf и safecrlf (рис. 4.2.2)

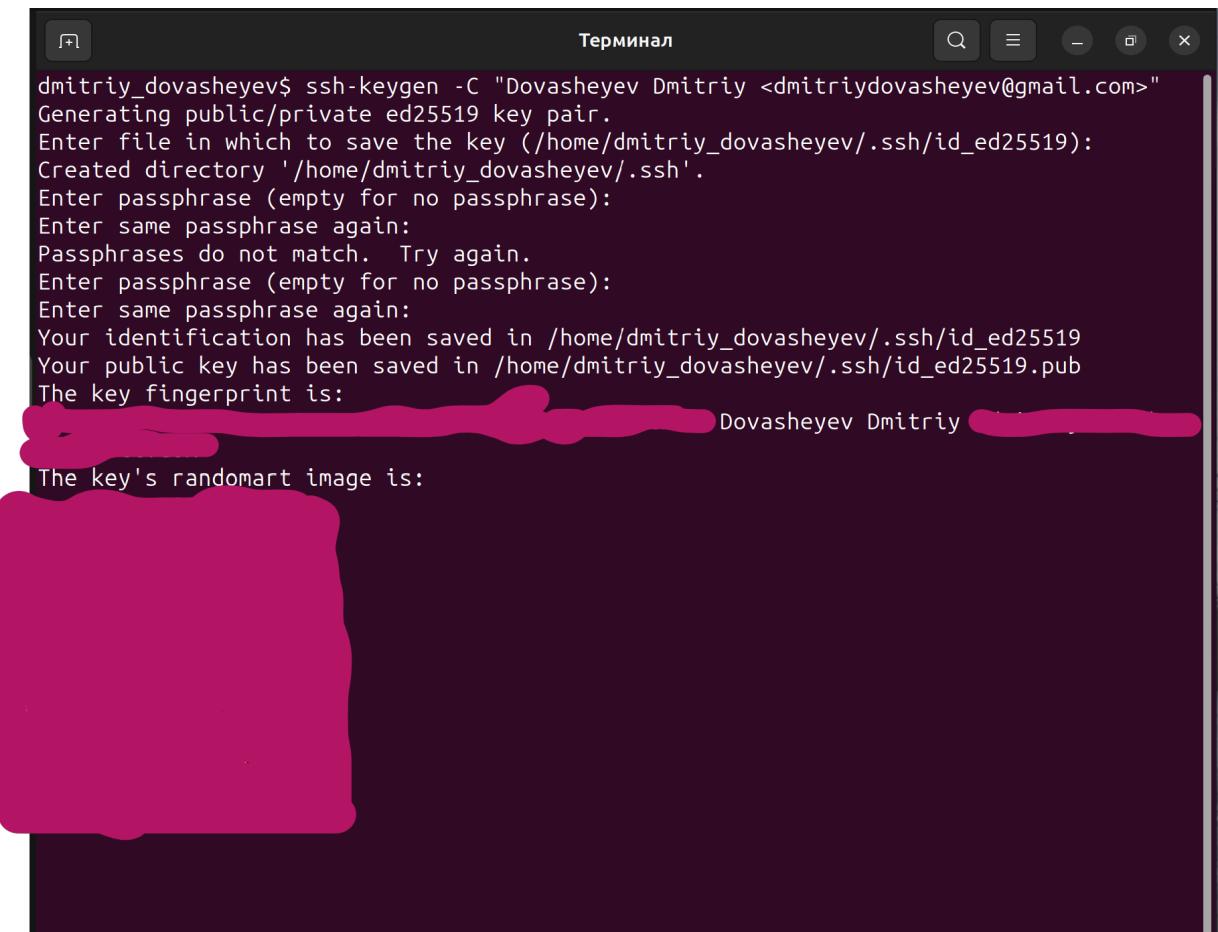


Terminal window titled "Терминал". The command history shows the following configuration commands:

```
dmitriy_dovasheyev$ git config --global user.name "Dovasheyev Dmitriy"
dmitriy_dovasheyev$ git config --global user.email "████████@gmail.com"
dmitriy_dovasheyev$ git config --global core.quotepath false
dmitriy_dovasheyev$ git config --global init.defaultBranch master
dmitriy_dovasheyev$ git config --global core.autocrlf input
dmitriy_dovasheyev$ git config --global core.safecrlf warn
dmitriy_dovasheyev$ █
```

Рис. 4.2.2 Настройка параметров Git.

Далее создаю пару ssh ключей для интеграции с платформой GitHub (рис. 4.2.3)

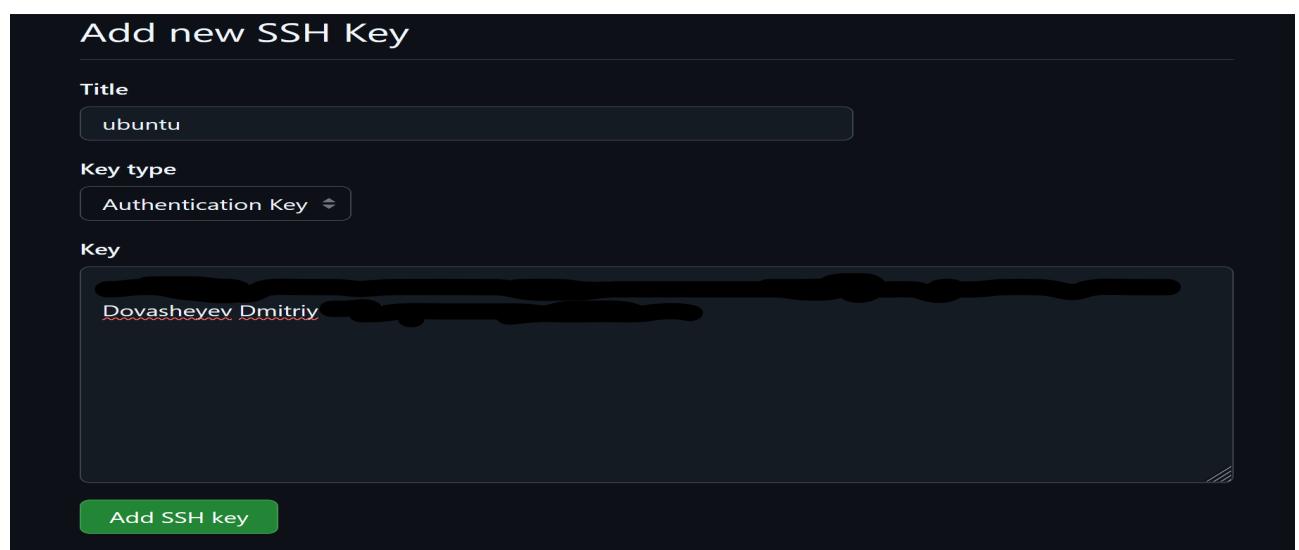


Terminal window showing the command `ssh-keygen -C "Dovasheyev Dmitriy <dmitriydovasheyev@gmail.com>"` being run. The output shows the generation of a public/private key pair, saving the private key to `/home/dmitriy_dovasheyev/.ssh/id_ed25519`, and saving the public key to `/home/dmitriy_dovasheyev/.ssh/id_ed25519.pub`. The key fingerprint is also displayed.

```
dmitriy_dovasheyev$ ssh-keygen -C "Dovasheyev Dmitriy <dmitriydovasheyev@gmail.com>"\nGenerating public/private ed25519 key pair.\nEnter file in which to save the key (/home/dmitriy_dovasheyev/.ssh/id_ed25519):\nCreated directory '/home/dmitriy_dovasheyev/.ssh'.\nEnter passphrase (empty for no passphrase):\nEnter same passphrase again:\nPassphrases do not match. Try again.\nEnter passphrase (empty for no passphrase):\nEnter same passphrase again:\nYour identification has been saved in /home/dmitriy_dovasheyev/.ssh/id_ed25519\nYour public key has been saved in /home/dmitriy_dovasheyev/.ssh/id_ed25519.pub\nThe key fingerprint is:\n[REDACTED] Dovasheyev Dmitriy [REDACTED]\n[REDACTED]\nThe key's randomart image is:\n[REDACTED]
```

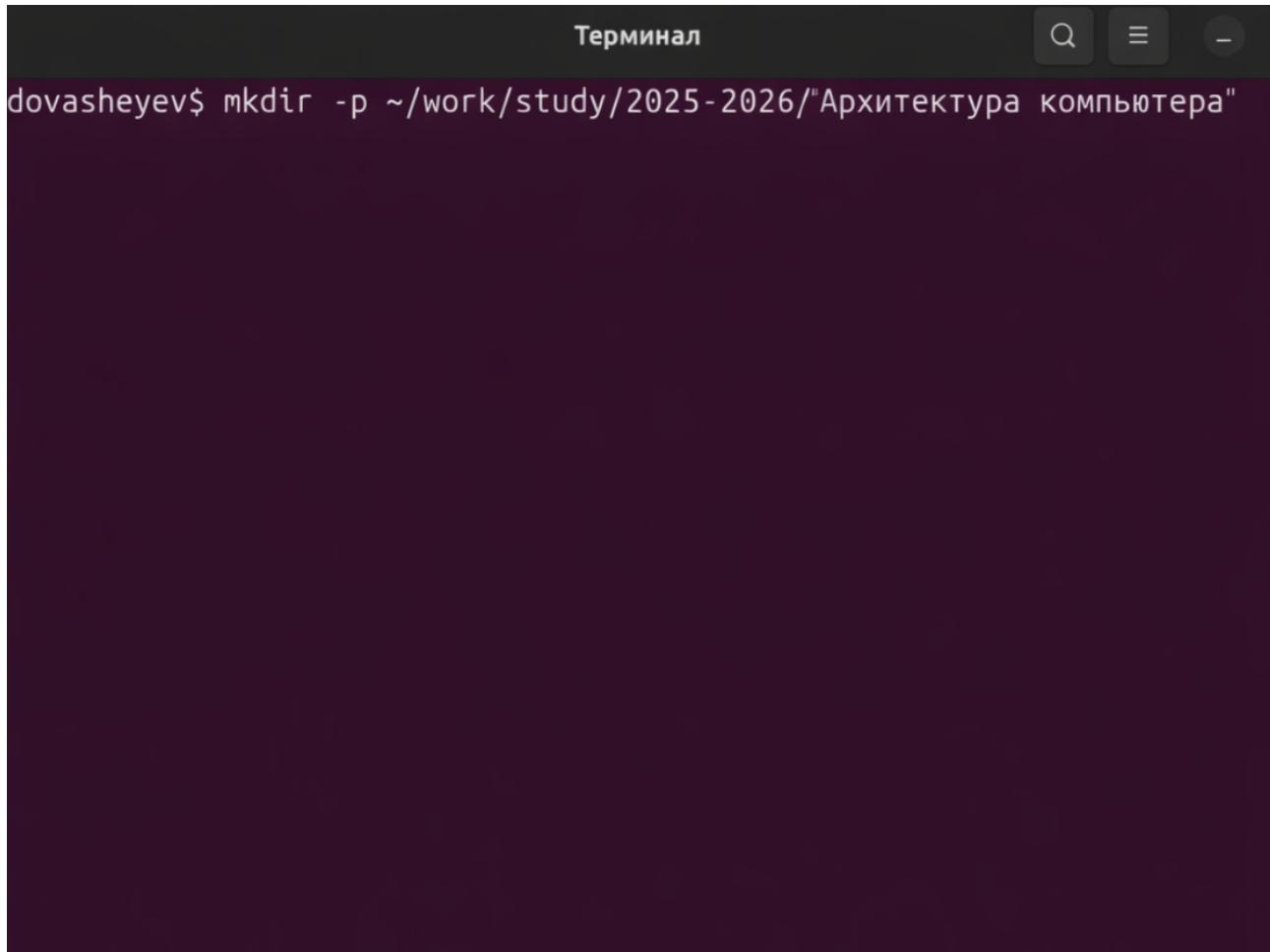
Рис. 4.2.3 Создание пары ssh ключей.

Далее я перехожу на сайт GitHub, авторизуюсь, перехожу в настройки аккаунта, вставляю публичный ключ в предназначенном для этого поле. (рис. 4.2.4)



4.3 Создание рабочего пространства и репозитория курса на основе шаблона.

Рабочее пространство при выполнении лабораторных работ должно придерживаться определённой структурной иерархии, для этого я создаю директорию на своем рабочем компьютере (рис. 4.3.1)



Терминал

```
dovasheyev$ mkdir -p ~/work/study/2025-2026/"Архитектура компьютера"
```

4.4 Создание репозитория курса на основе шаблона.

Создаю репозиторий на основе имеющего шаблона (рис. 4.4.1) через функционал клонирования интерфейса GitHub. (рис 4.4.2)

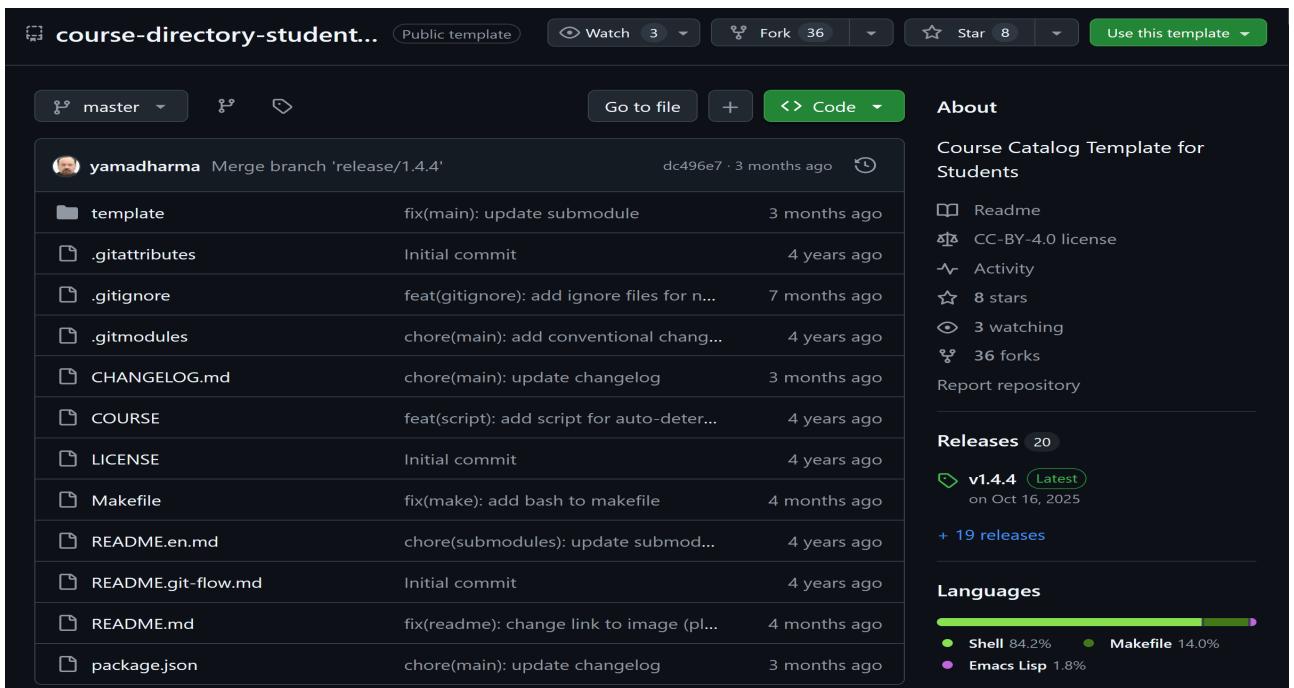


Рис. 4.4.1 Шаблон для клонирования на платформе GitHub.

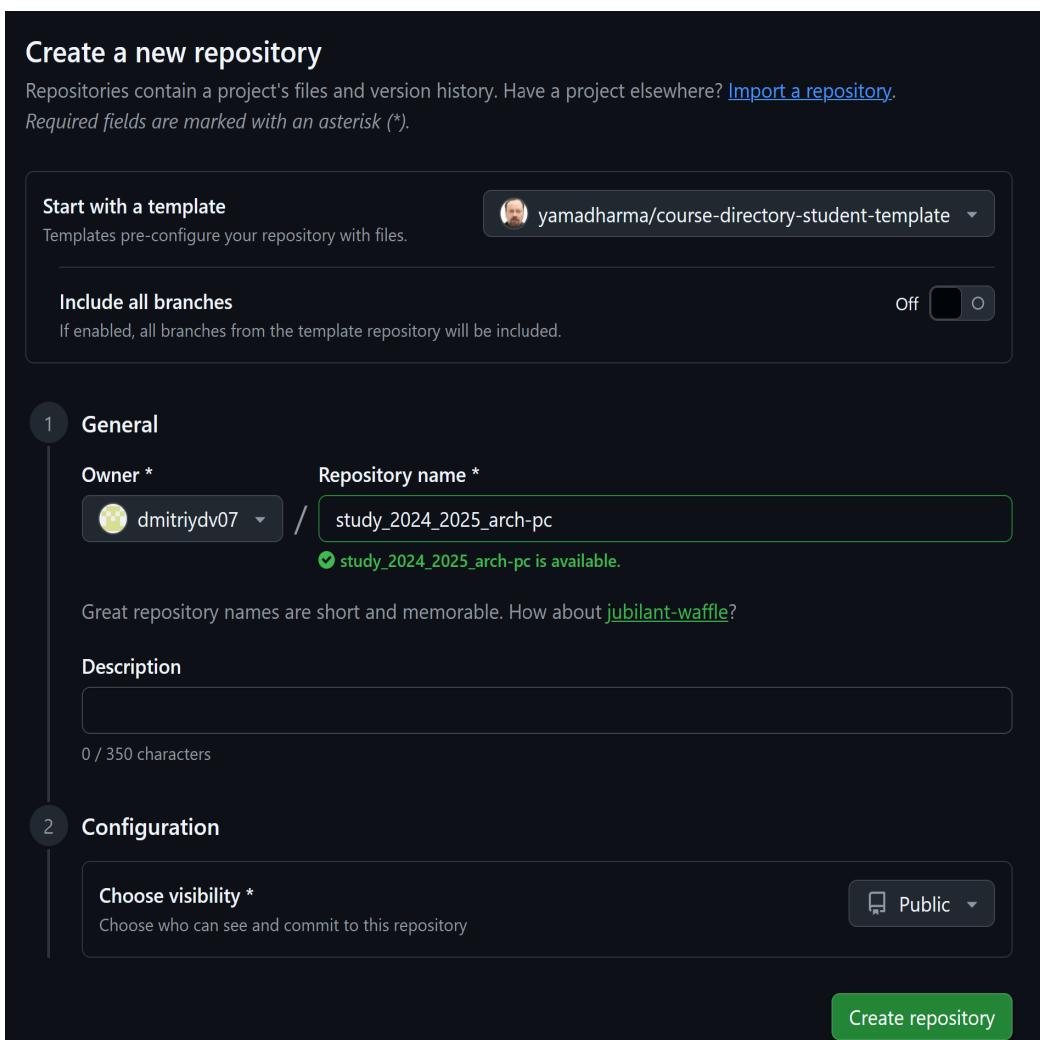


Рис. 4.4.2 Процесс клонирования репозитория через интерфейс GitHub.

Сгенерированный репозиторий на основе шаблона клонирую на свой рабочий компьютер, для этого беру ссылку для клонирования через интерфейс GitHub (рис. 4.4.3) и затем ввожу в терминале git clone. (рис 4.4.4)

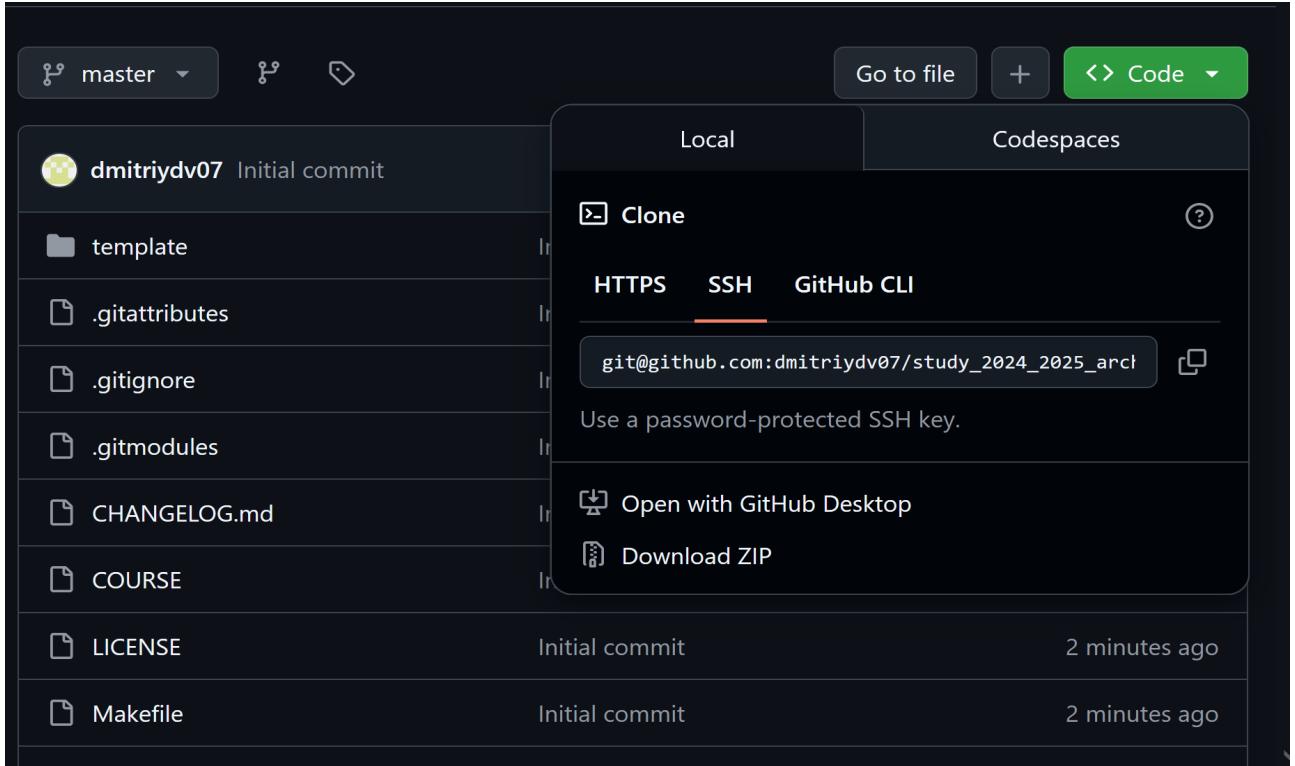


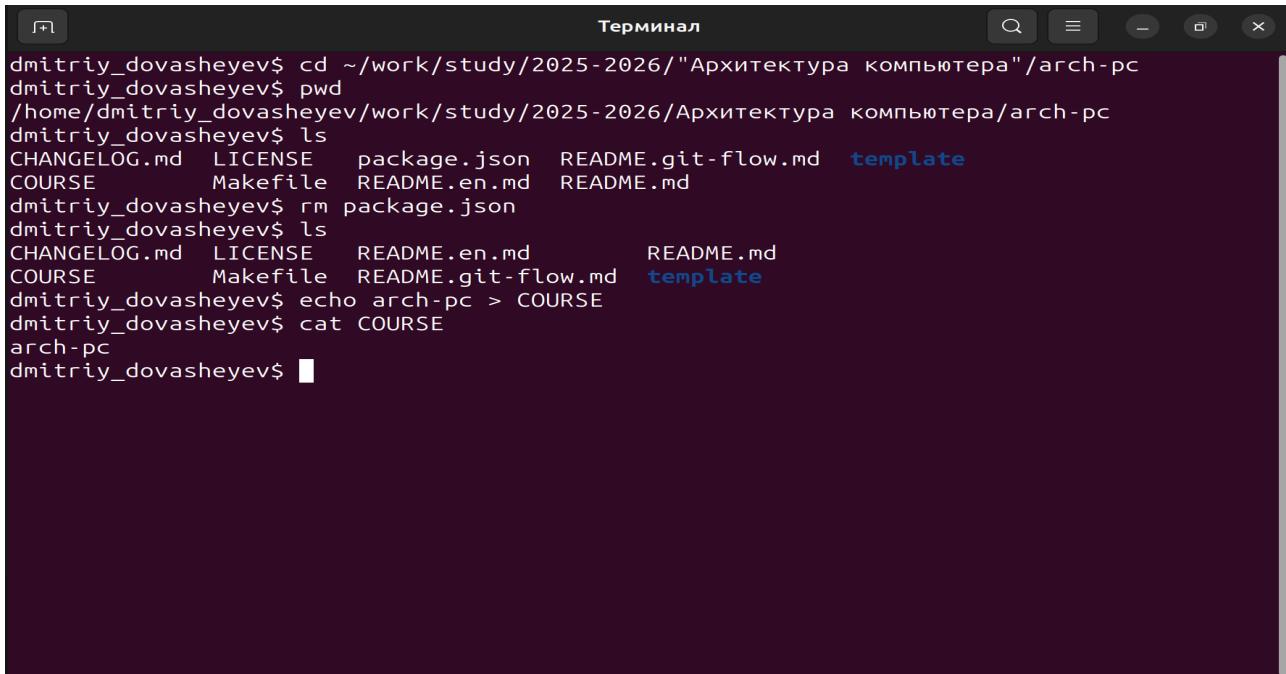
Рис. 4.4.3 Копирование ссылки для последующей вставки в терминал.

```
dmitriy_dovasheyev$ mkdir -p ~/work/study/2025-2026/"Архитектура компьютера"
dmitriy_dovasheyev$ cd ~/work/study/2025-2026/"Архитектура компьютера"
dmitriy_dovasheyev$ pwd
/home/dmitriy_dovasheyev/work/study/2025-2026/Архитектура компьютера
dmitriy_dovasheyev$ git clone --recursive git@github.com:dmitriydv07/study_2025-2026_arch-pc.git arch-pc
Клонирование в «arch-pc»...
The authenticity of host 'github.com (140.82.121.4)' can't be established.
ED25519 key fingerprint is SHA256:DtV3wvvv6TuJjhbpZlsf/zLDA0zPM5vhdkr4UvOquU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com (ED25519)' to the list of known hosts.
remote: Enumerating objects: 38, done.
remote: Counting objects: 100% (38/38), done.
remote: Compressing objects: 100% (36/36), done.
remote: Total 38 (delta 1), reused 26 (delta 1), pack-reused 0 (from 0)
Получение объектов: 100% (38/38), 23.57 Кб | 492.00 Кб/с, готово.
Определение изменений: 100% (1/1), готово.
Подмодуль «template/presentation» (https://github.com/yamadharma/academic-presentation-markdown-template.git) зарегистрирован по пути «template/presentation»
Подмодуль «template/report» (https://github.com/yamadharma/academic-laboratory-report-template.git) зарегистрирован по пути «template/report»
Клонирование в «/home/dmitriy_dovasheyev/work/study/2025-2026/Архитектура компьютера/arch-pc/template/presentation»...
remote: Enumerating objects: 219, done.
remote: Counting objects: 100% (219/219), done.
remote: Compressing objects: 100% (151/151), done.
remote: Total 219 (delta 86), reused 189 (delta 56), pack-reused 0 (from 0)
Получение объектов: 100% (219/219), 2.66 Мб | 4.09 Мб/с, готово.
Определение изменений: 100% (86/86), готово.
Клонирование в «/home/dmitriy_dovasheyev/work/study/2025-2026/Архитектура компьютера/arch-pc/template/report»...
remote: Enumerating objects: 251, done.
remote: Counting objects: 100% (251/251), done.
remote: Compressing objects: 100% (172/172), done.
remote: Total 251 (delta 111), reused 204 (delta 64), pack-reused 0 (from 0)
Получение объектов: 100% (251/251), 775.12 Кб | 2.38 Мб/с, готово.
Определение изменений: 100% (111/111), готово.
Submodule path 'template/presentation': checked out '1c93acf9e731bf186384c85de4aff70037314240'
Submodule path 'template/report': checked out '8ee157c58b3362947b1c71492a65d4dc6882d5ad'
dmitriy_dovasheyev$
```

Рис. 4.4.4 Копирование репозитория на рабочий компьютер.

4.5 Настройка каталога курса.

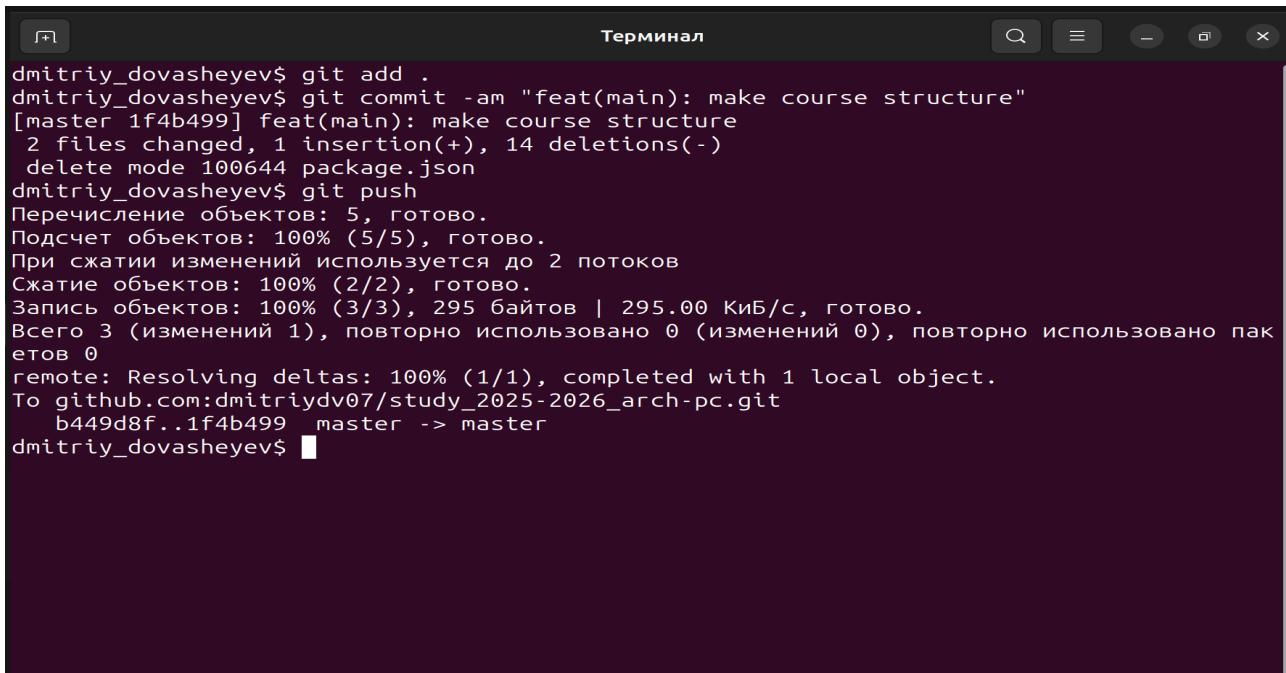
В каталоге курса удаляю лишние файлы и формирую необходимые каталоги (рис. 4.3.1)



```
dmitriy_dovasheyev$ cd ~/work/study/2025-2026/"Архитектура компьютера"/arch-pc
dmitriy_dovasheyev$ pwd
/home/dmitriy_dovasheyev/work/study/2025-2026/Архитектура компьютера/arch-pc
dmitriy_dovasheyev$ ls
CHANGELOG.md LICENSE package.json README.git-flow.md template
COURSE Makefile README.en.md README.md
dmitriy_dovasheyev$ rm package.json
dmitriy_dovasheyev$ ls
CHANGELOG.md LICENSE README.en.md README.md
template
dmitriy_dovasheyev$ echo arch-pc > COURSE
dmitriy_dovasheyev$ cat COURSE
arch-pc
dmitriy_dovasheyev$
```

Рис. 4.3.1 Настройка каталога курса.

Делаю снимок сделанных изменений и push'у их на свой репозиторий в GitHub. (рис 4.3.2)

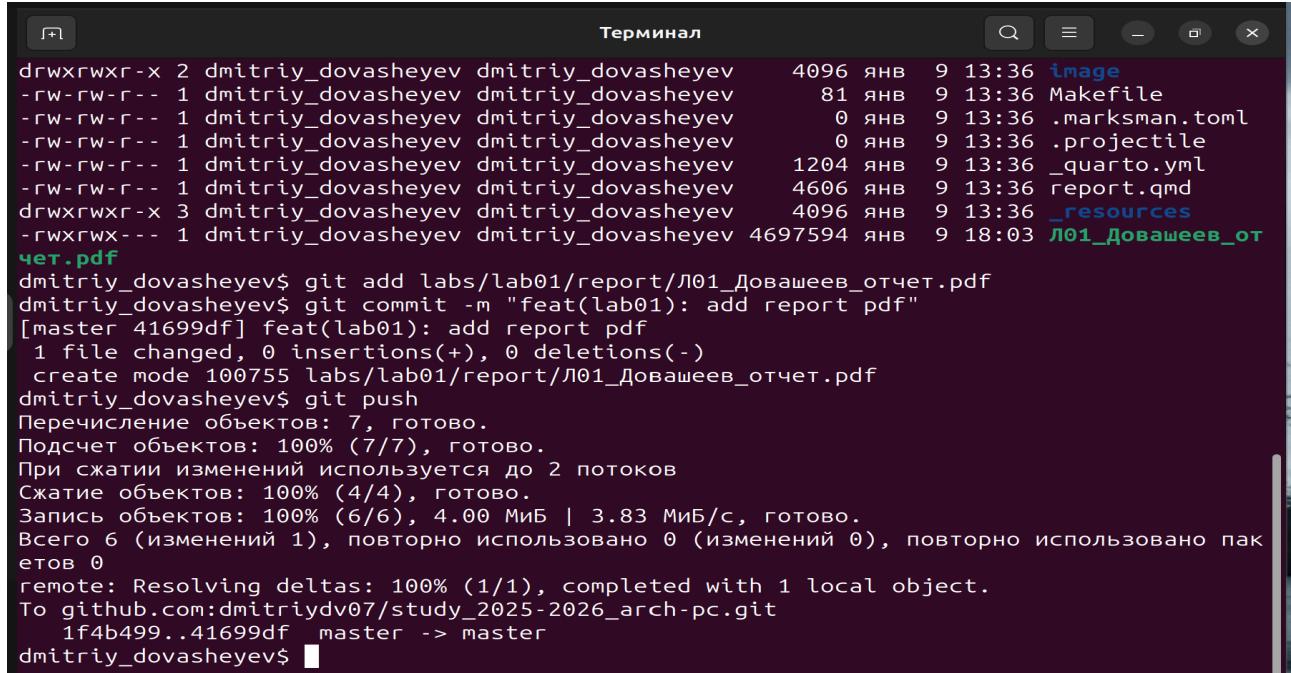


```
dmitriy_dovasheyev$ git add .
dmitriy_dovasheyev$ git commit -am "feat(main): make course structure"
[master 1f4b499] feat(main): make course structure
 2 files changed, 1 insertion(+), 14 deletions(-)
   delete mode 100644 package.json
dmitriy_dovasheyev$ git push
Перечисление объектов: 5, готово.
Подсчет объектов: 100% (5/5), готово.
При сжатии изменений используется до 2 потоков
Сжатие объектов: 100% (2/2), готово.
Запись объектов: 100% (3/3), 295 байтов | 295.00 КиБ/с, готово.
Всего 3 (изменений 1), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:dmitriydv07/study_2025-2026_arch-pc.git
  b449d8f..1f4b499 master -> master
dmitriy_dovasheyev$
```

Рис 4.3.2 Отправка изменений на удаленный репозиторий.

5 Задания для самостоятельной работы.

Через терминал отправляю предыдущий отчет по лабораторной работе на свой удаленный репозиторий в GitHub (рис. 5.1), затем проверяю изменения на самом GitHub. (рис 5.2)



```
Терминал
dmitriy_dovasheyev$ ls
dmitriy_dovasheyev$ git add labs/lab01/report/Л01_Довашеев_отчет.pdf
dmitriy_dovasheyev$ git commit -m "feat(lab01): add report pdf"
[master 41699df] feat(lab01): add report pdf
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100755 labs/lab01/report/Л01_Довашеев_отчет.pdf
dmitriy_dovasheyev$ git push
Перечисление объектов: 7, готово.
Подсчет объектов: 100% (7/7), готово.
При сжатии изменений используется до 2 потоков
Сжатие объектов: 100% (4/4), готово.
Запись объектов: 100% (6/6), 4.00 МиБ | 3.83 МиБ/с, готово.
Всего 6 (изменений 1), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:dmitriydv07/study_2025-2026_arch-pc.git
 1f4b499..41699df master -> master
dmitriy_dovasheyev$
```

Рис 5.1 Отправка изменений на удаленный репозиторий.

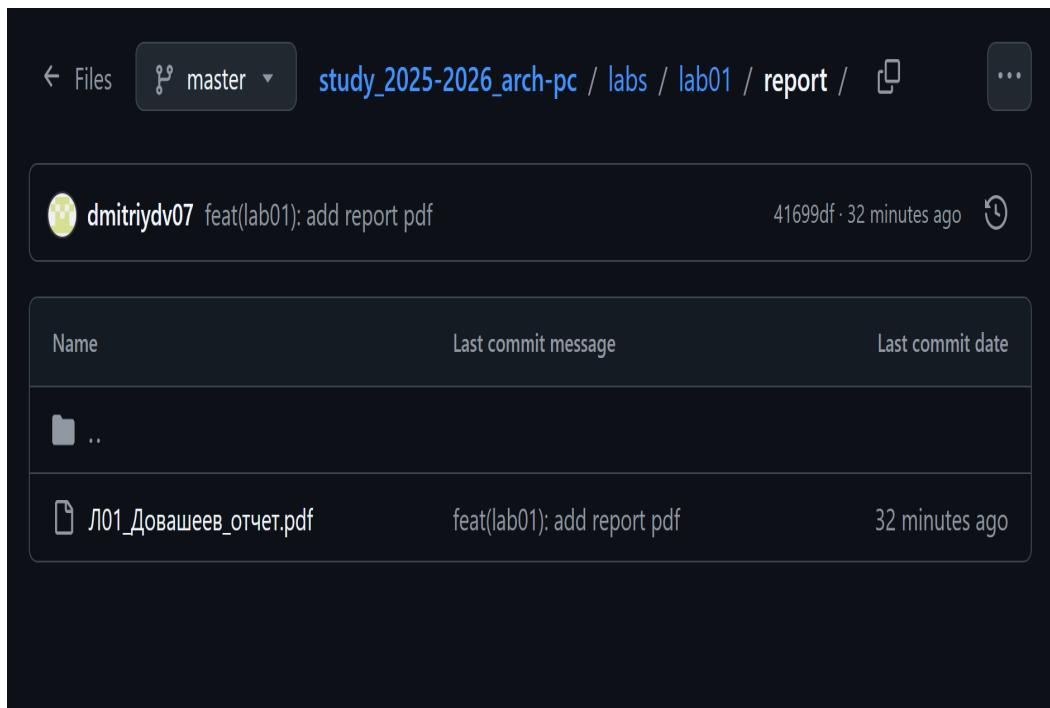


Рис 5.2 Проверка выполненных изменений на GitHub.

6 Выводы

При выполнении данной лабораторной работы я изучил идеологию и применение средств контроля версий, а также приобрел практические навыки по работе с реализацией VSC git.

Список литературы