Минестерство образования и науки Российской Федерации

ФГБОУ ВО Ульяновский государственный технический университет

Кафедра «вычислительная техника»

Лабораторная работа №5

«GIT»

Выполнил студент

Группы ИВТАСбд-21

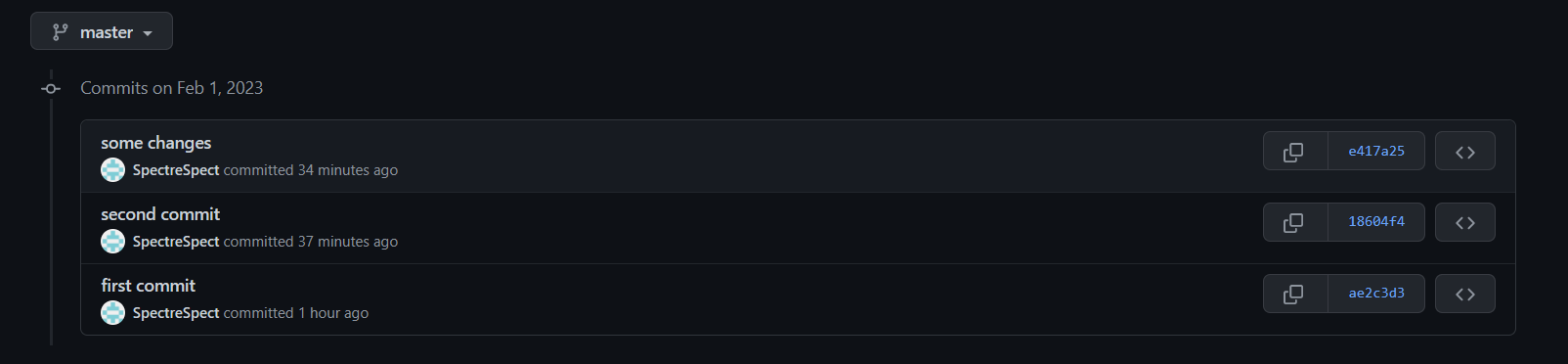
Долгов Д. П.

Ульяновск, 2022

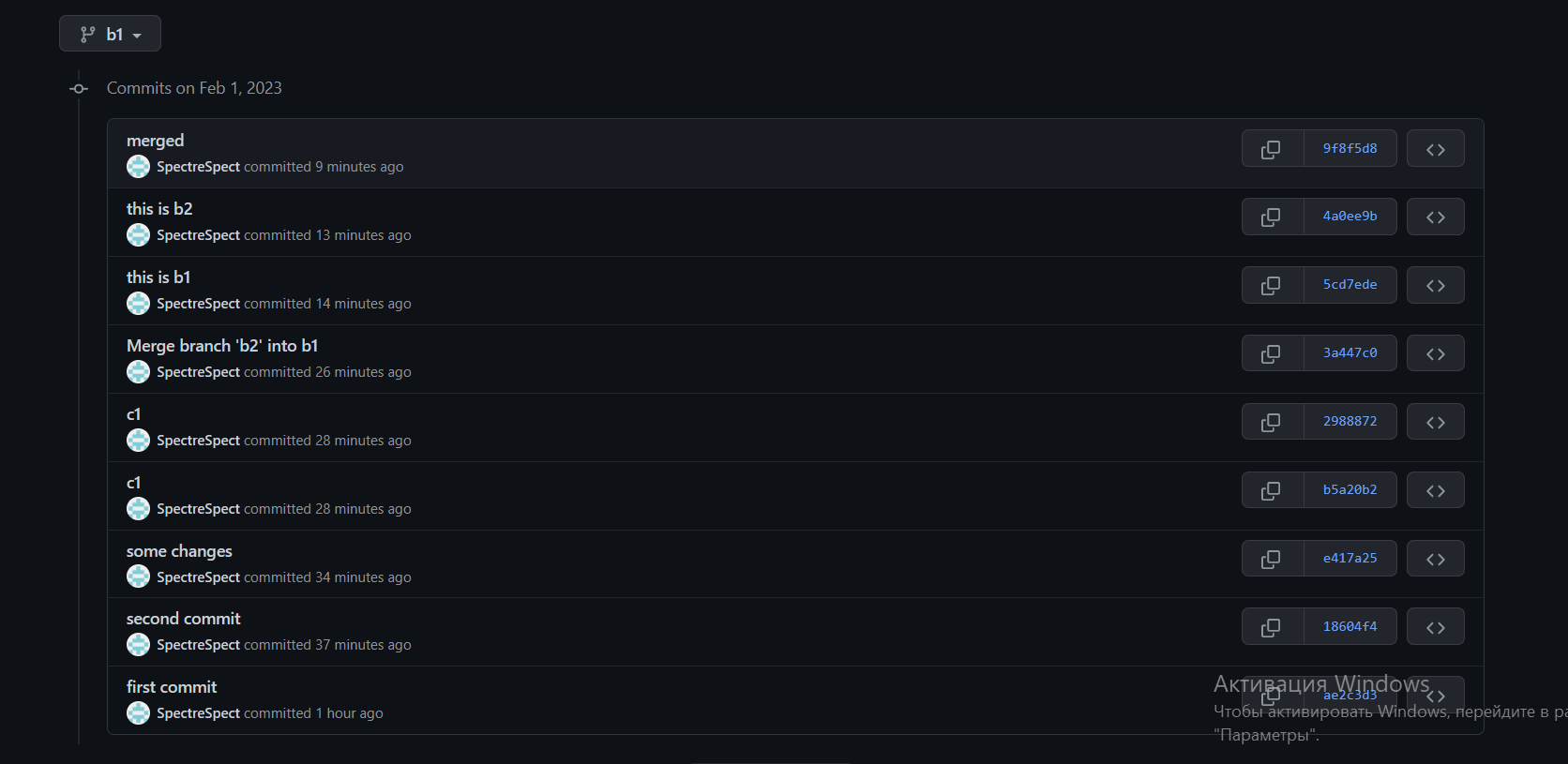
**Ссылка на гит:** [**https://github.com/KoTeuKaSeeker/laba4.git**](https://github.com/KoTeuKaSeeker/laba4.git)

**Адрес репозитория:** [**KoTeuKaSeeker/laba4: Это репозиторий, на котором хранится лаба 4 (github.com)**](https://github.com/KoTeuKaSeeker/laba4)

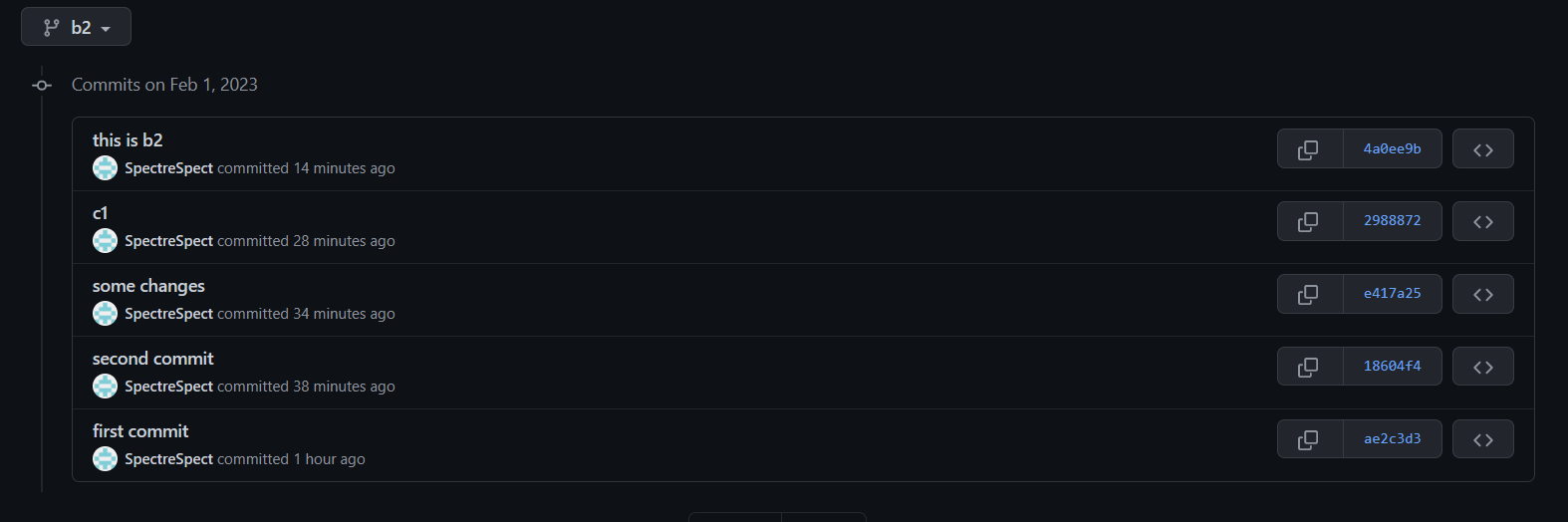
**История коммитов ветки мастер:**

****

**История коммитов ветки b1:**

****

**История коммитов ветки b2:**

****

**Команды**

git init - создает локальный репозиторий

git remote add origin <ссылка> - подключается к удалённому репозиторию

git add . - добавляет файлы в индекс

git status - показывает, как файлы находятся в индексе

git commit -m “название коммита” - фиксирует изменения

git push origin master - отправляет данные ветки на удалённый репозиторий

git pull origin master - получает данные с удалённого репозитория

git branch - показывает список веток

git checkout <название ветки> - переключается между ветками

git merge <название ветки> - сливает ветки

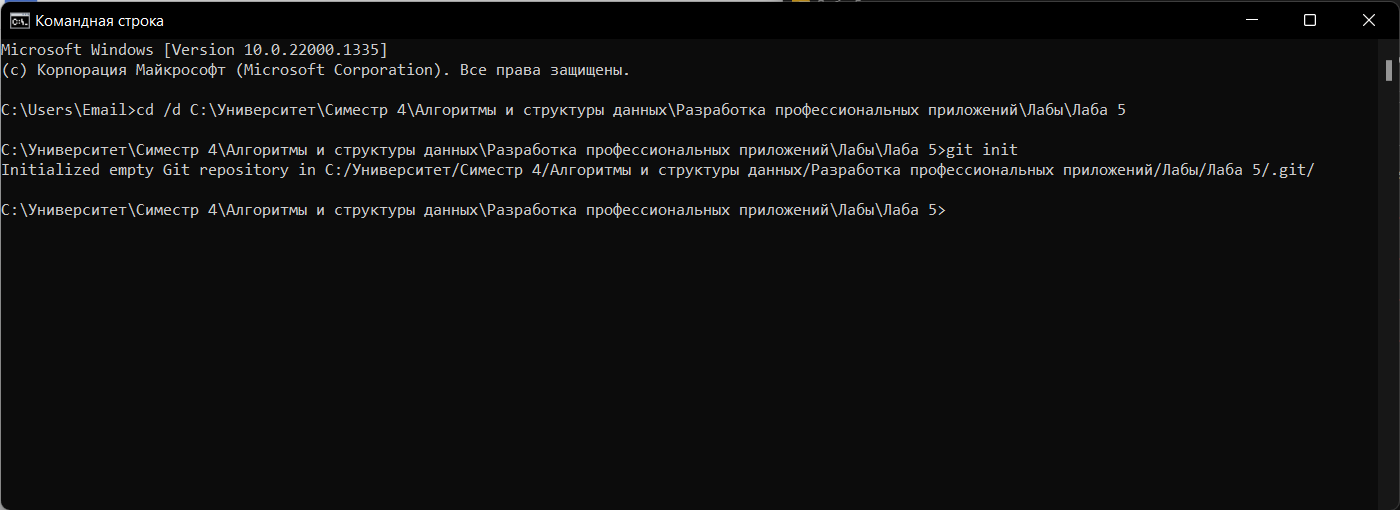
git clone <ссылка> - клонирует удалённый репозиторий

git checkout --ours . - принимает данные текущей ветки в конфликте

git checkout --theirs . - принимает данные сливаемой ветки в конфликте

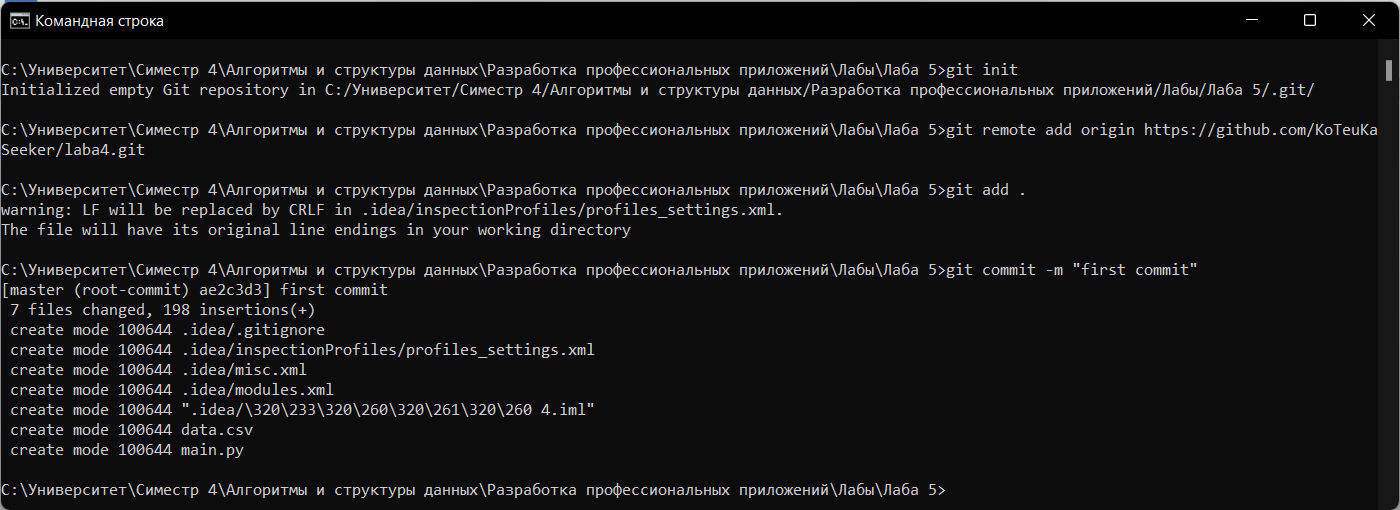
**2.1 Создать локальный репозиторий**

Для создания локального репозитория нужно зайти в консоль в каталог с проектом и прописать команду git init:



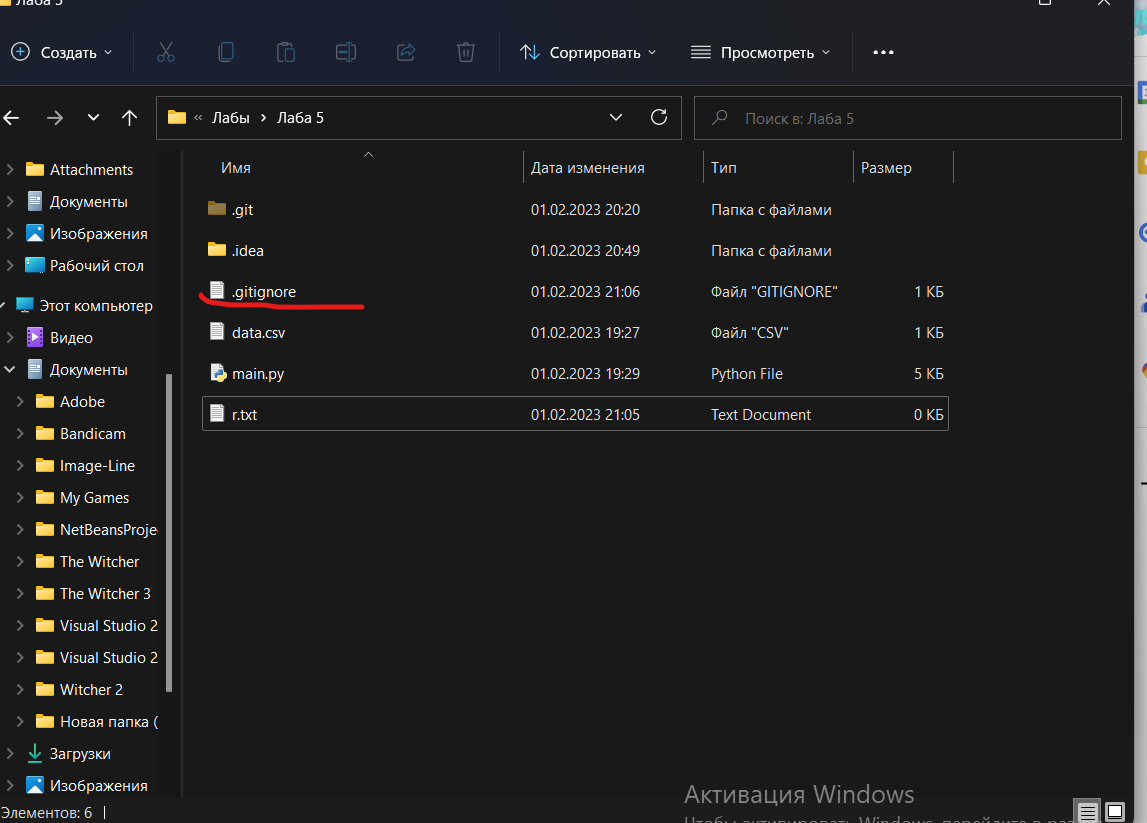
**2.2 Добавить несколько файлов**

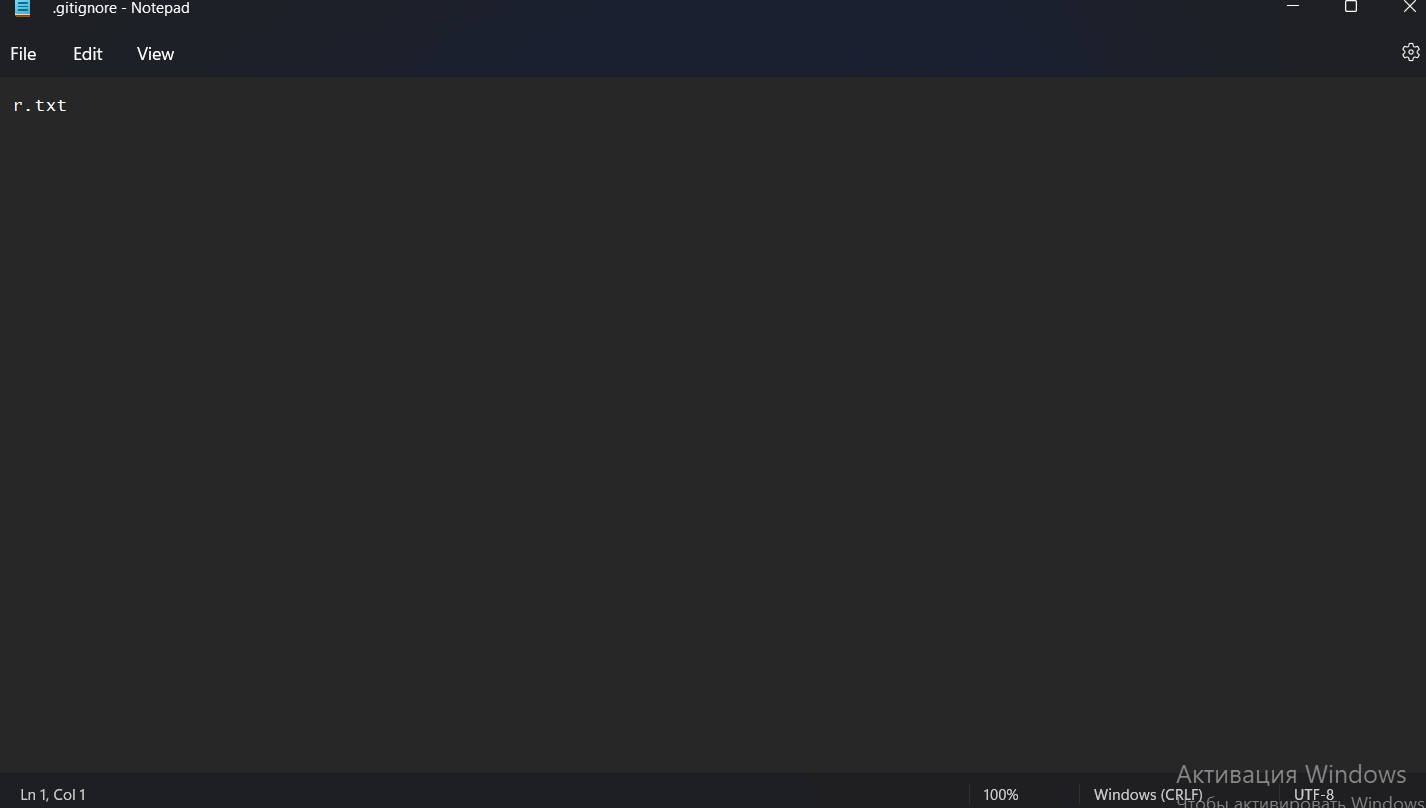
Для добавления файлов нужно добавить их в статус командой git add ., а затем создать коммит командой git commit -m “название коммита”



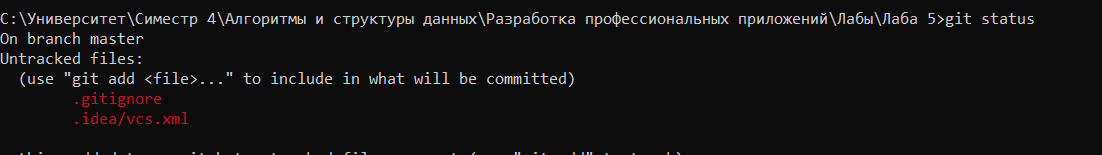
**2.3 Сделать один из файлов игнорируемым**

gitignore представляет собой некий чёрный список для git. Чтобы его создать, нужно создать файл .gitignore и в него прописать название игнорируемых файлов:





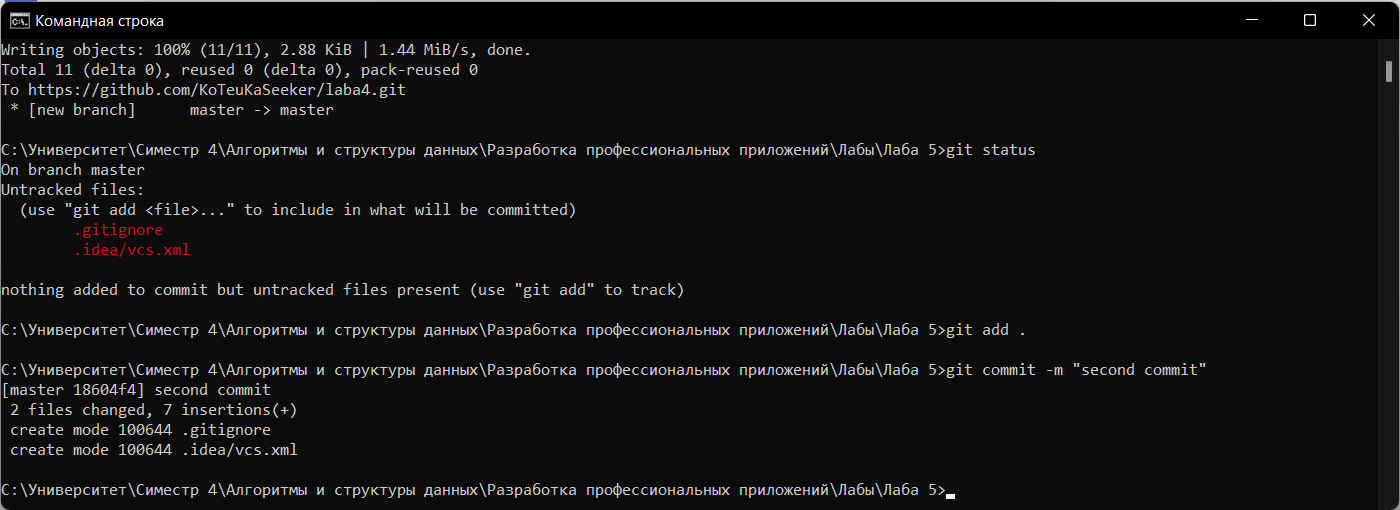
Теперь если закоммитить данные, то r.txt не попадёт в этот коммит. Можно это проверить с помощью команды git status:



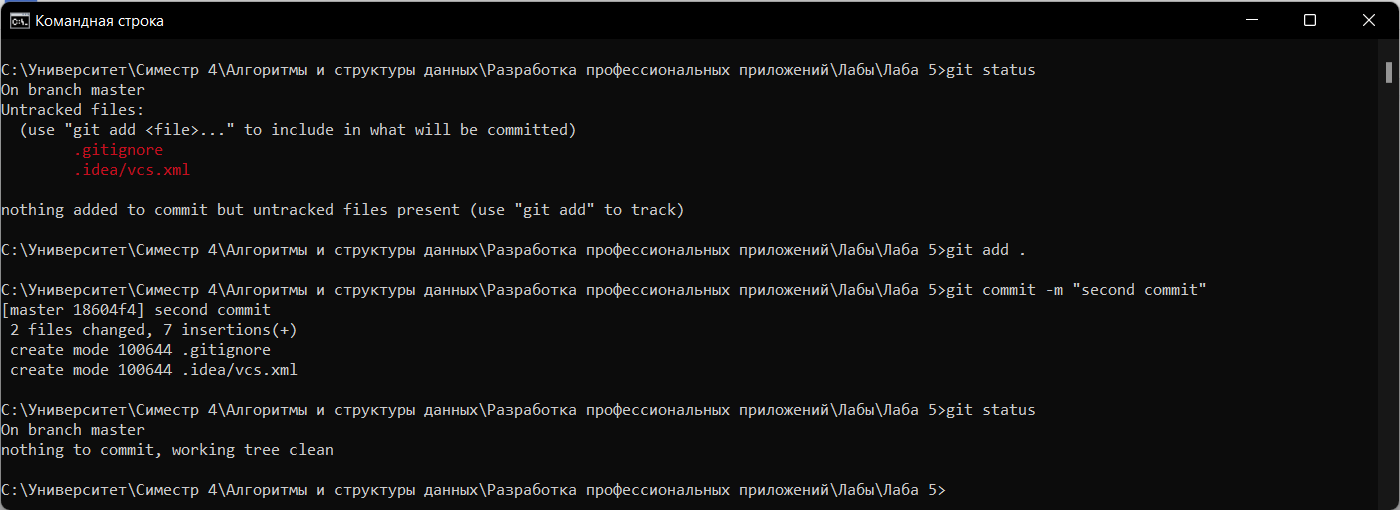
Как видно, файла r.txt здесь нет.

2.4 Зафиксировать изменения.

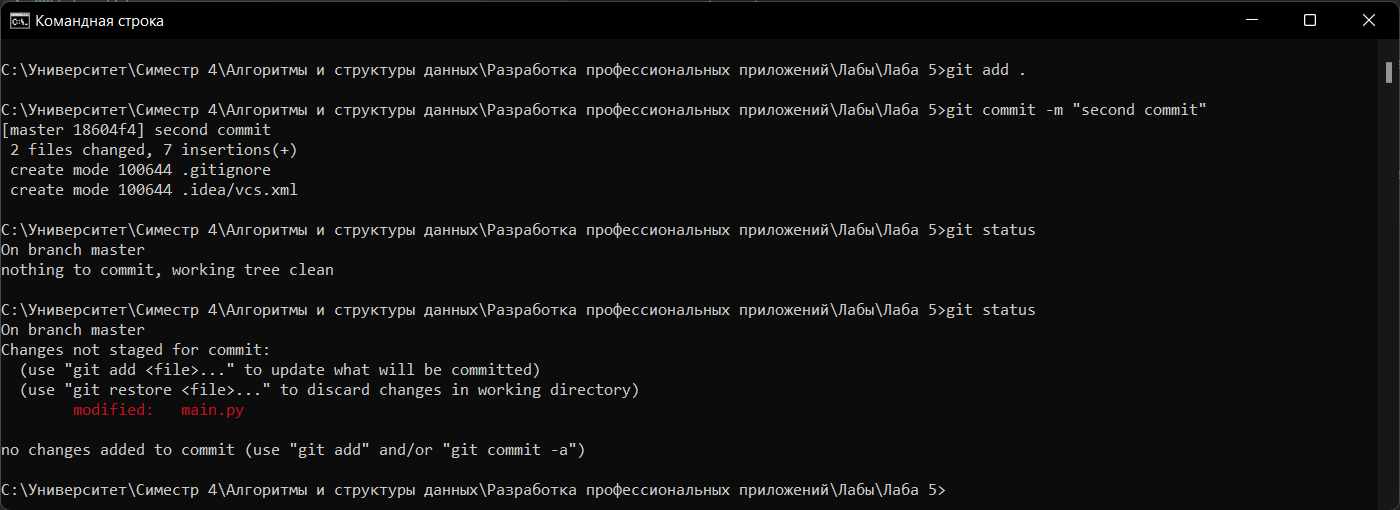
Чтобы зафиксировать изменения нужно добавить их в статус с помощью команды git add ., а затем создать коммит с помощью команды git commit -m “название коммита”



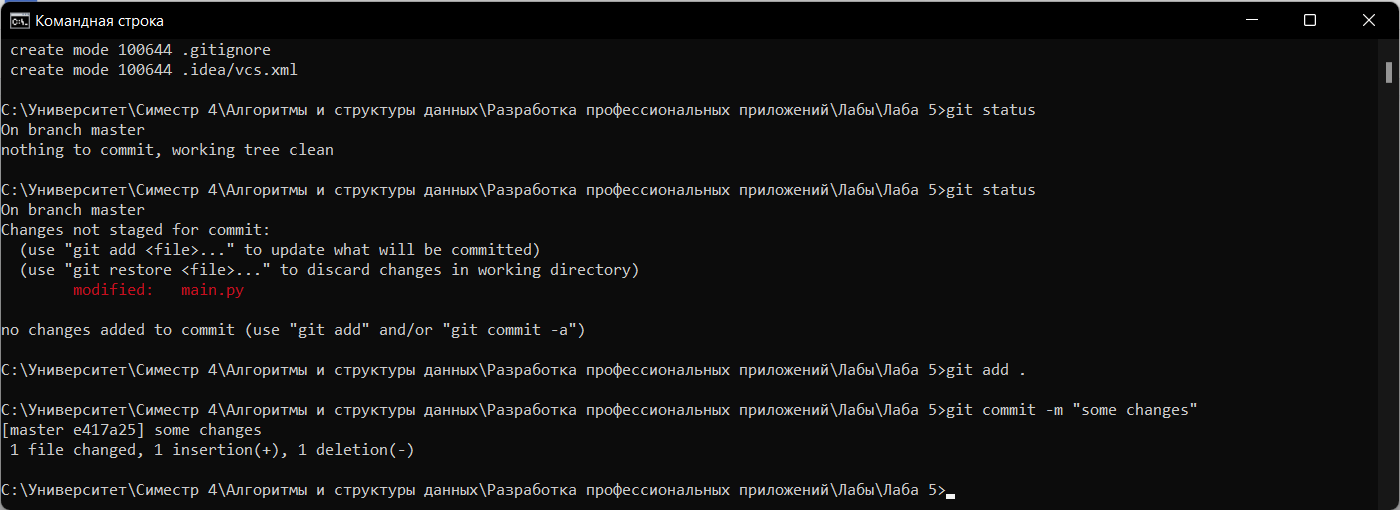
Напишем теперь команду git status:



Как видно, изменений в папке нет. Изменим файл main.py, добавив в него комментарий, а затем снова пропишем команду git status:

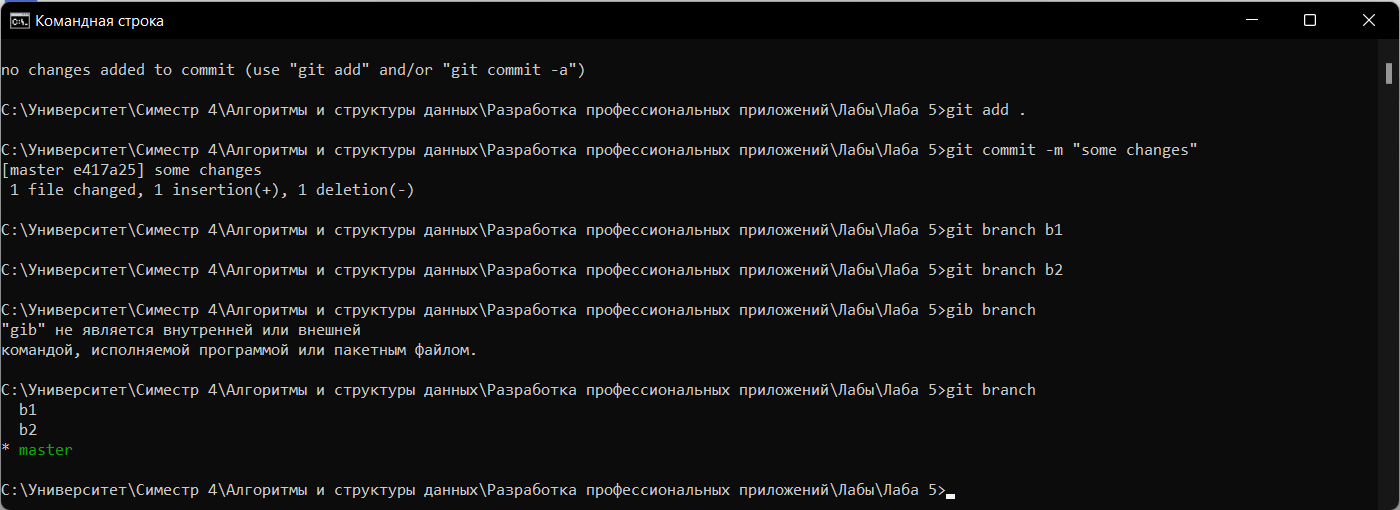


Как видно, изменения были замечены гитом. Добавим их в статус и зафиксируем с помощью команд git add . и git commit -m “название коммита”:

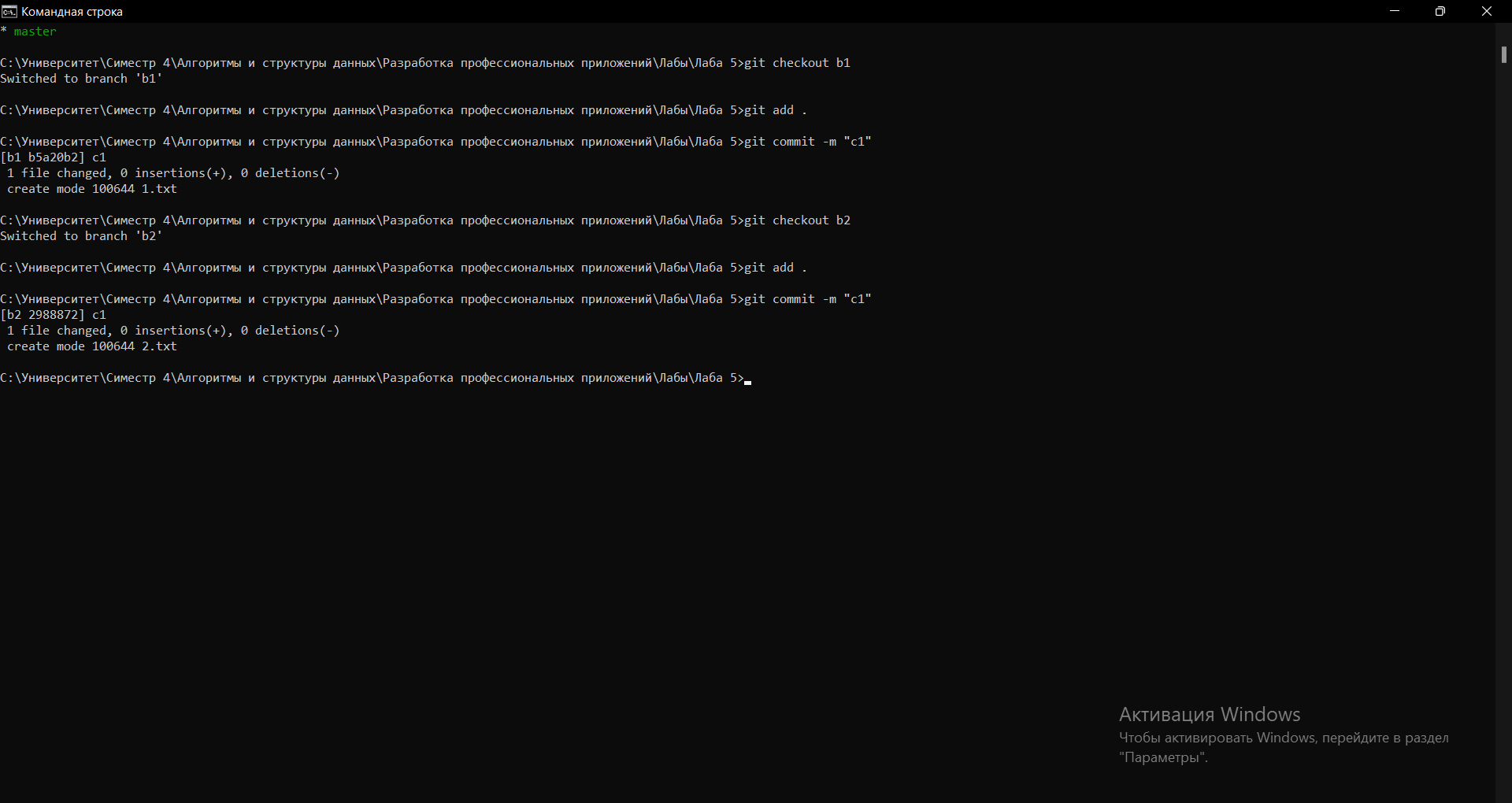


**2.6 Создать несколько бранчей**

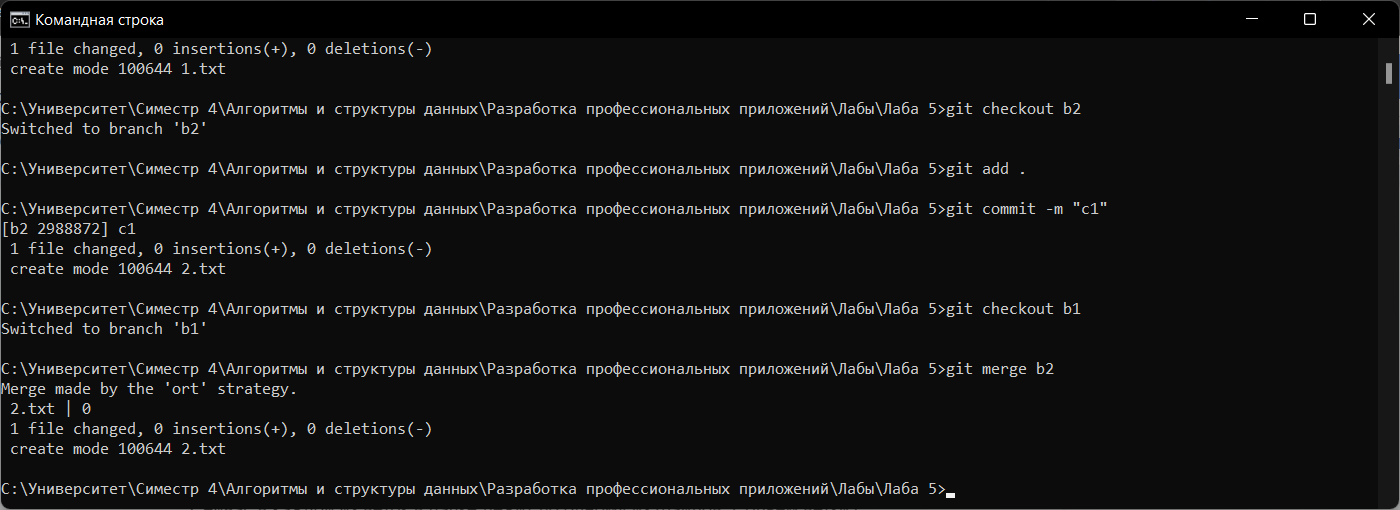
Чтобы создать ветки в git нужно использовать команду git branch <название ветки>



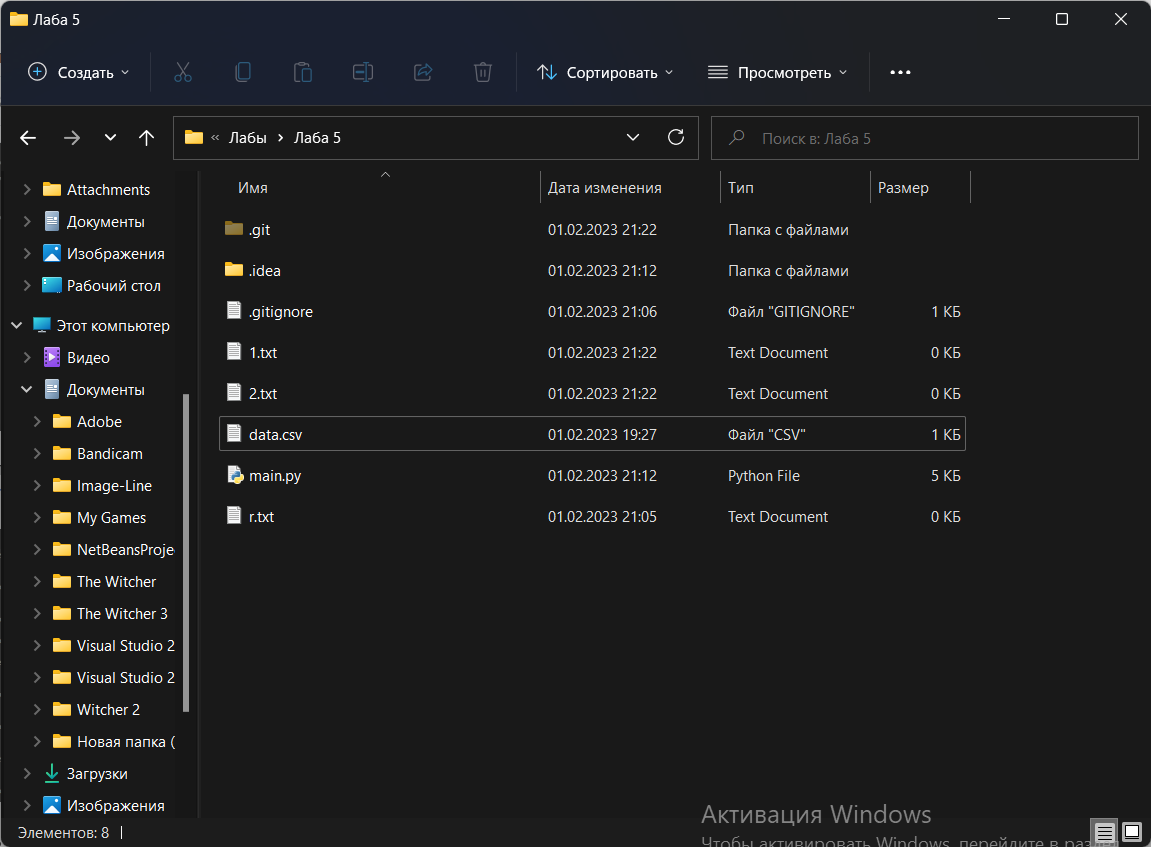
Перейдём в ветку b1 и создадим файл 1.txt, а в ветки b2 создадим файл 2.txt. Для перехода на ветку используется команда git checkout <название ветки>



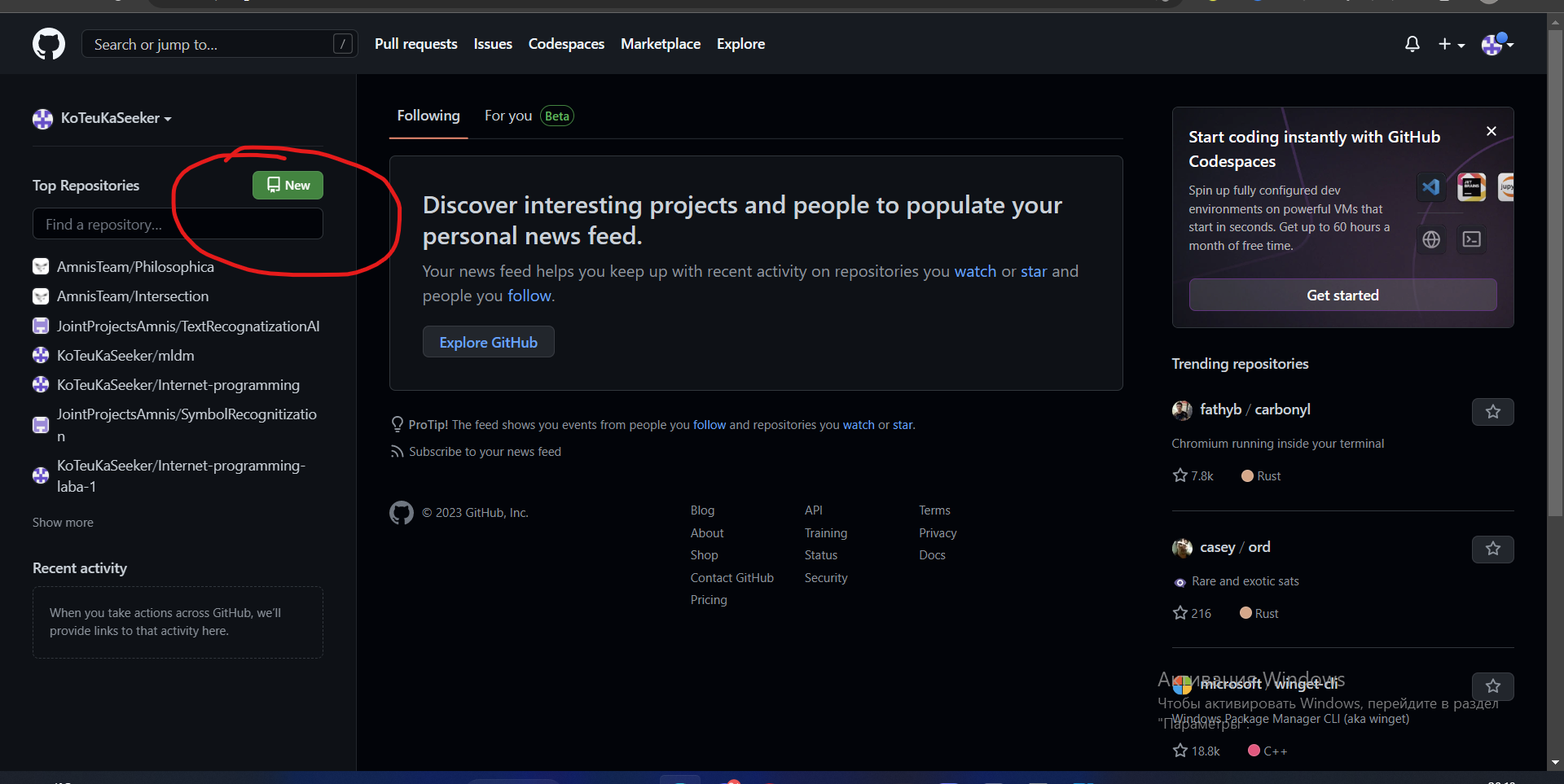
Сейчас в каждой из веток в папке лежит по одному из файлов. Сольём ветки с помощью команды git merge <название ветки>:

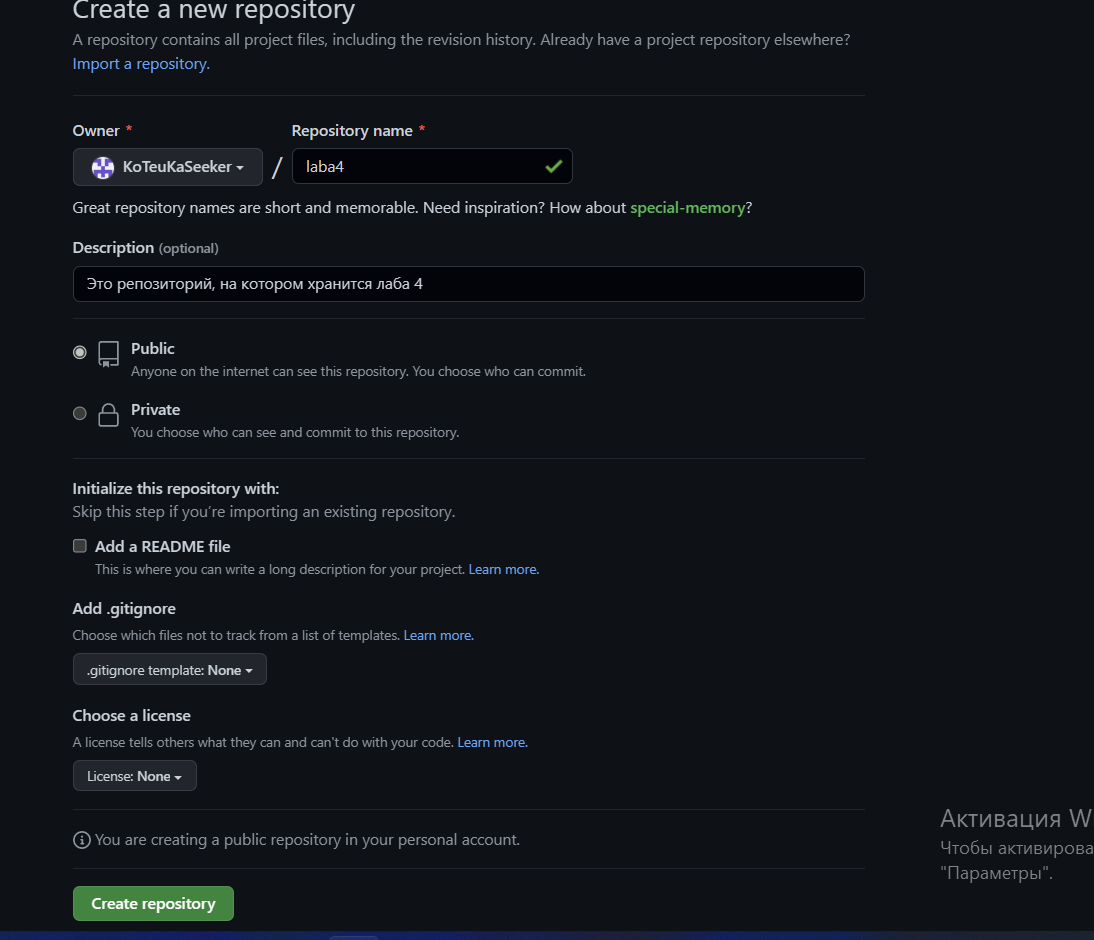


Теперь в каталоге есть оба файла:

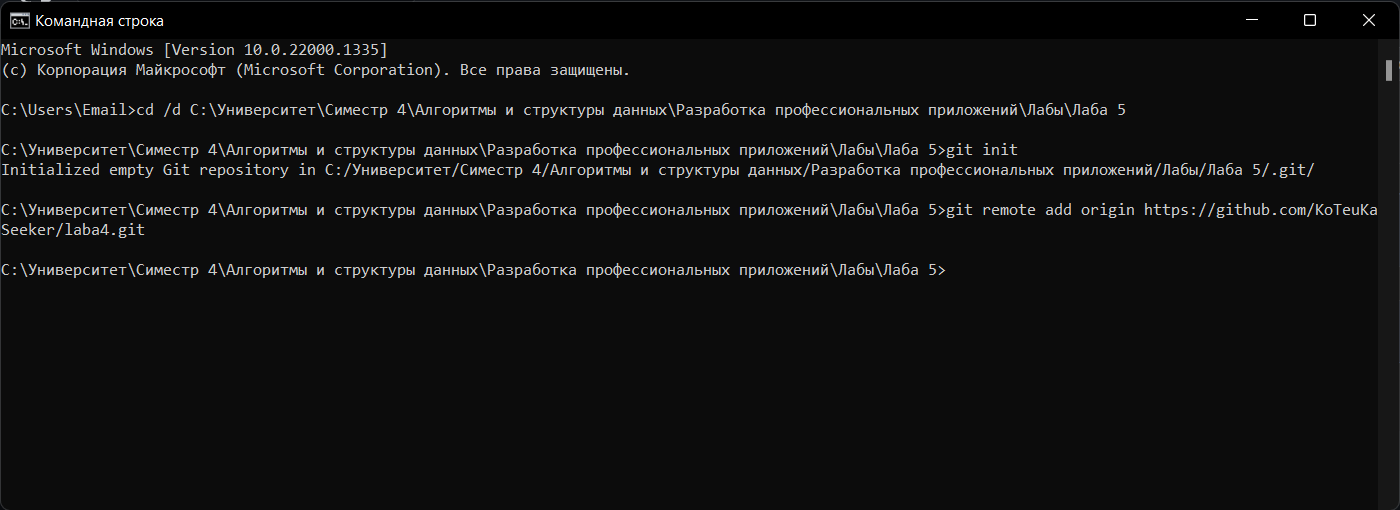


**27. Создать удаленный репозиторий и отправить туда данные**

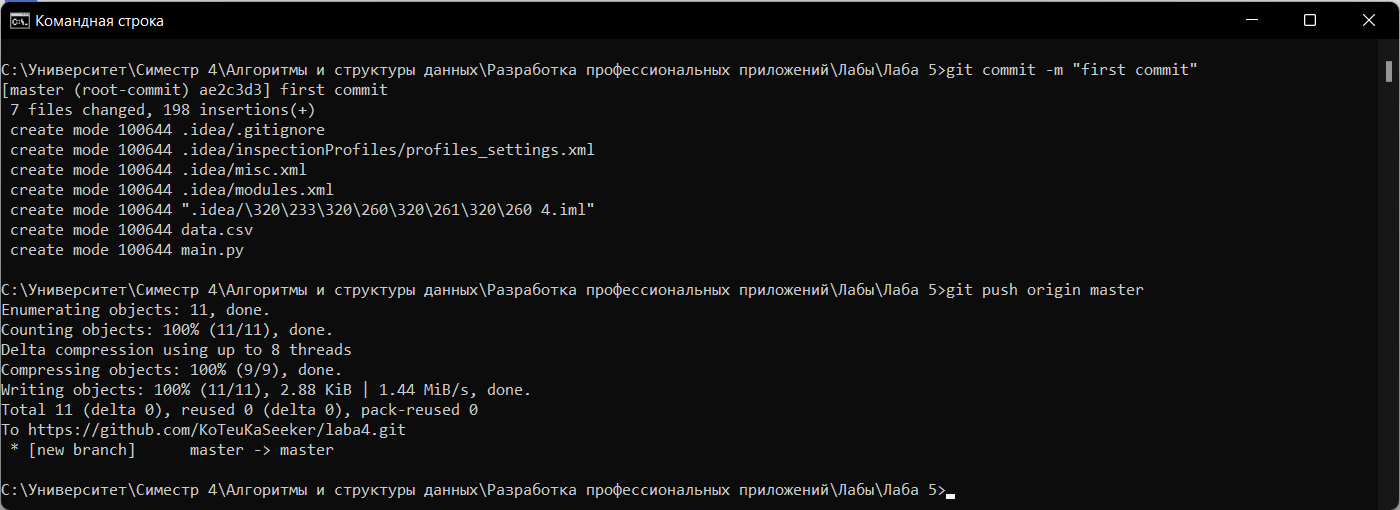
Чтобы создать удаленный репозиторий нужно нажать на эту кнопку:



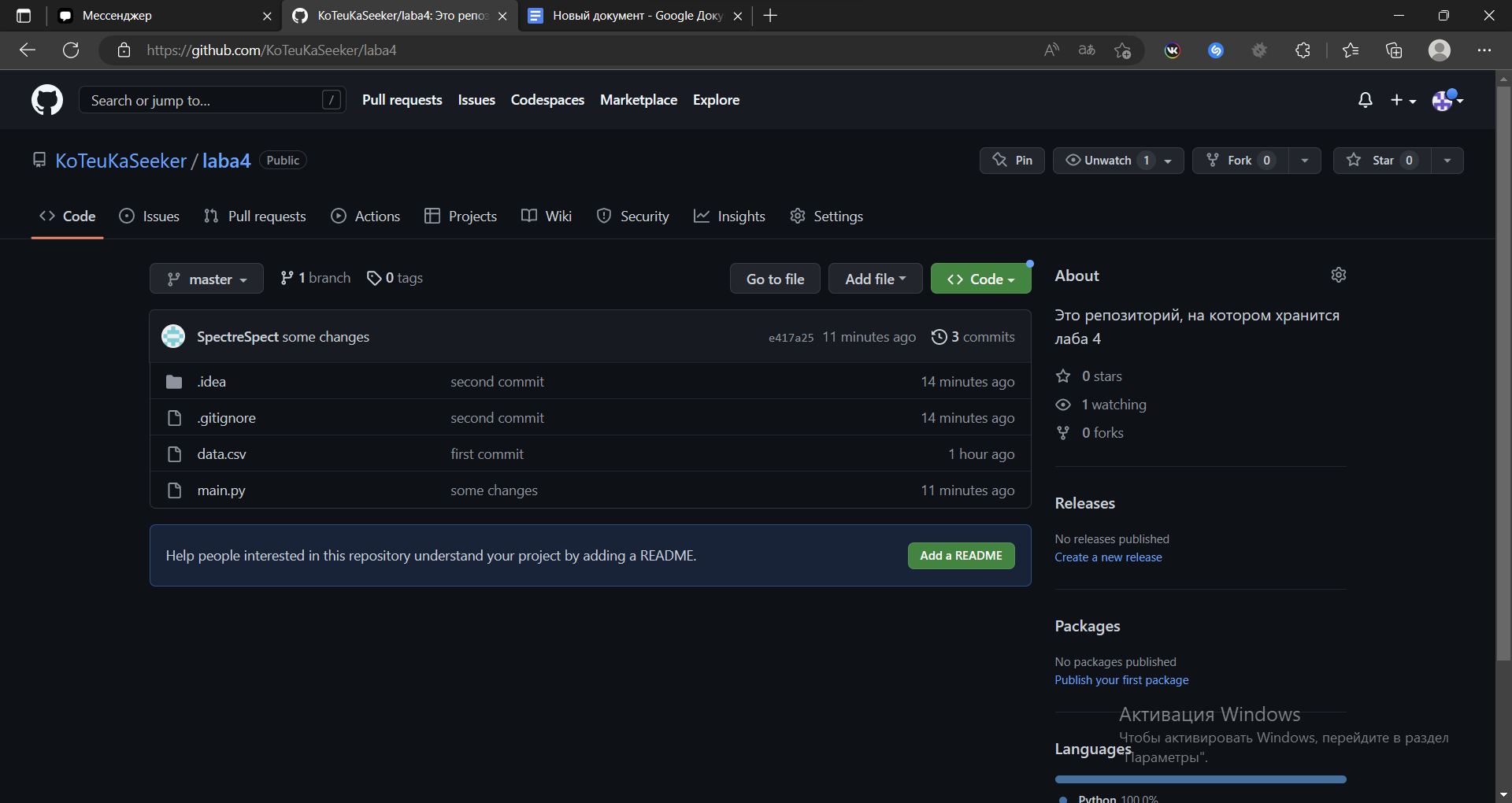
Для подключения удалённого репозитория нужно написать команду git remote add origin <ссылка на удалённый репозиторий>



Чтобы данные появились на удаленном репозитории нужно использовать команду git push origin master:

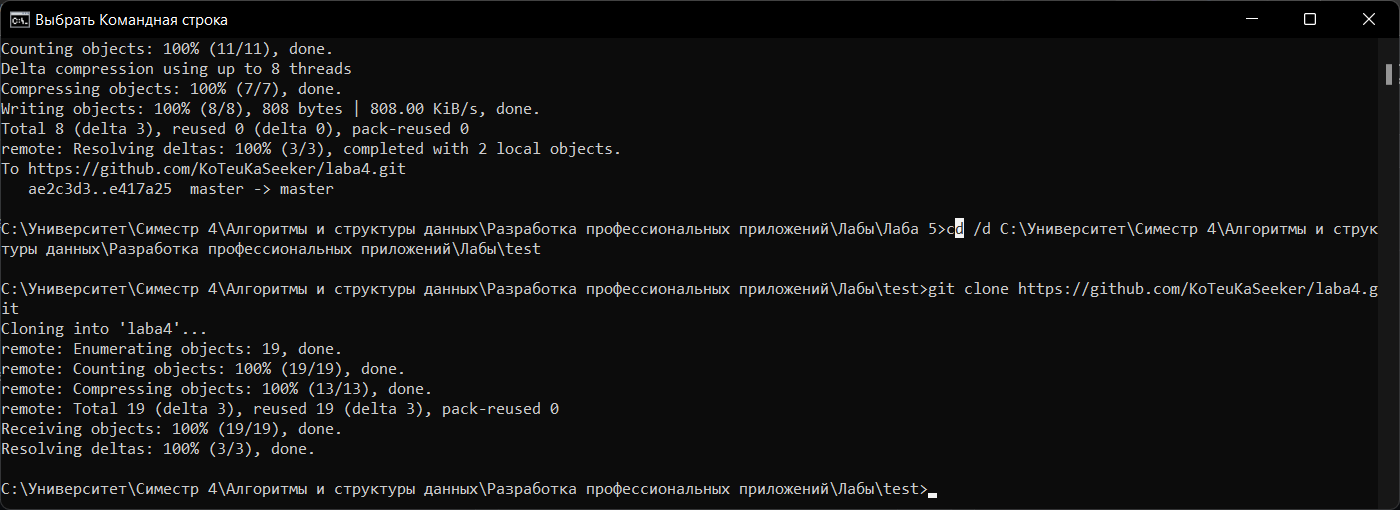


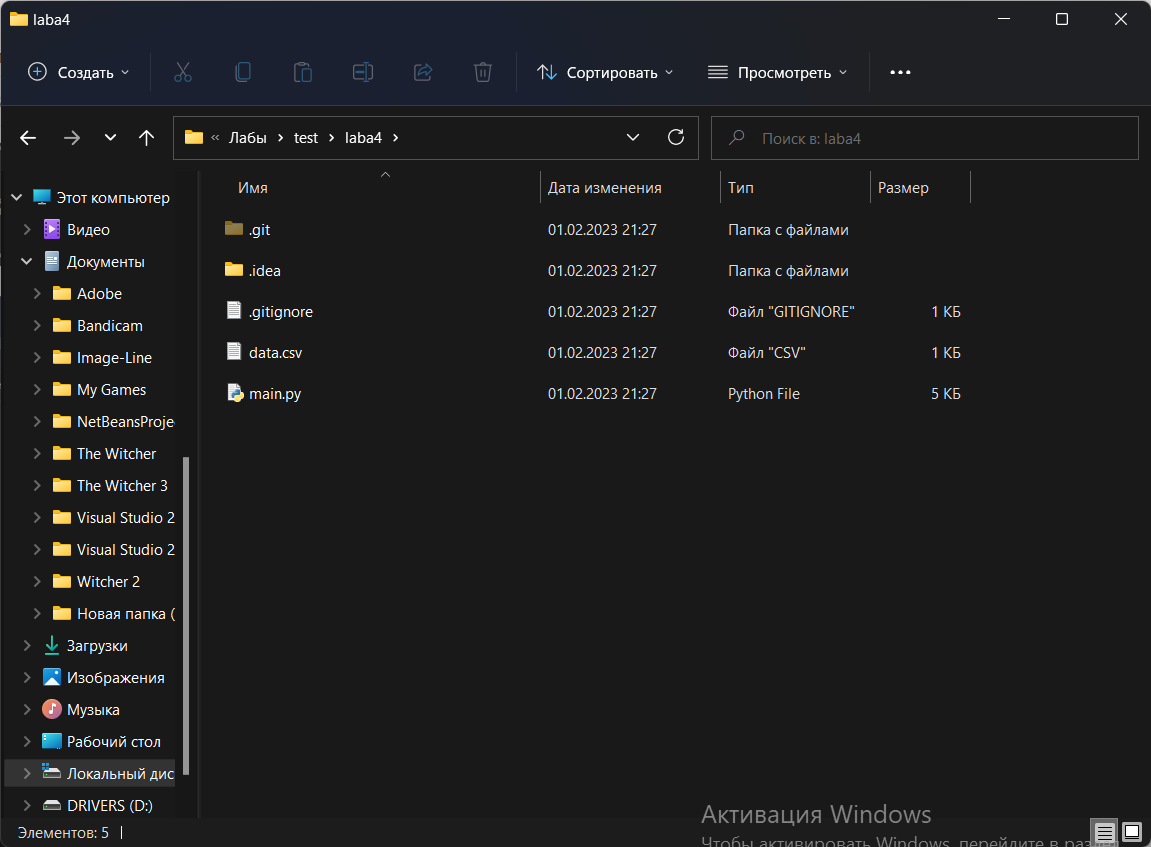
Теперь вот как выглядит удаленный репозиторий



**2.8 Склонировать удаленный репозиторий**

Для того, чтобы склонировать удалённый репозиторий, нужно использовать команду git clone <ссылка на удалённый репозиторий>:





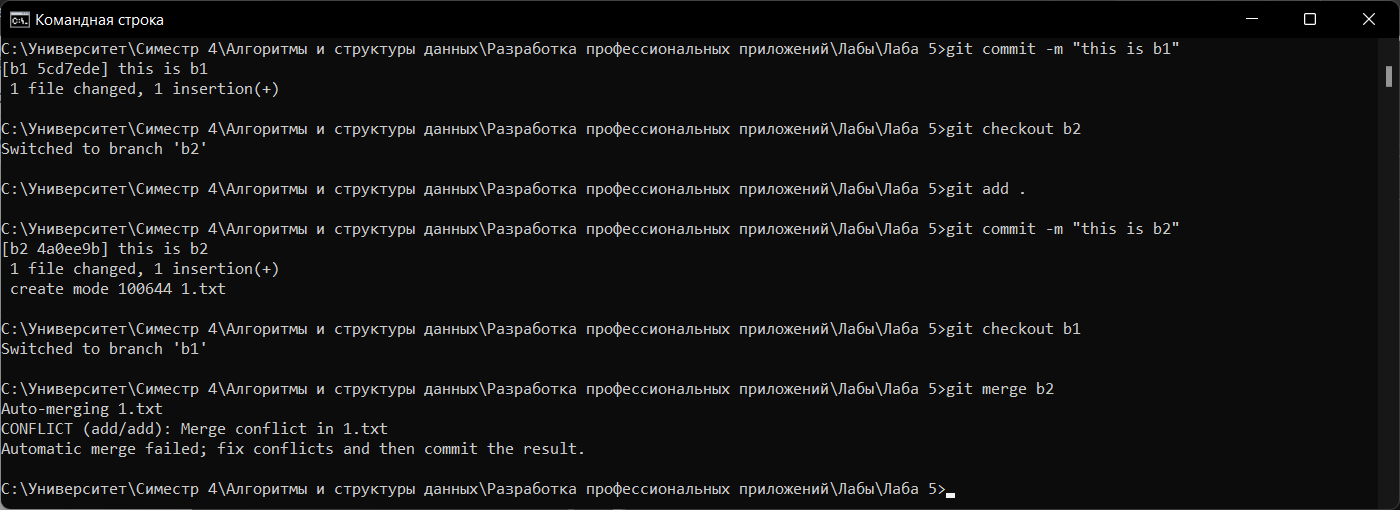
**2.9 Добиться конфликта**

Для того, чтобы добиться конфликта, нужно слить ветки, в которых изменён один и тот же файл. Проделаем это с ветками b1 и b2, а также с файлом 1.txt. Изменим файл 1.txt в ветке b1 и b2:





После выполнения команды git merge b2 появится сообщение о конфликте:



А файл 1.txt будет выглядеть следующим образом:



**2.10 Разрешение конфликта**

Чтобы разрешить конфликт, нужно исправить файл нужным образом, добавить всё в статус, а затем создать дополнительный коммит, в которым уже будет всё исправлено:



