

**МИНОБРНАУКИ РОССИИ**

**федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Московский государственный технологический университет «СТАНКИН»**

**(ФГБОУ ВО «МГТУ «СТАНКИН»)**

Институт цифровых и интеллектуальных систем  
Кафедра компьютерных систем управления

Образовательная программа 15.03.04

«Автоматизация технологических процессов и производств»

Дисциплина «Основы системного ПО»

**Отчет**

**по лабораторной работе №1**

«Работа с GitHub»

Выполнил:

студент гр.АДБ-18-06 \_\_\_\_\_\_\_\_\_\_\_\_ Еремин Д.Д.

(дата) (подпись)

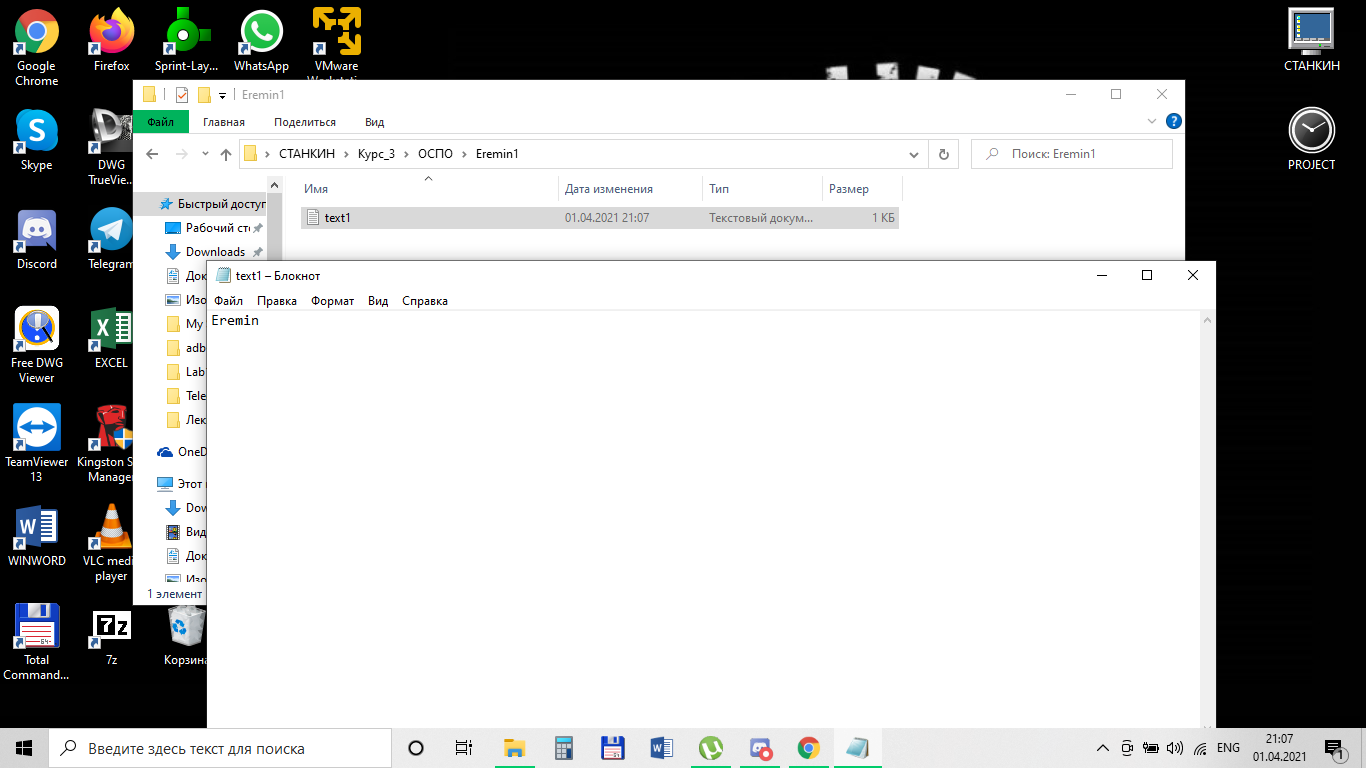
Принял:

к.т.н., доцент \_\_\_\_\_\_\_\_\_\_\_\_\_ Ковалев И.А.

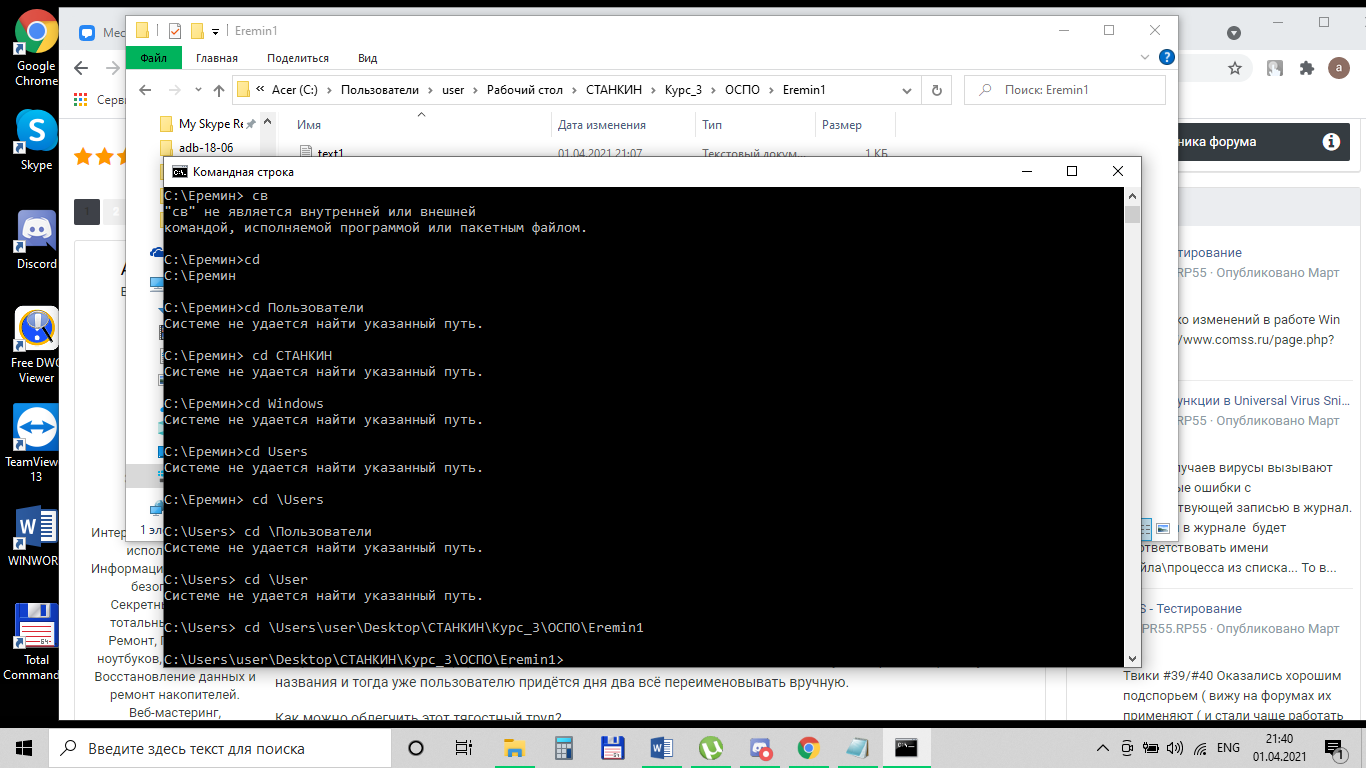
(дата) (подпись)

**Москва 2021**

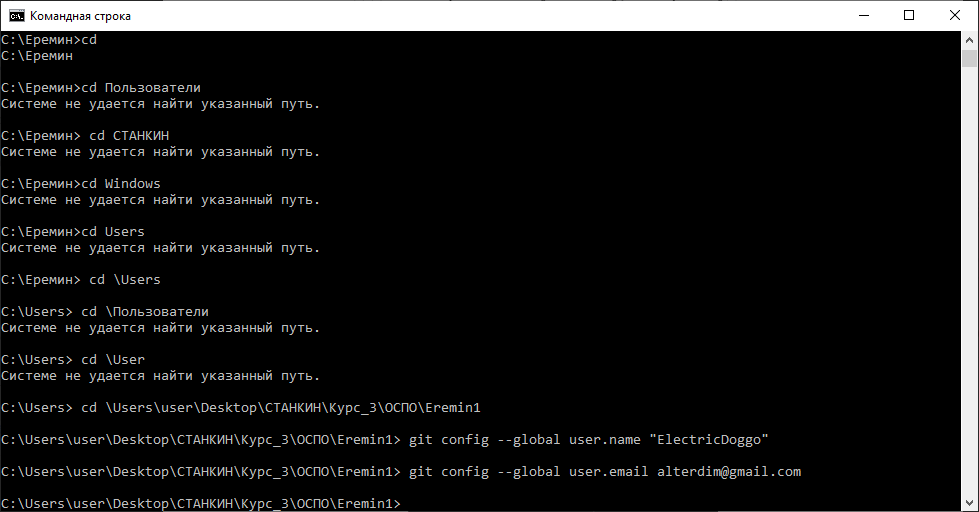
На диске создаем папку со своей фамилией и в ней документ, в который записана фамилия



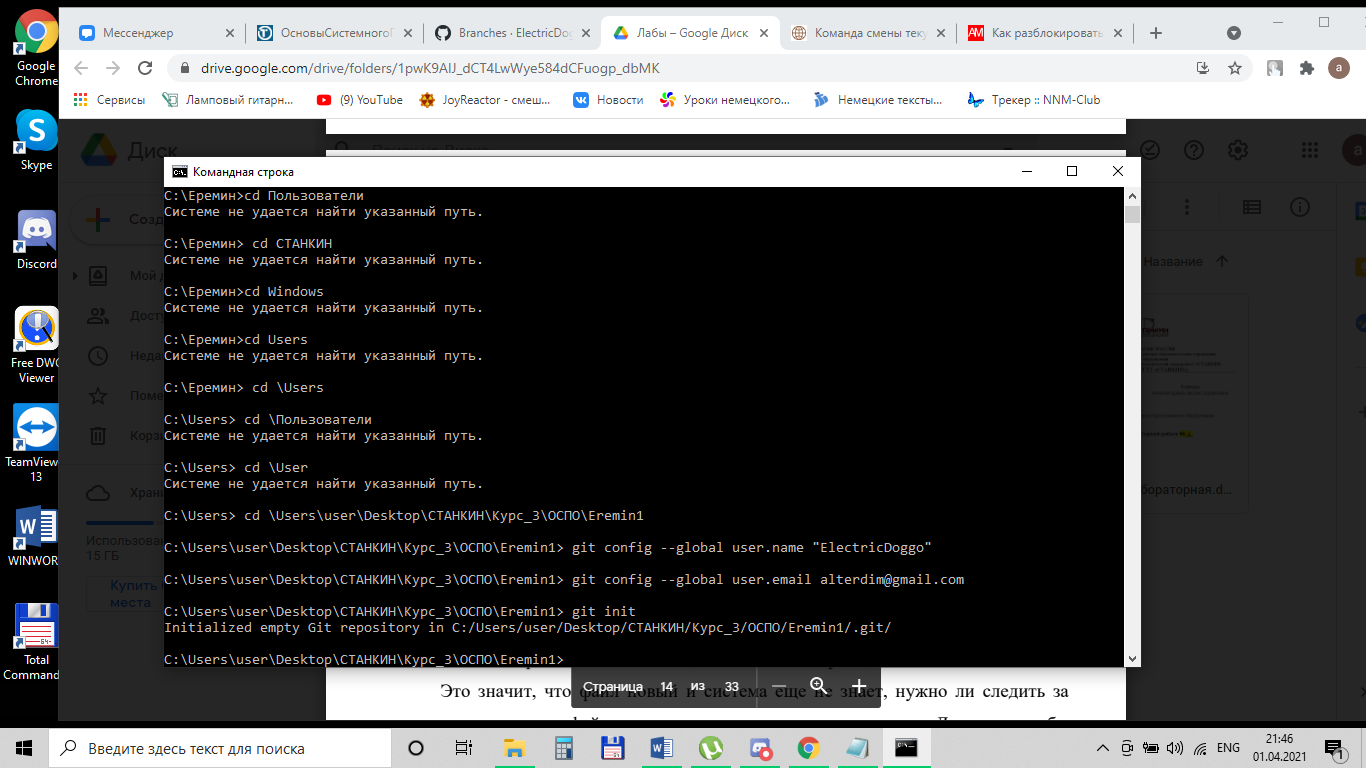
В командной строке переходим в указанную папку



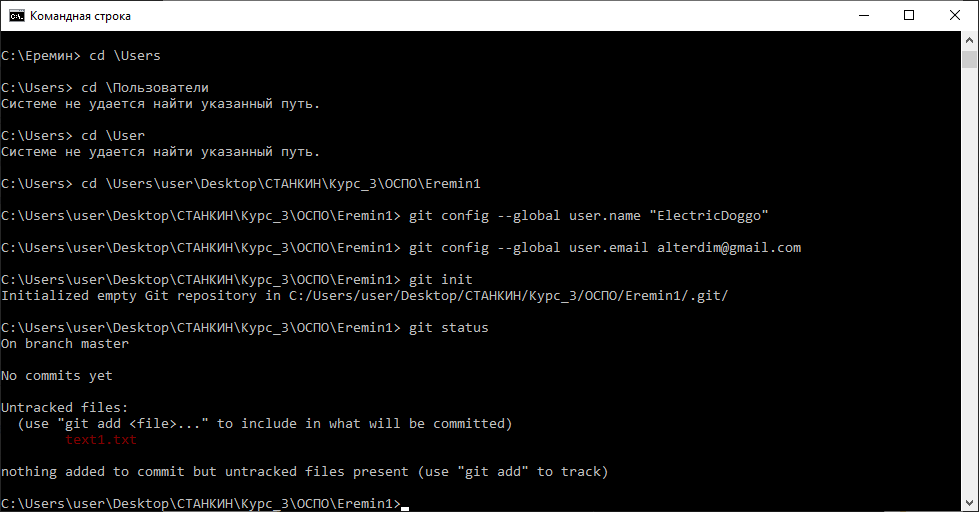
Зарегистрируем свое имя и электронную почту



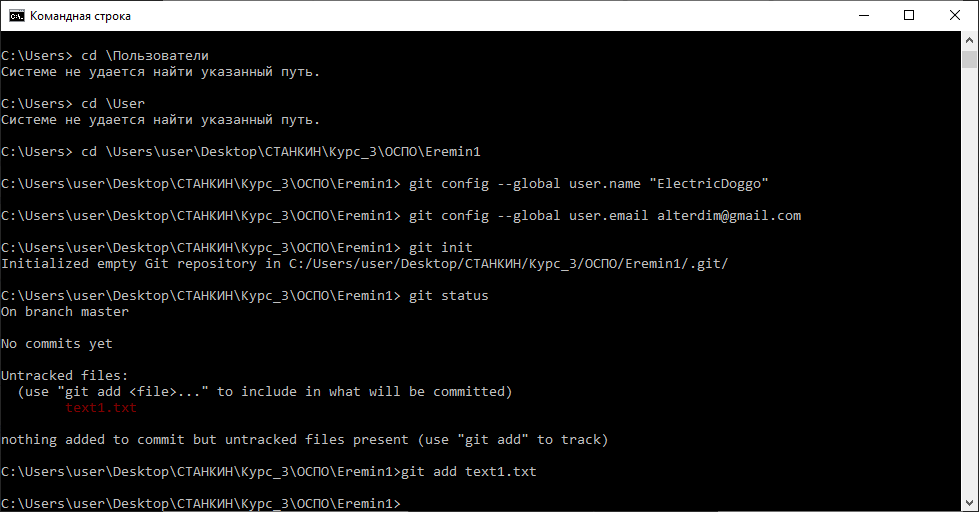
Проинициализируем папку как git репозиторий



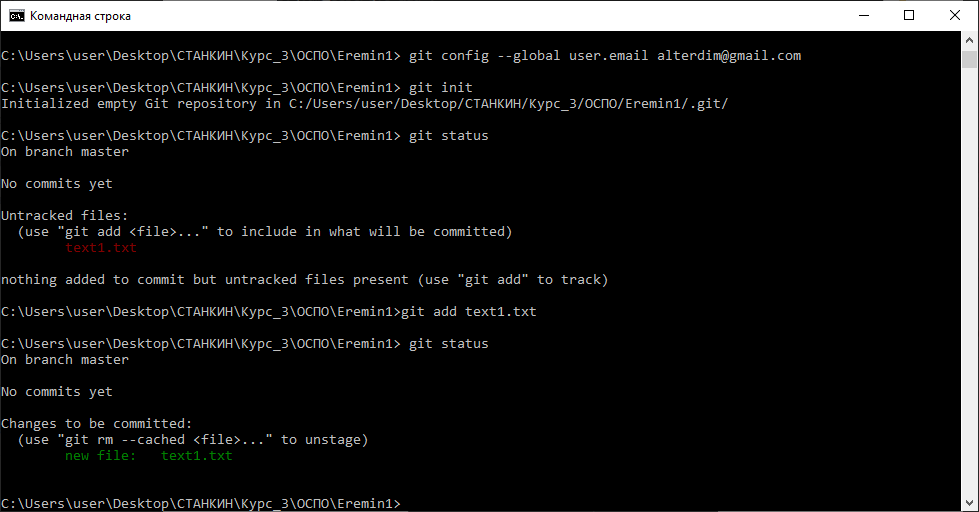
Отобразим статус



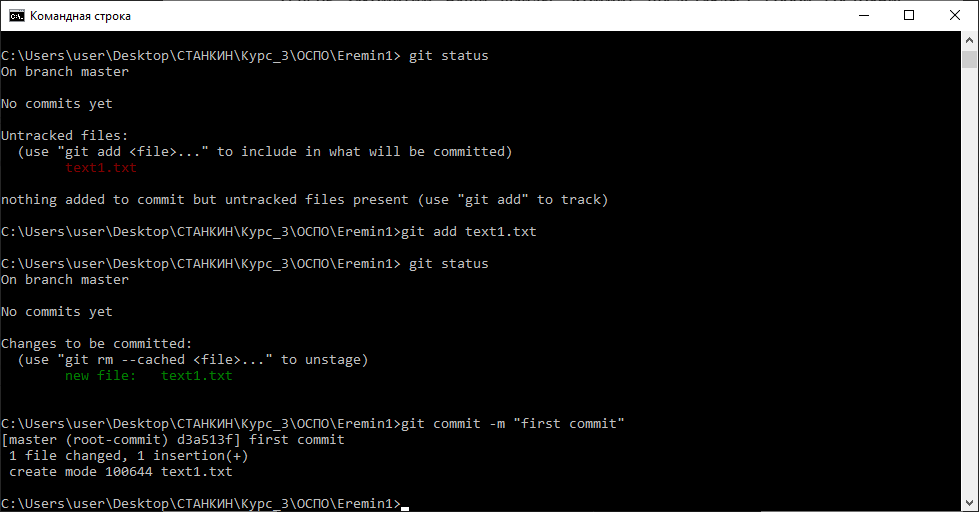
Добавляем файл в репозиторий

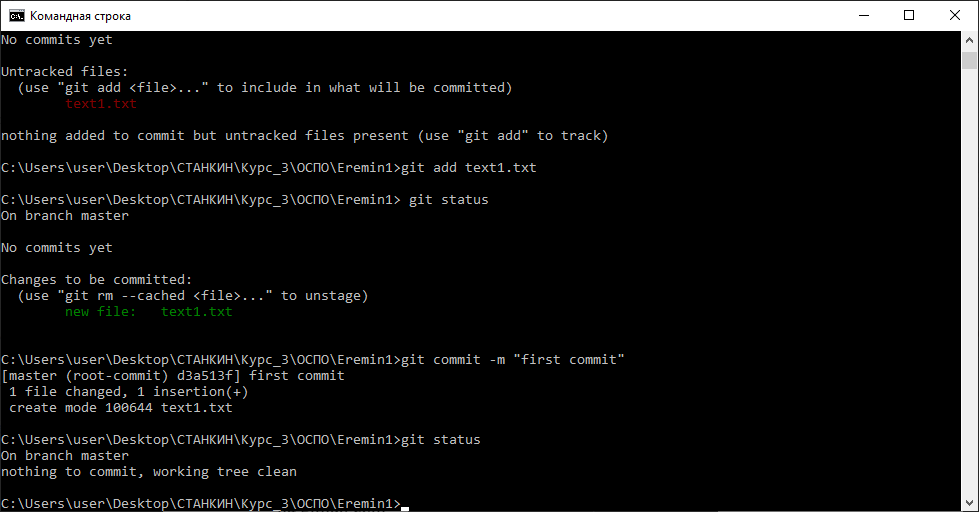


Файл готов к коммиту



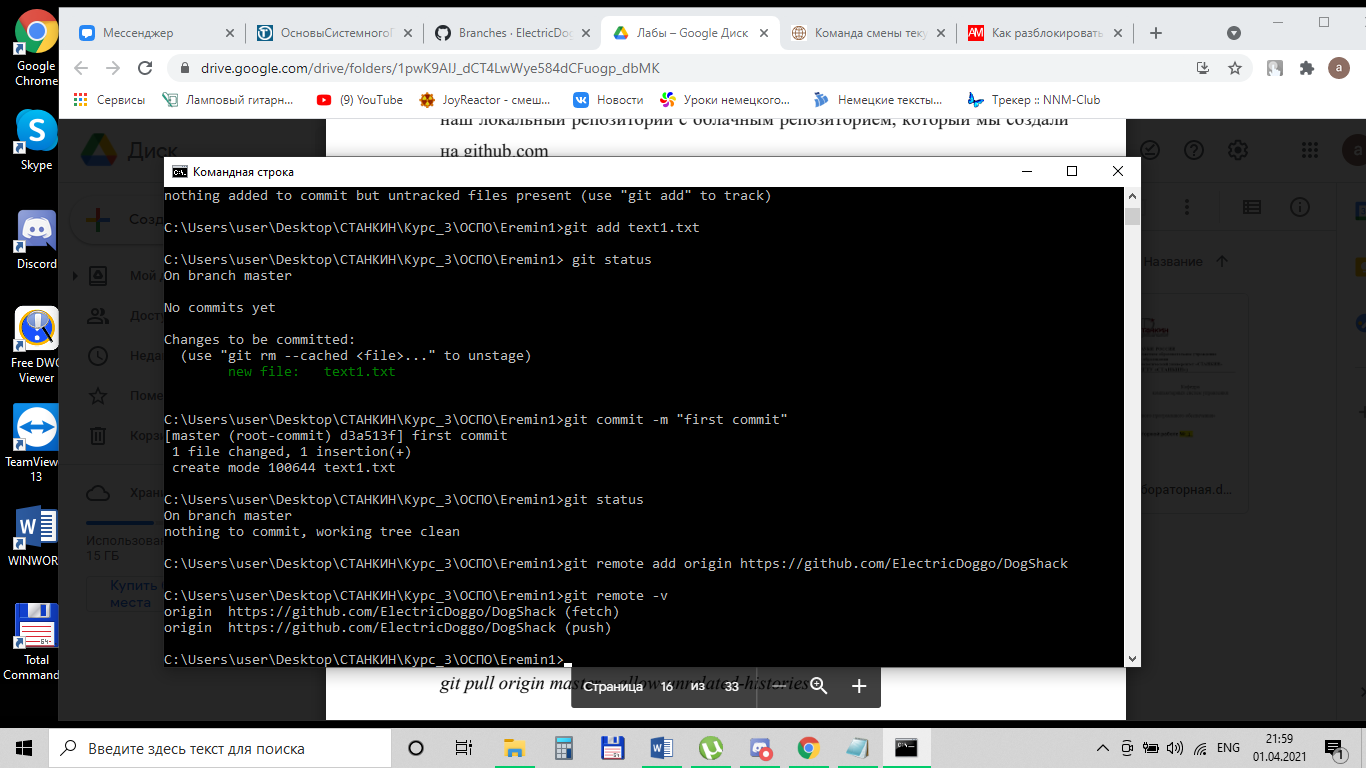
Закоммитим файл



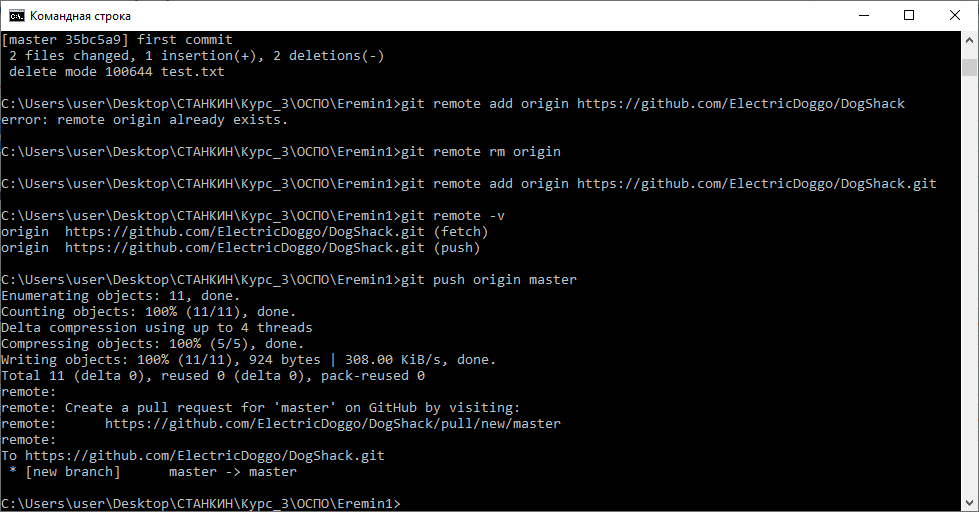


Свяжем локальный репозиторий с созданным на Гит (git remote add origin “URL репозитория”)

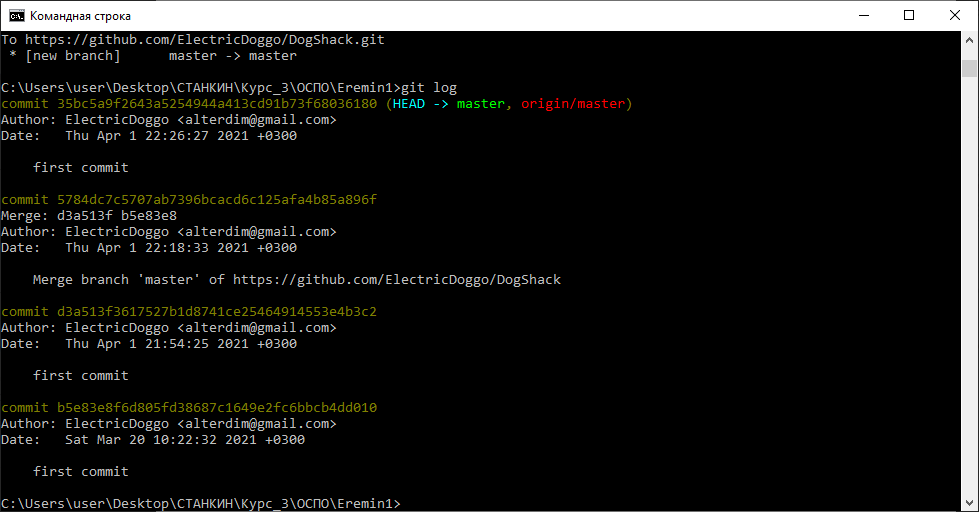
Припомощи команды ниже посмотрим, к какому проекту мы подключены



Запушим файл

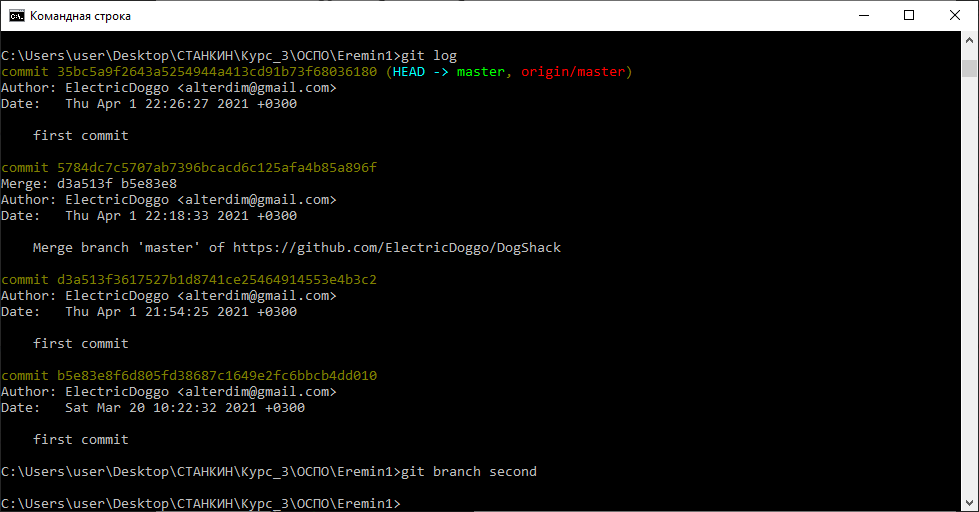


Проверим состояние, увидим коммиты

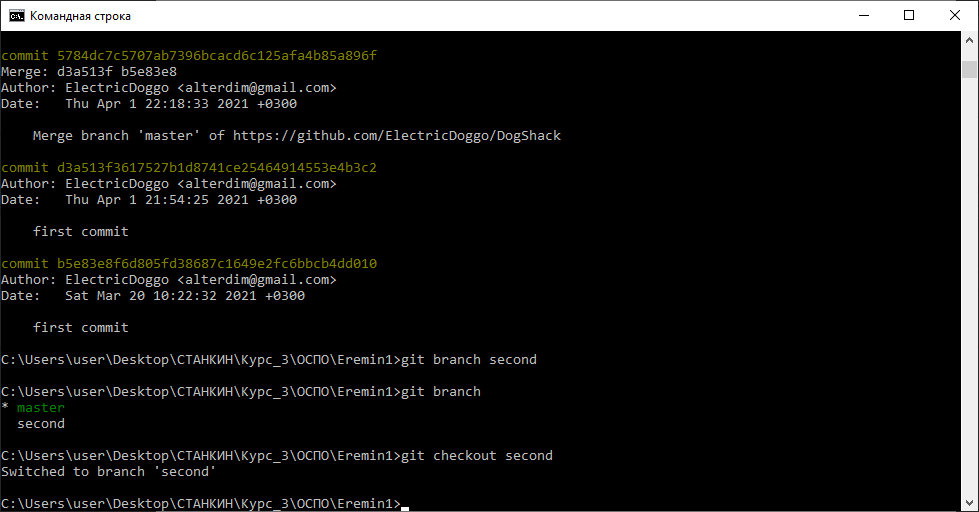


Так же все изменения можно посмотреть на GitHub

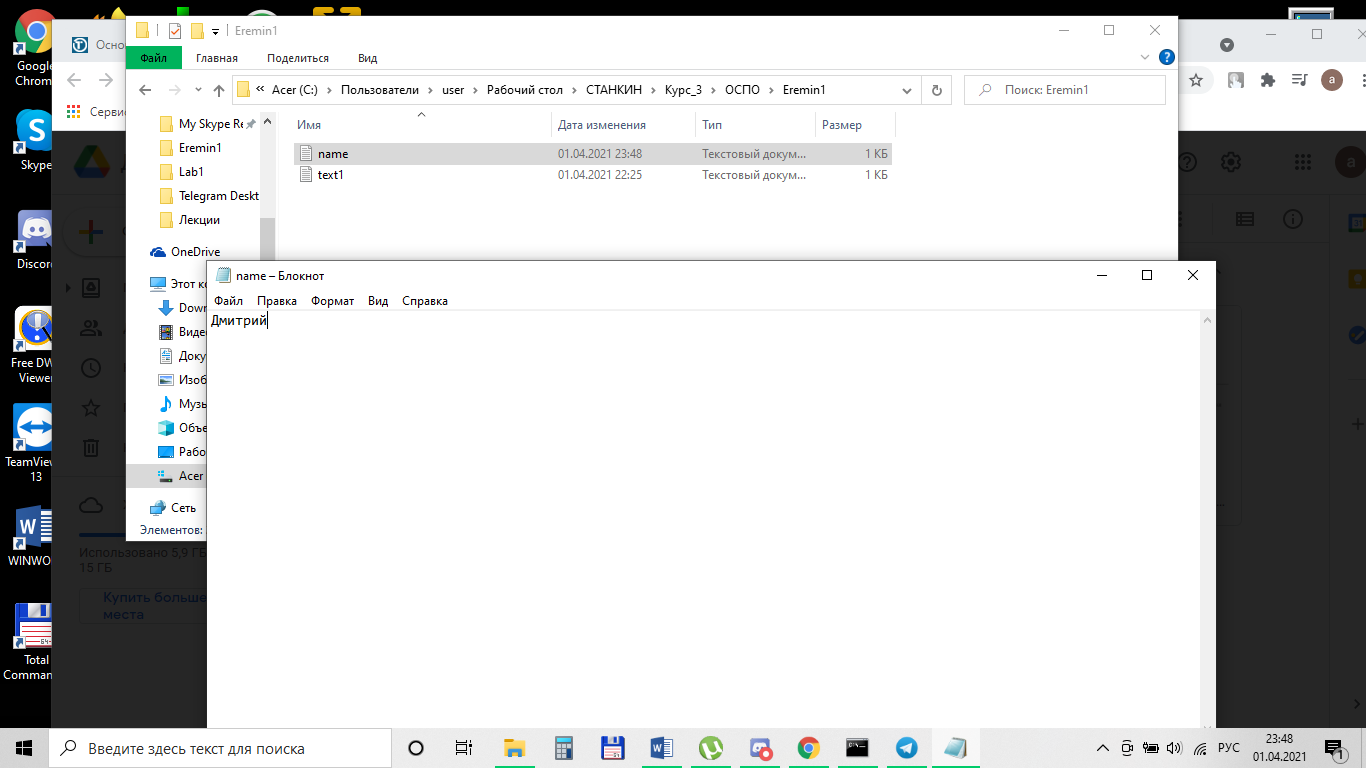
Далее займемся созданием новой ветки. Ветка с названием second



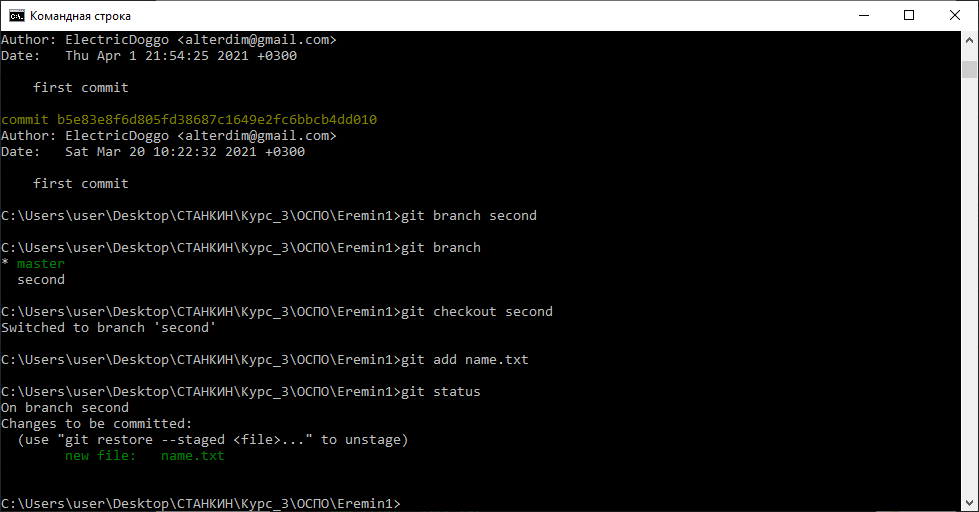
Ниже видно 2 ветки. Переключаемся на ветку second

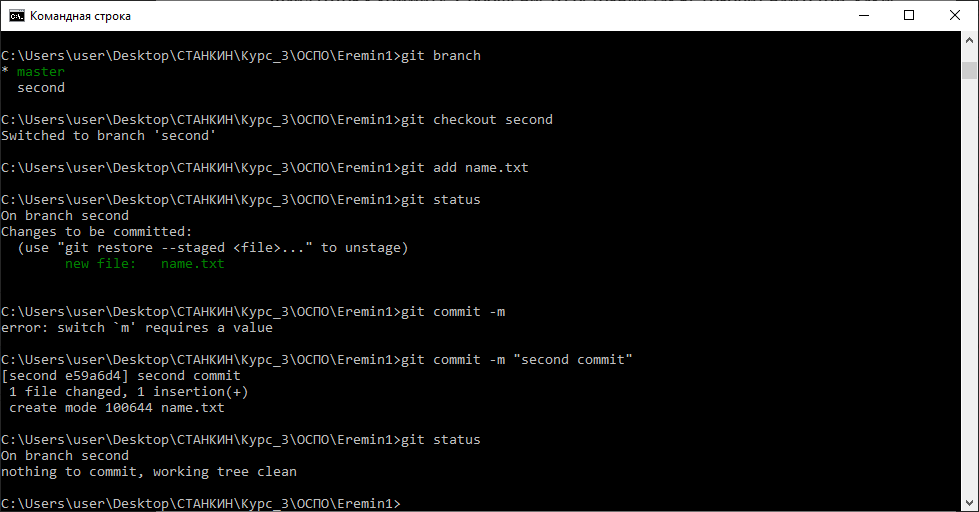


Создаем на локальной машине файл с именем

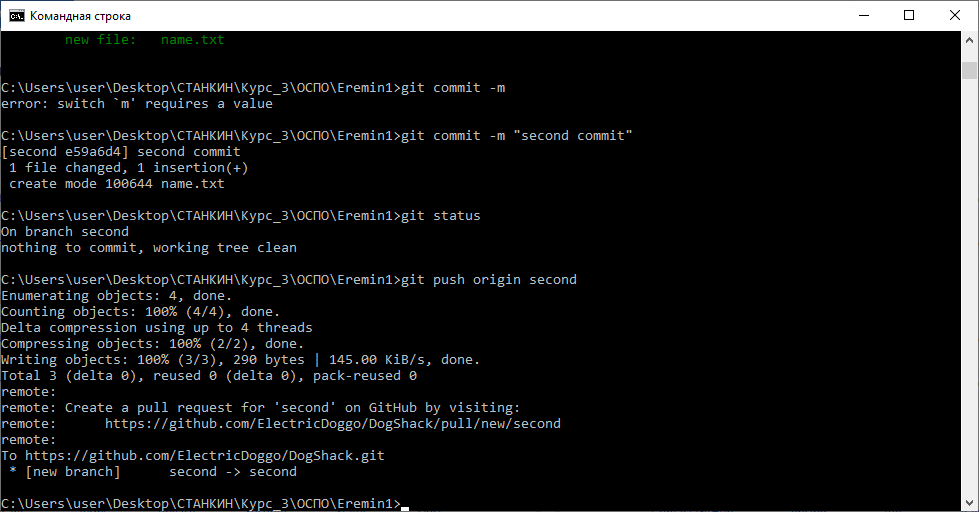


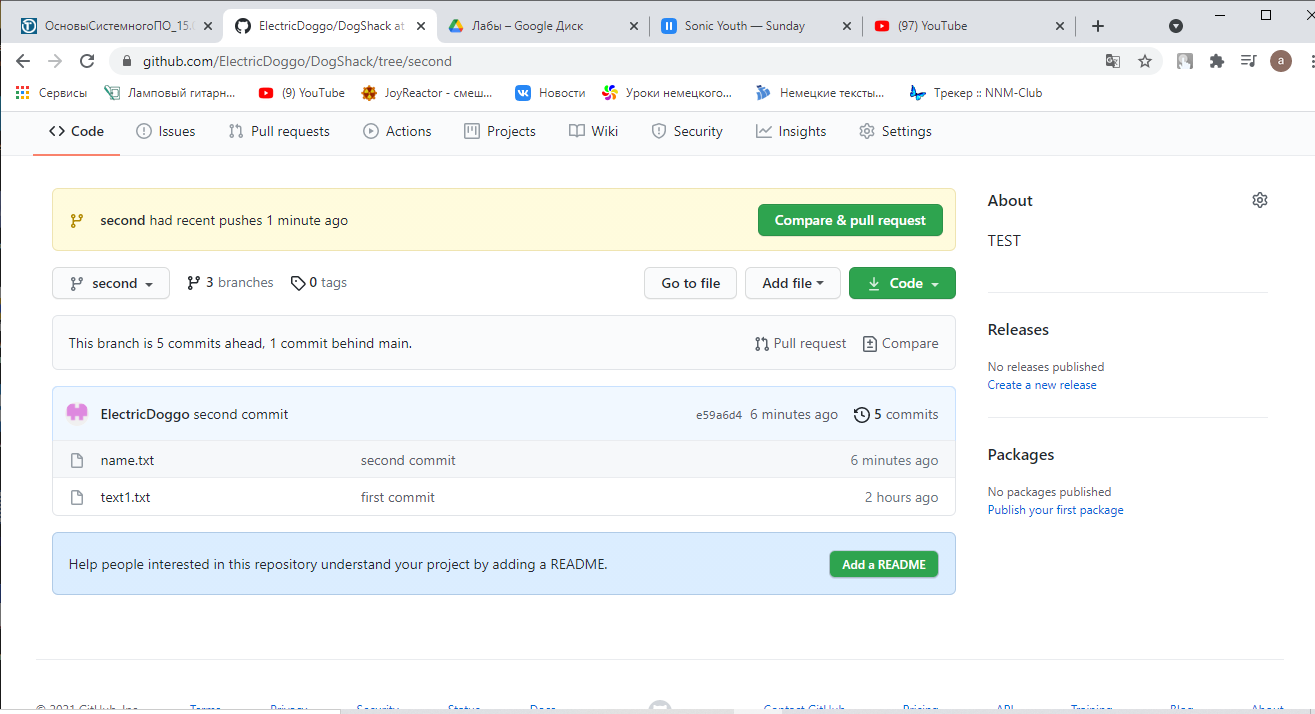
Добавляем этот файл и коммитим его



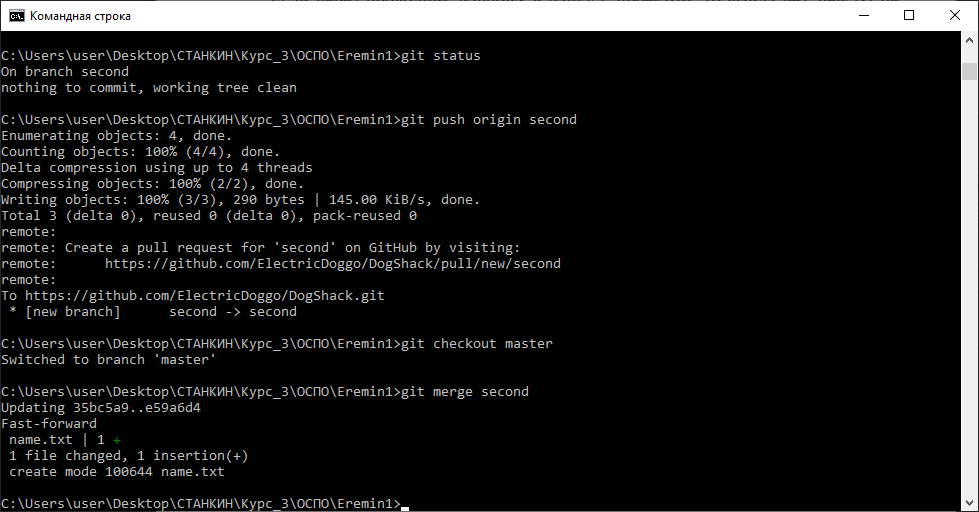


Теперь файл появился в ветке second

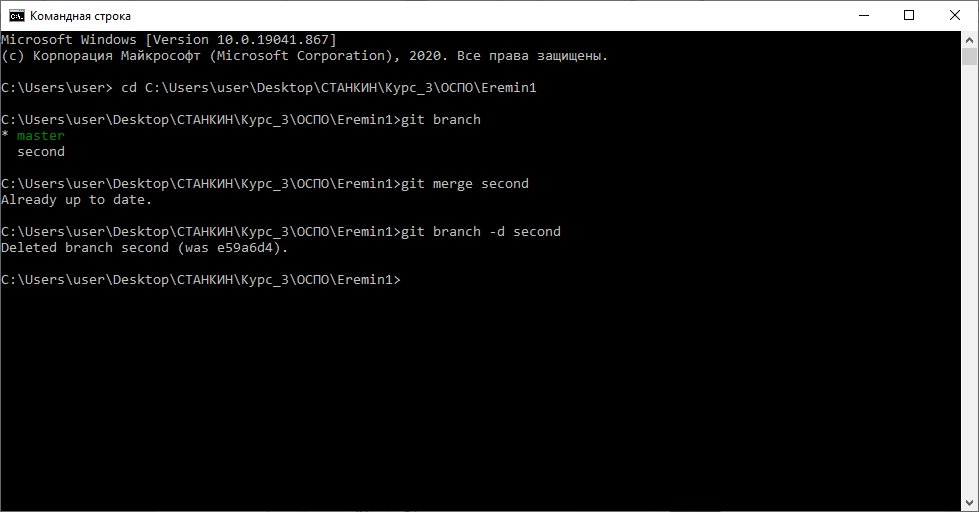




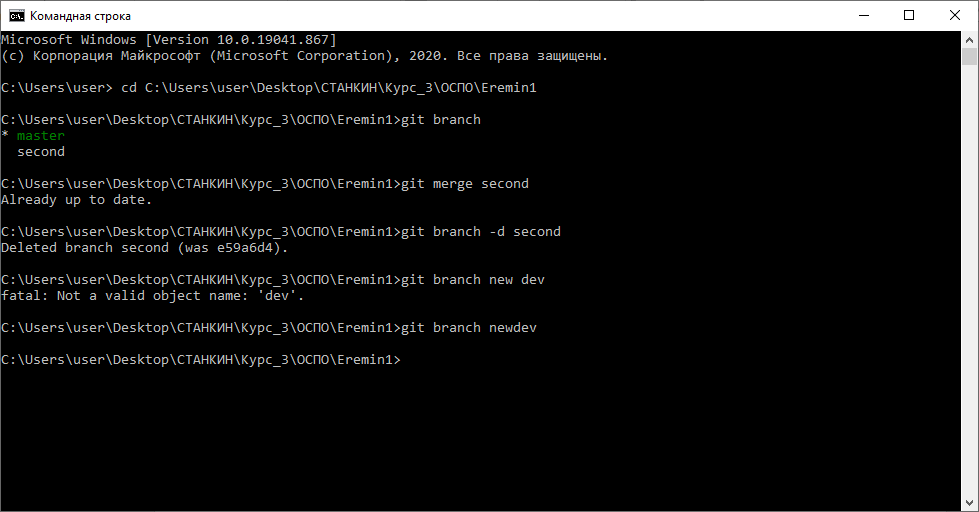
После займемся слиянием веток. Для этого переключимся на ветку master и воспользуемся командой merge



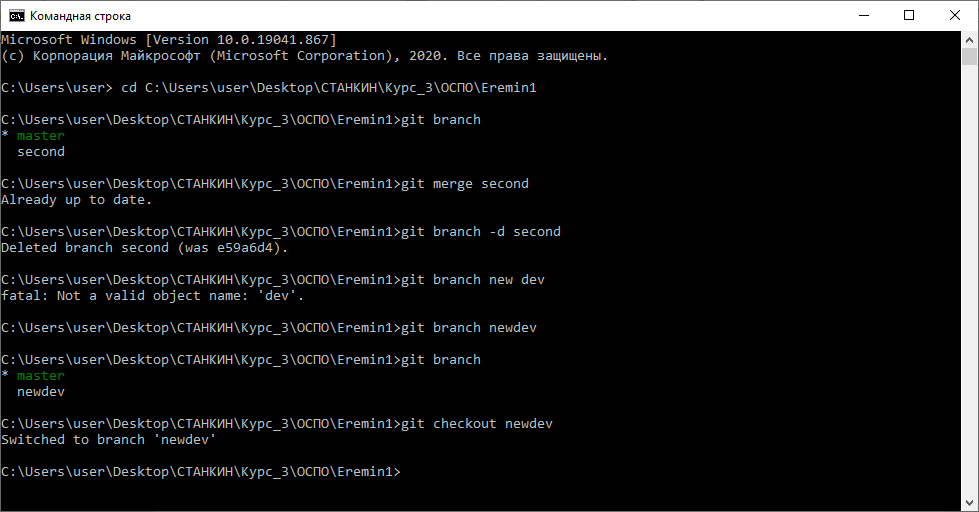
После займемся слиянием веток. Для этого переключимся на ветку master и воспользуемся командой merge



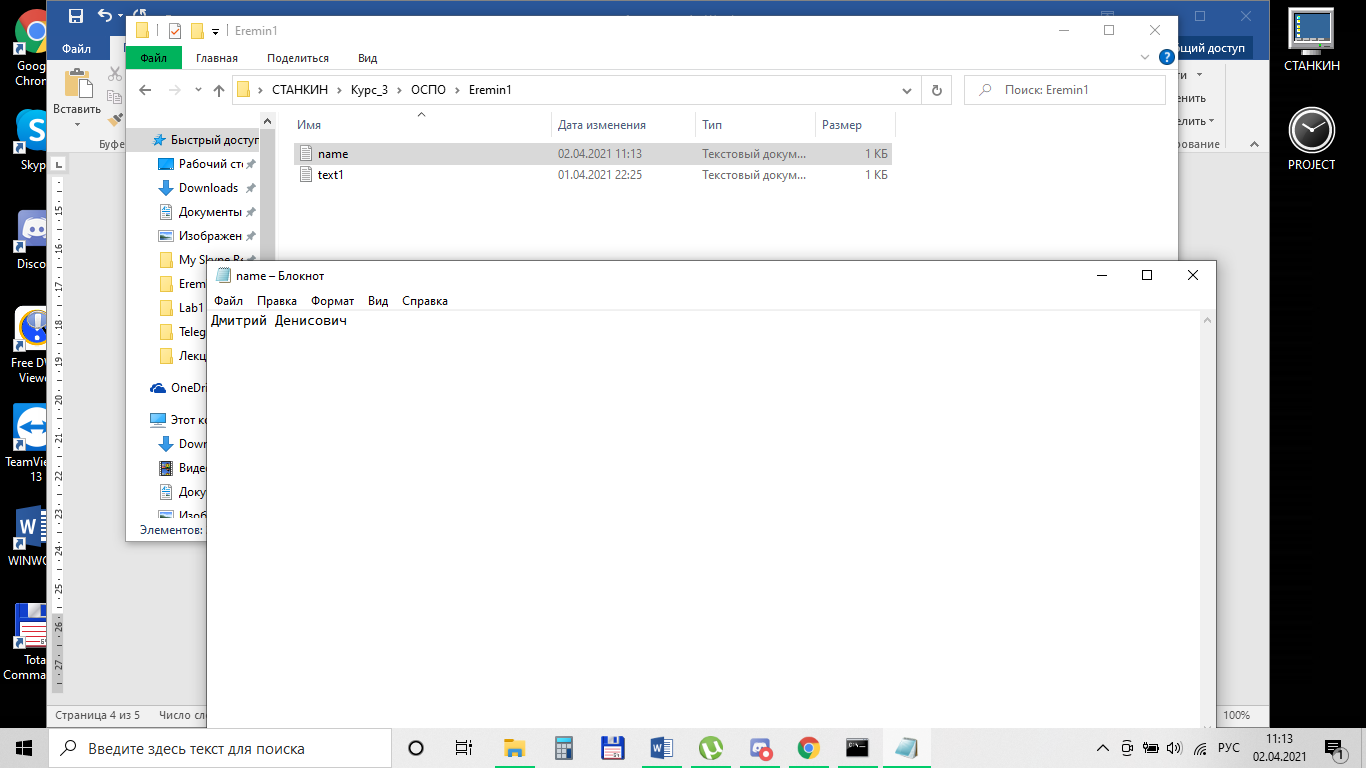
Усложним задачу. Создадим ветку newdev



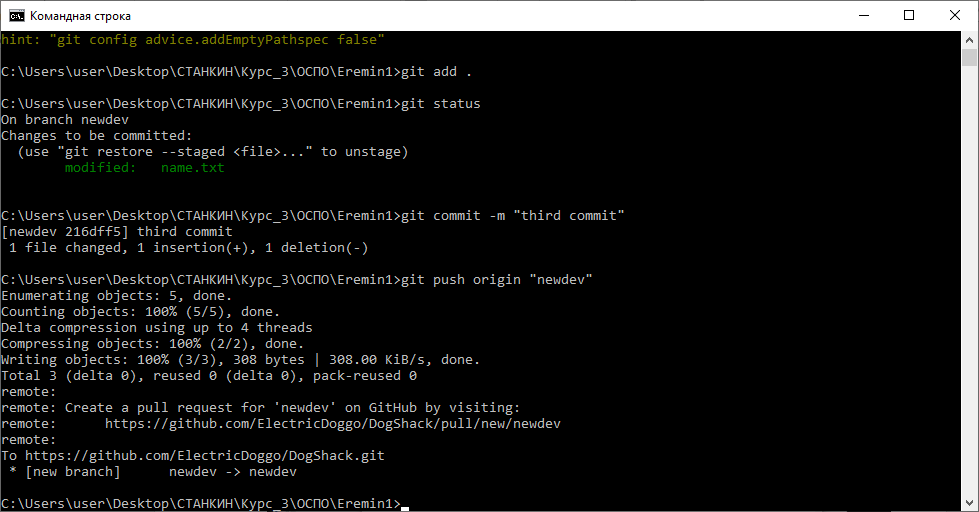
Переключимся на newdev



Добавляем в файл с именем отчество

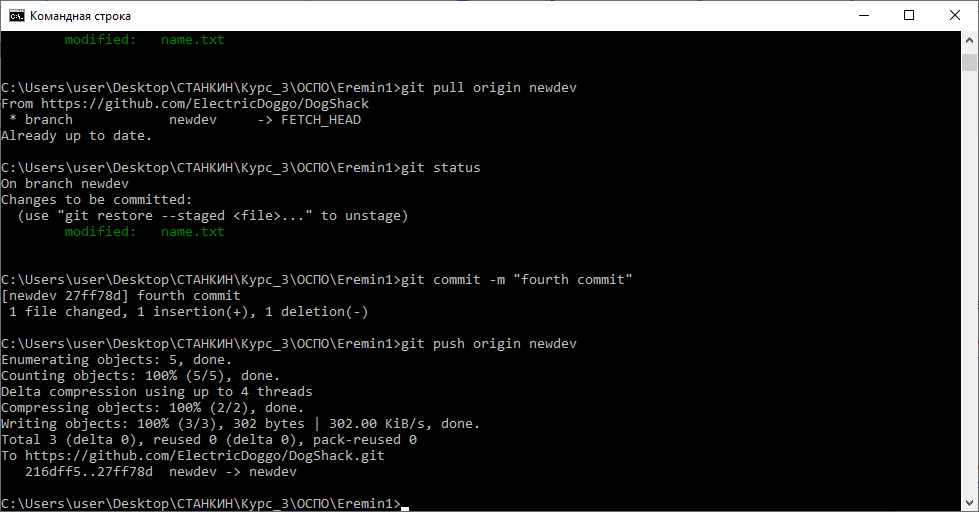
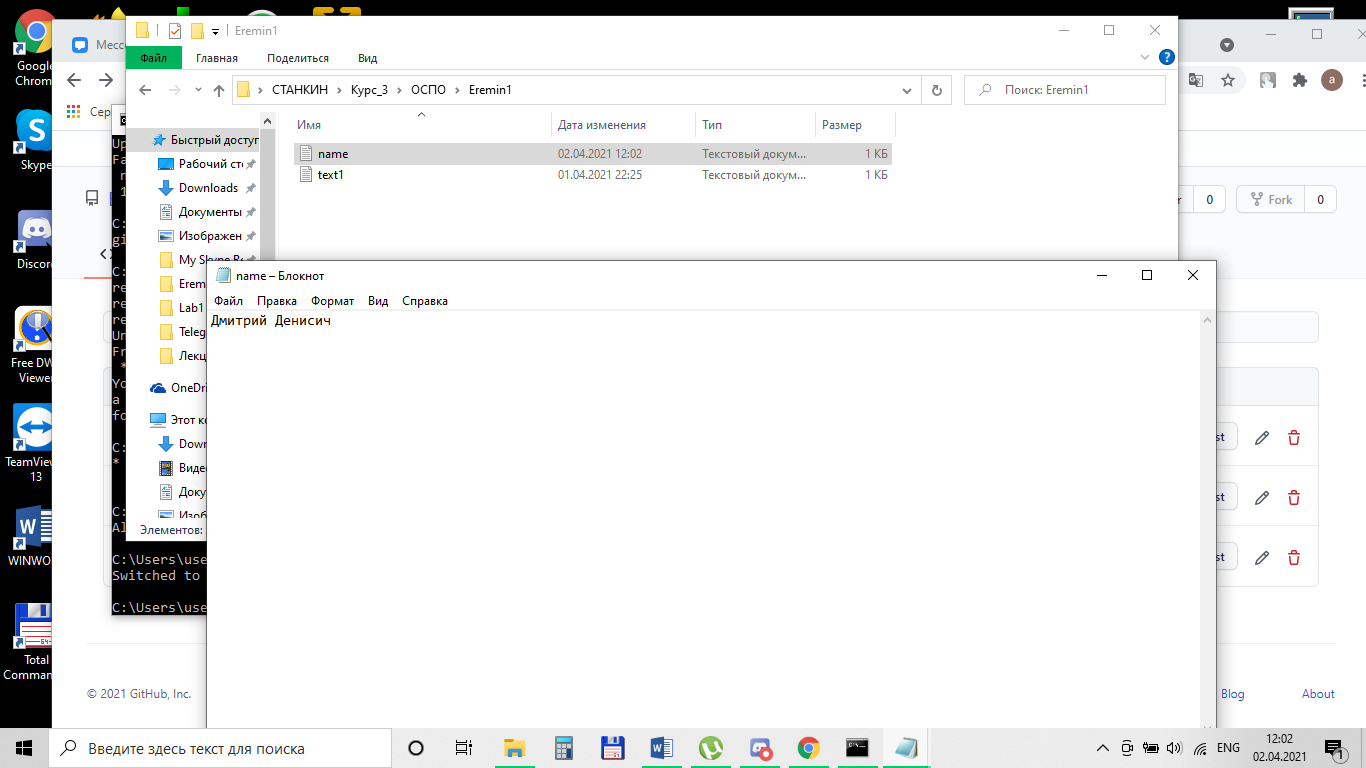


Фиксируем и коммитим изменения

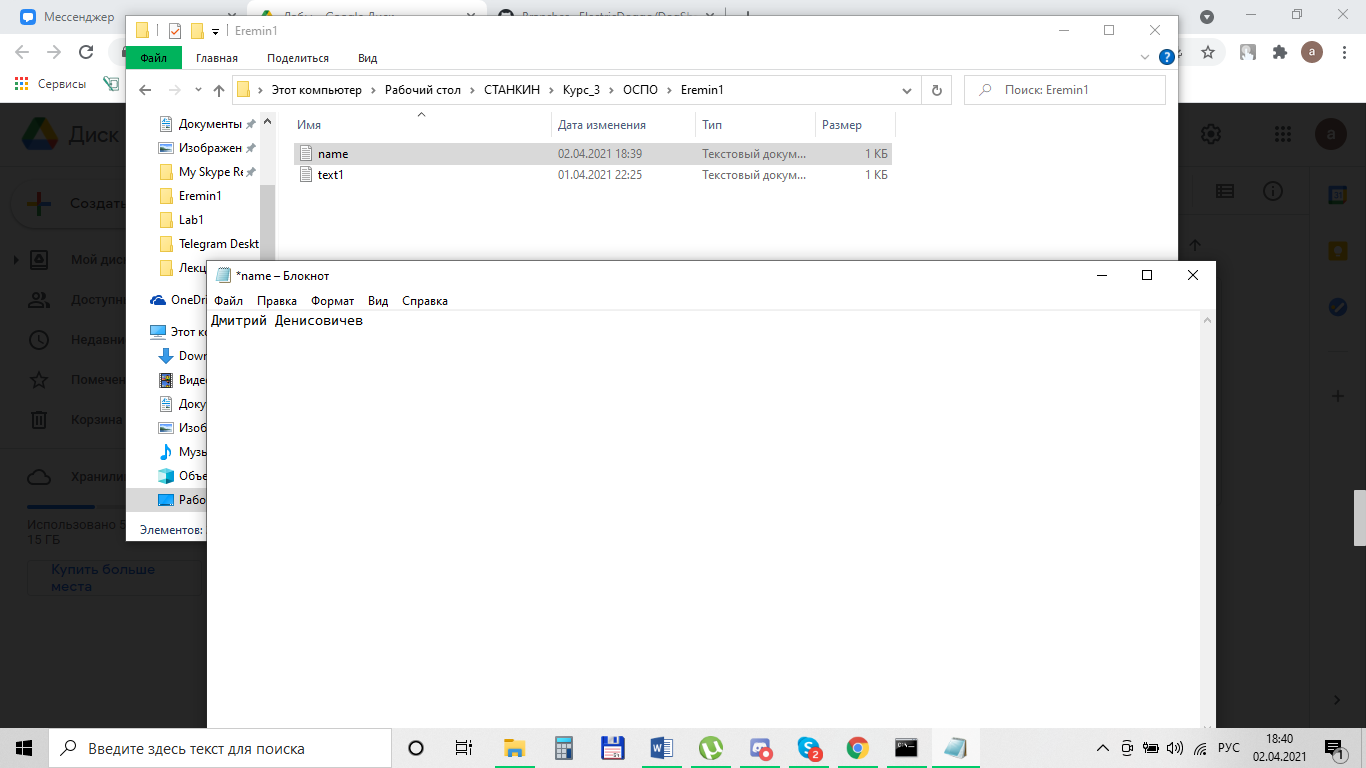


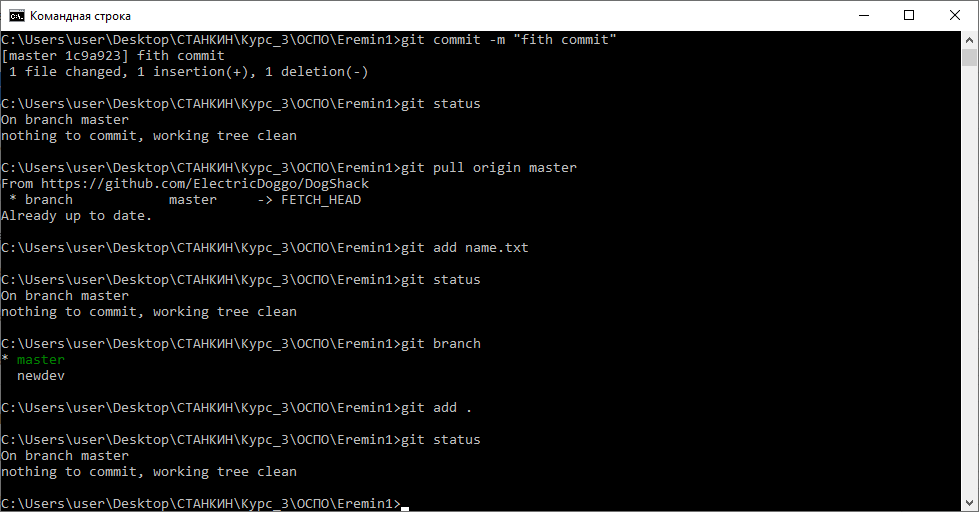
При этом, если переключиться на ветку master, в файле будет только имя

Теперь переключимся на ветку master, и удалим из отчества несколько букв, зафиксируем изменения

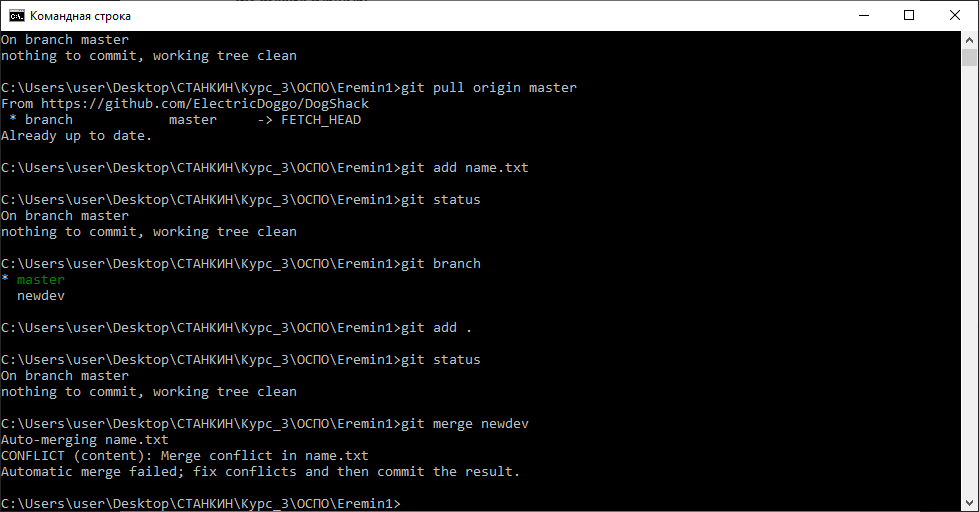


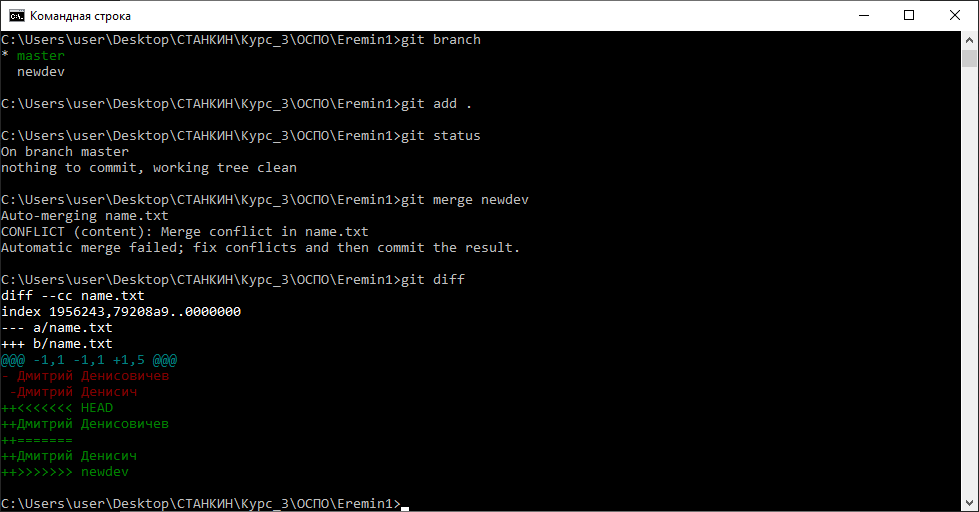
А в ветке newdev добавим несколько новых букв и зафиксируем изменения



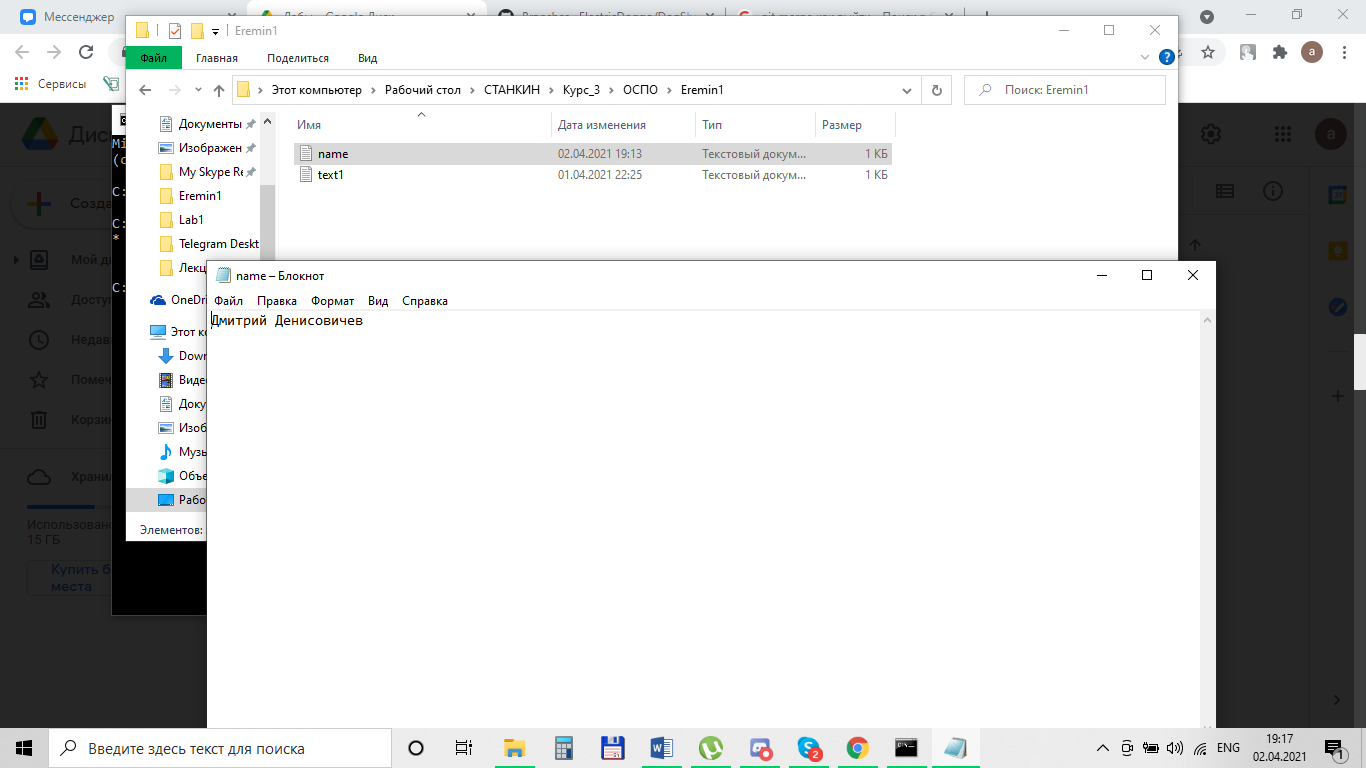
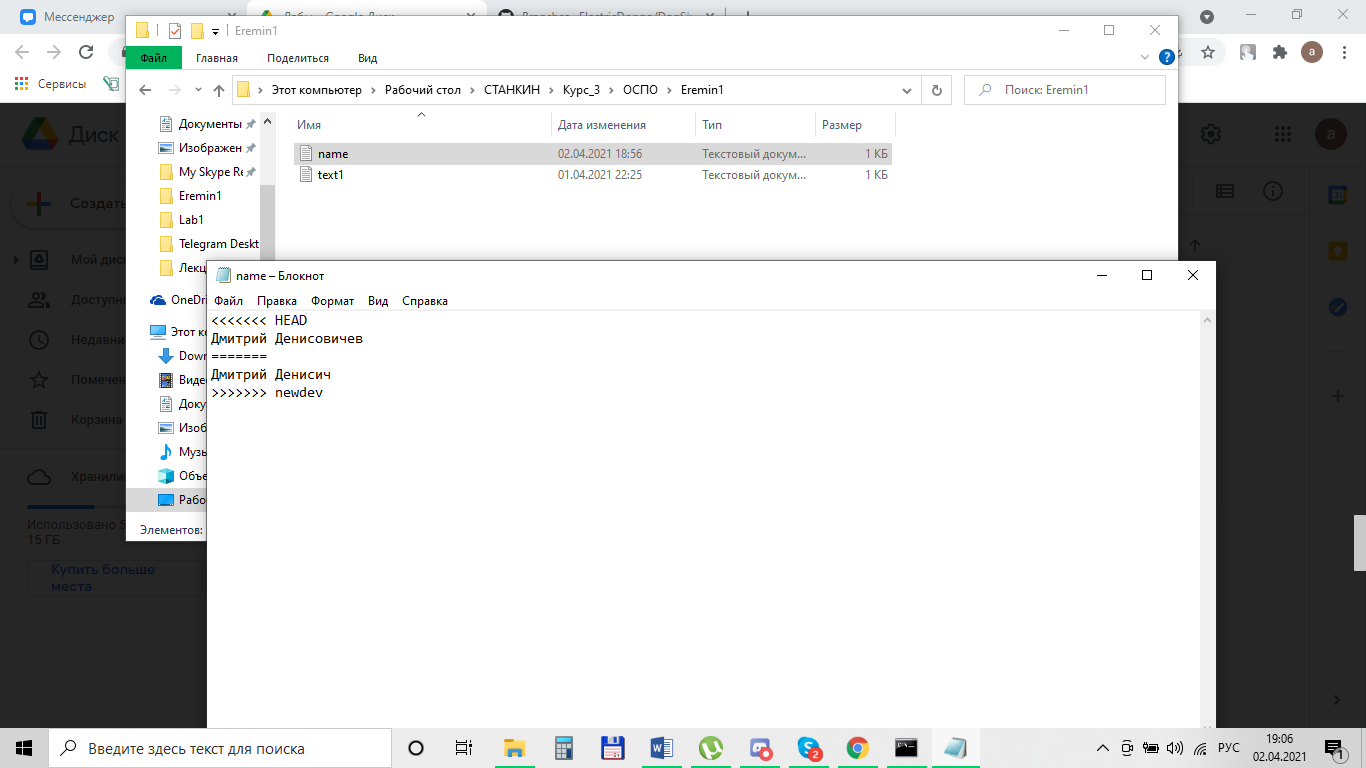


Попробуем слить 2 ветки, получим конфликт, т.к. есть изменения в обеих ветках

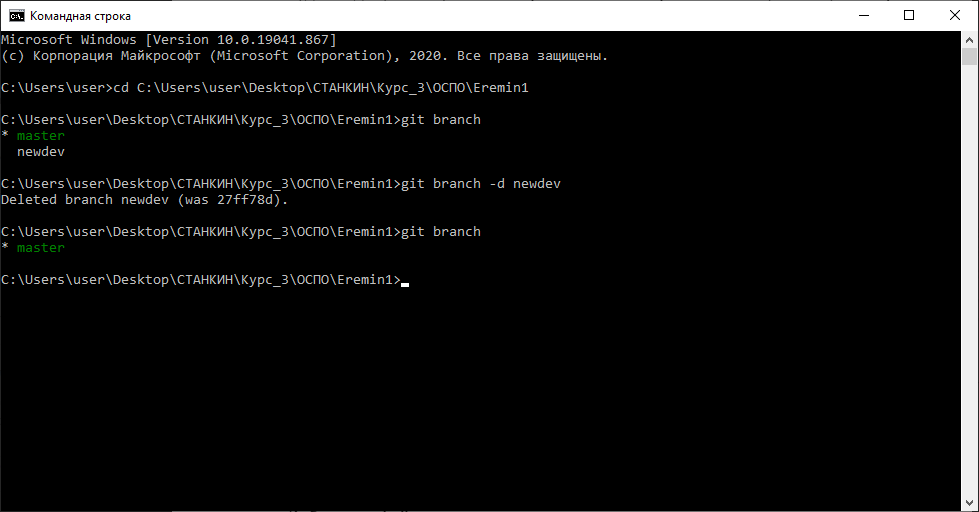




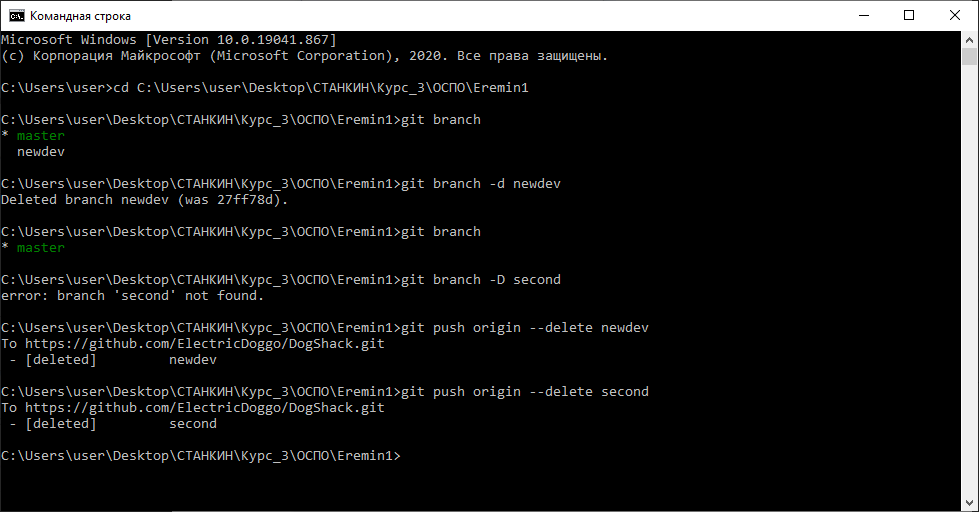
Для разрешения конфликта заходим в файл с именем и отчеством. Выбираем нужную строку (какую строку оставить, может вообще все хотим оставить), удаляем разделители, сохраняем файл, фиксируем изменения, коммитим, отправляем на сервер

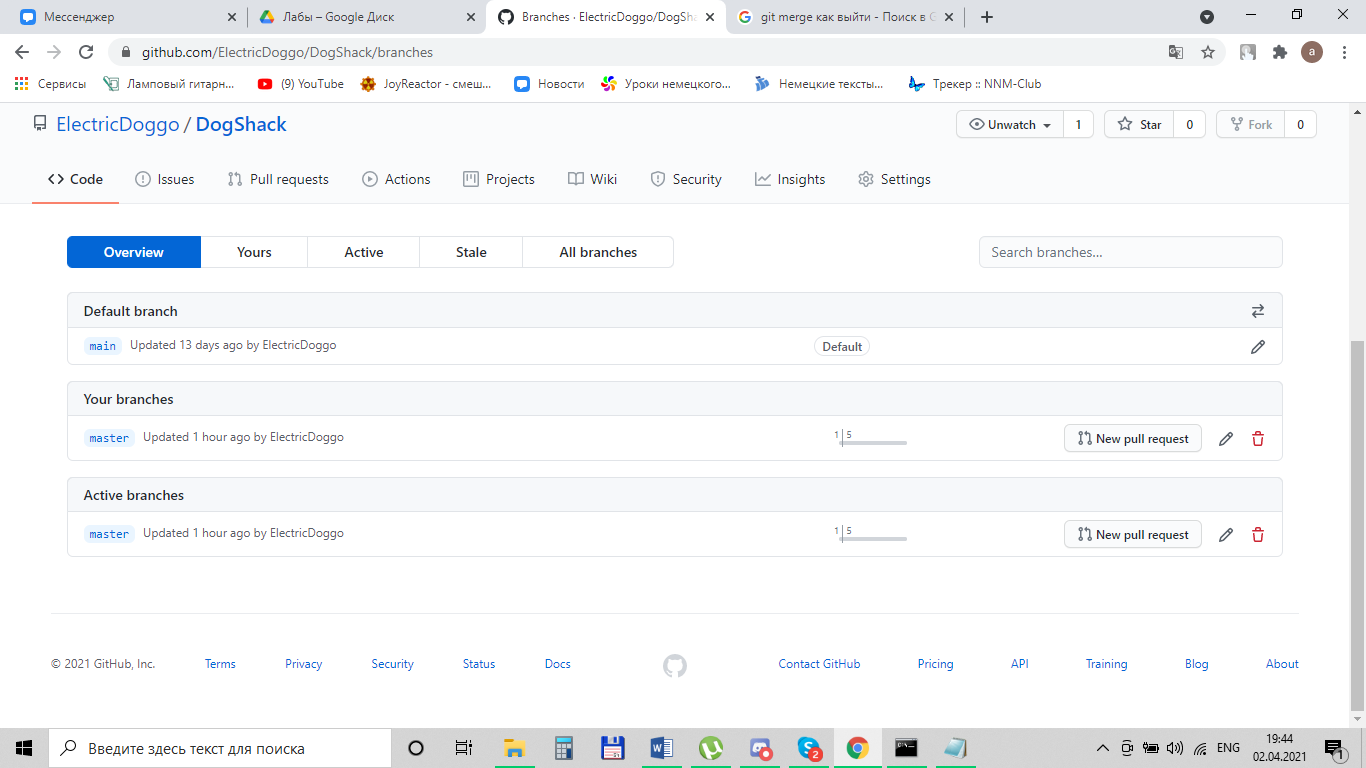


Далее просто удаляем ставшую ненужной ветку newdev

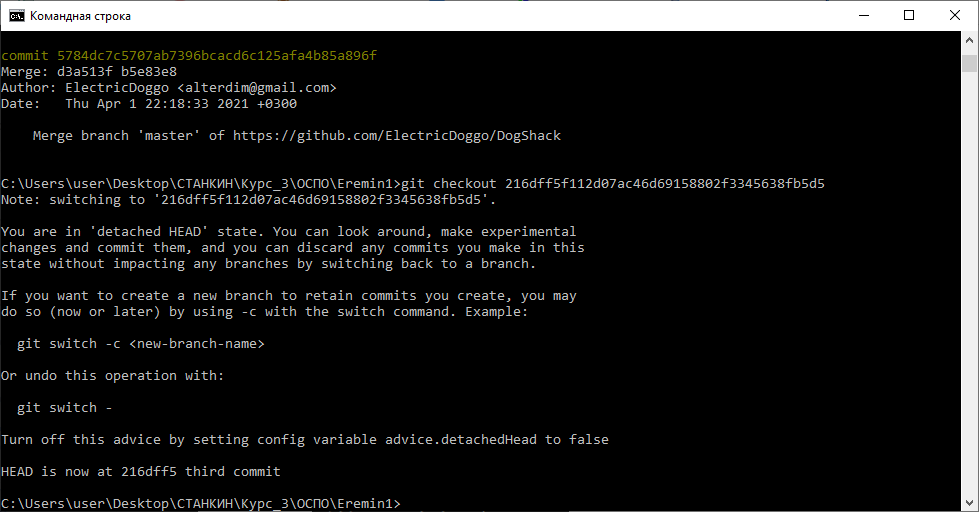


Удалим ненужные ветки из github. Если зайти на страницу, увидим, что из веток осталасть только main

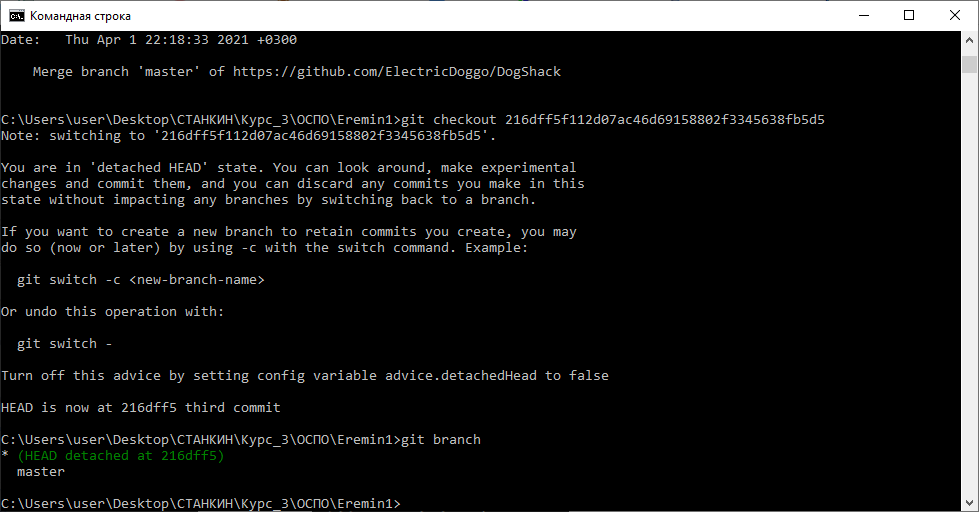




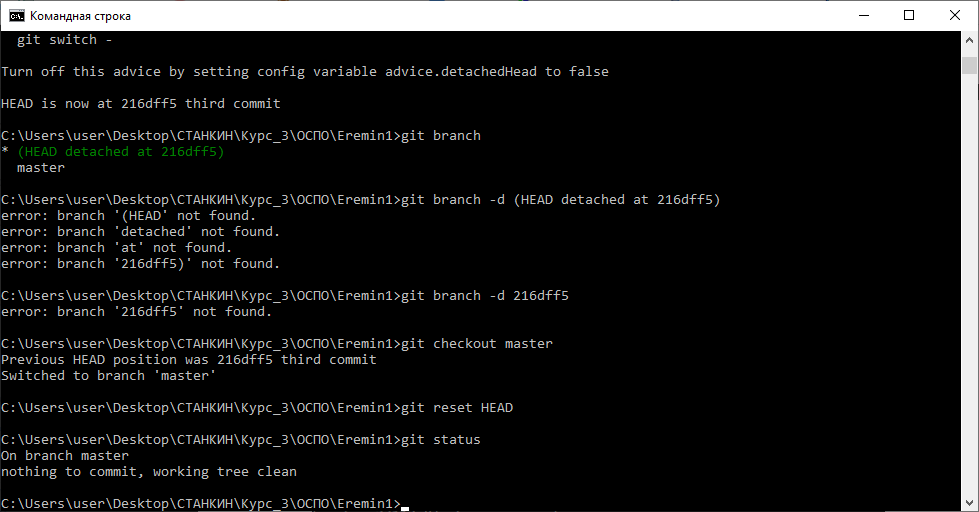
Далее откатимся к исходному состоянию. Посмотрим все коммиты



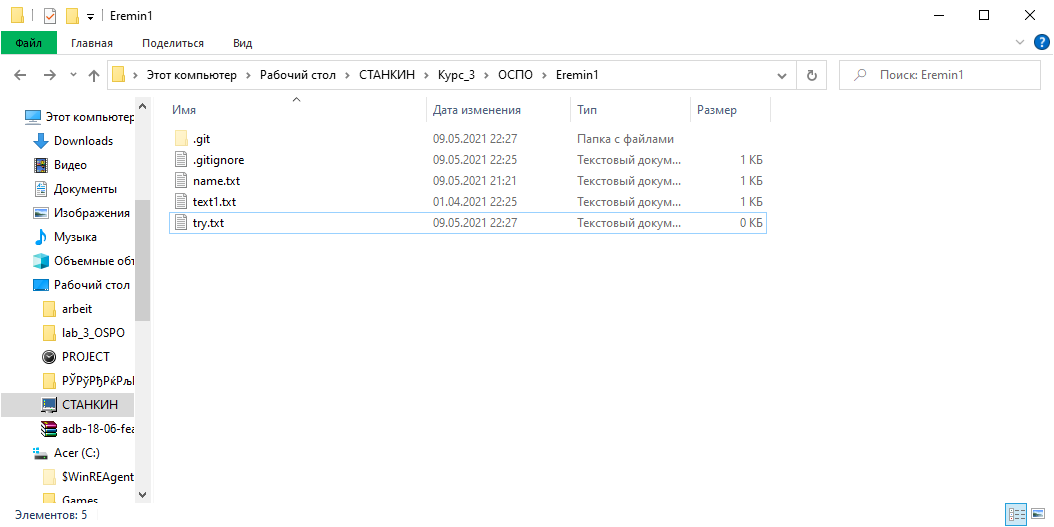
Выбираем версию, к которой желаем откатиться и создаем псевдо ветку



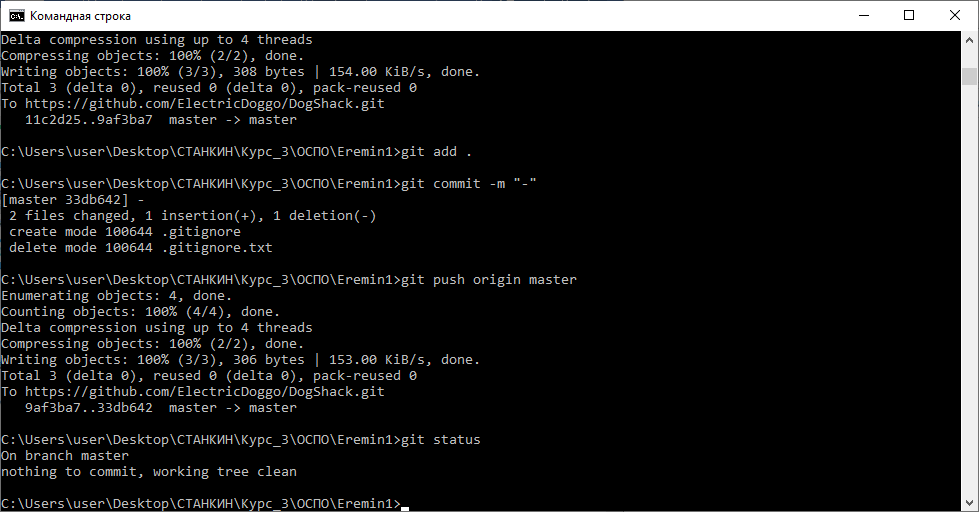
Откатываемся при помощи reset head

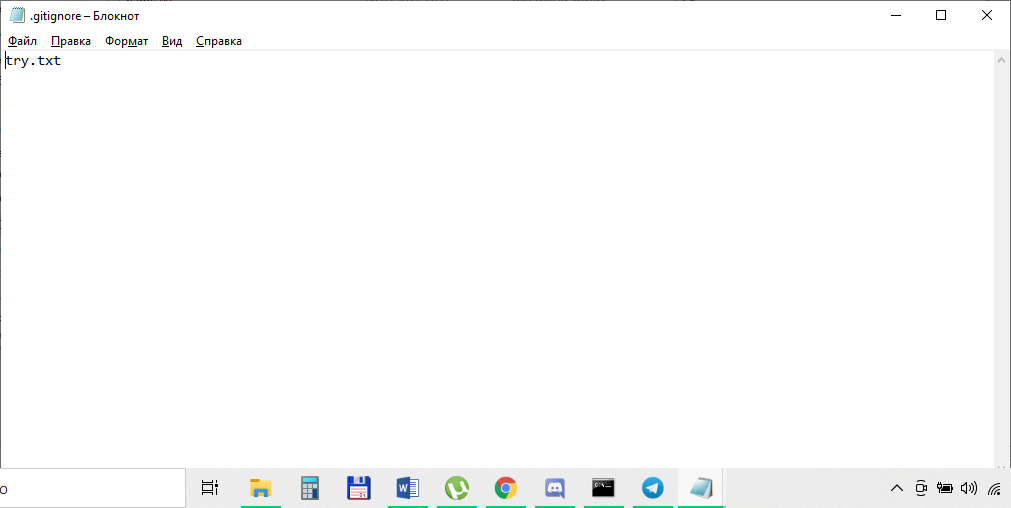


Далее приступим к изучению файла .gitignore. Он позволяет добавить в себя названия файлов, которые не должны быть закоммичены. Для изучения создадим файл таким названием, добавим в него название «нежелательного» файла и посмотрим при помощи git status состояние локального репозитория

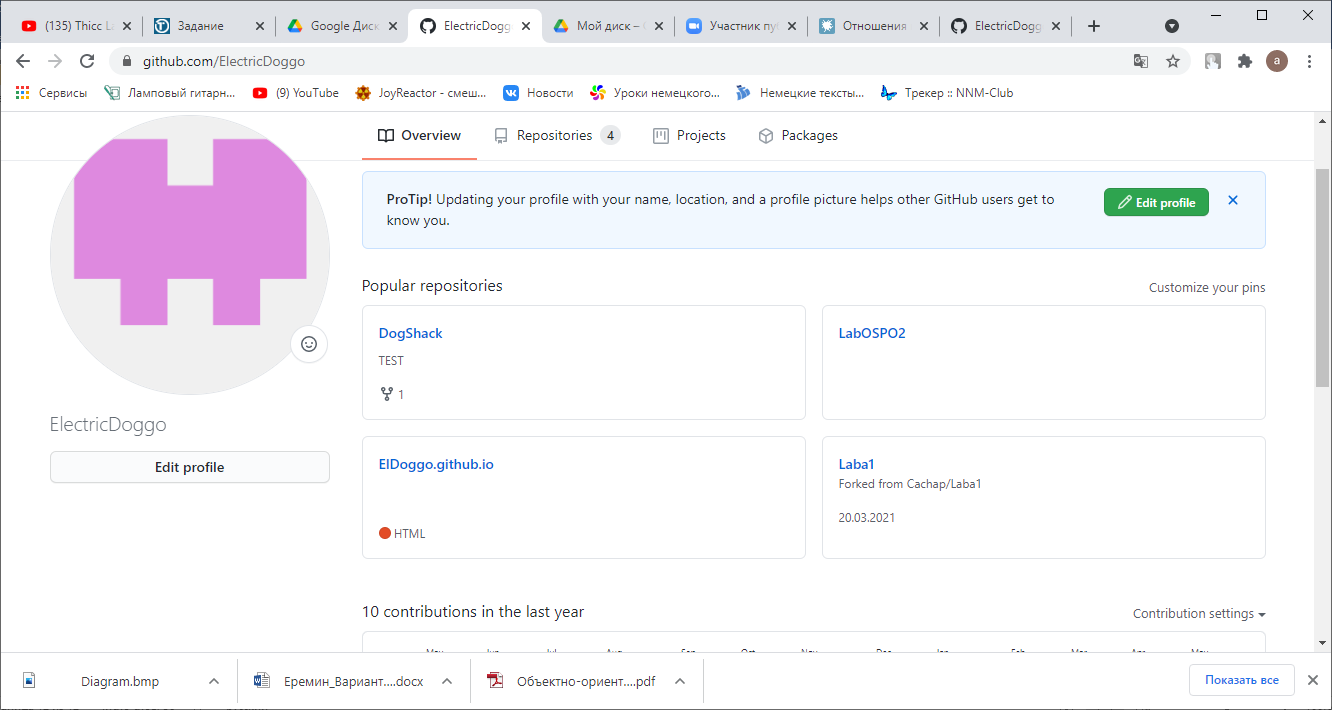
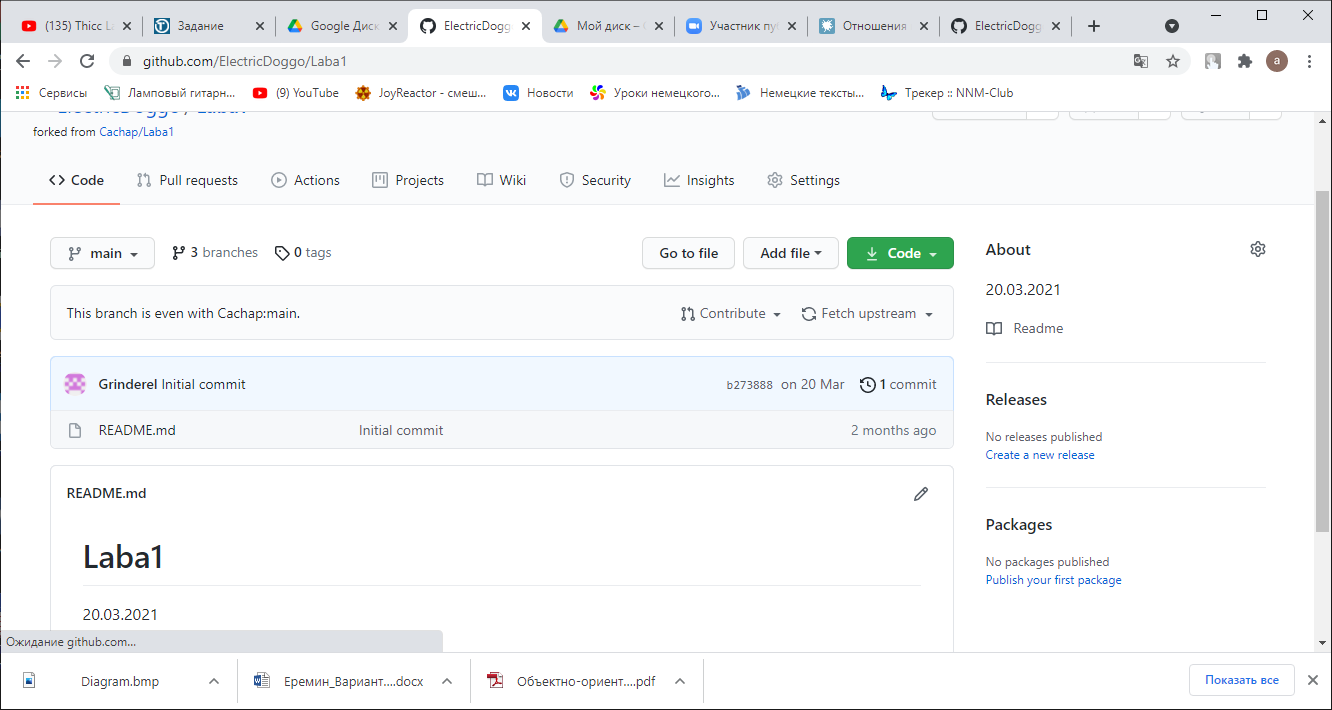
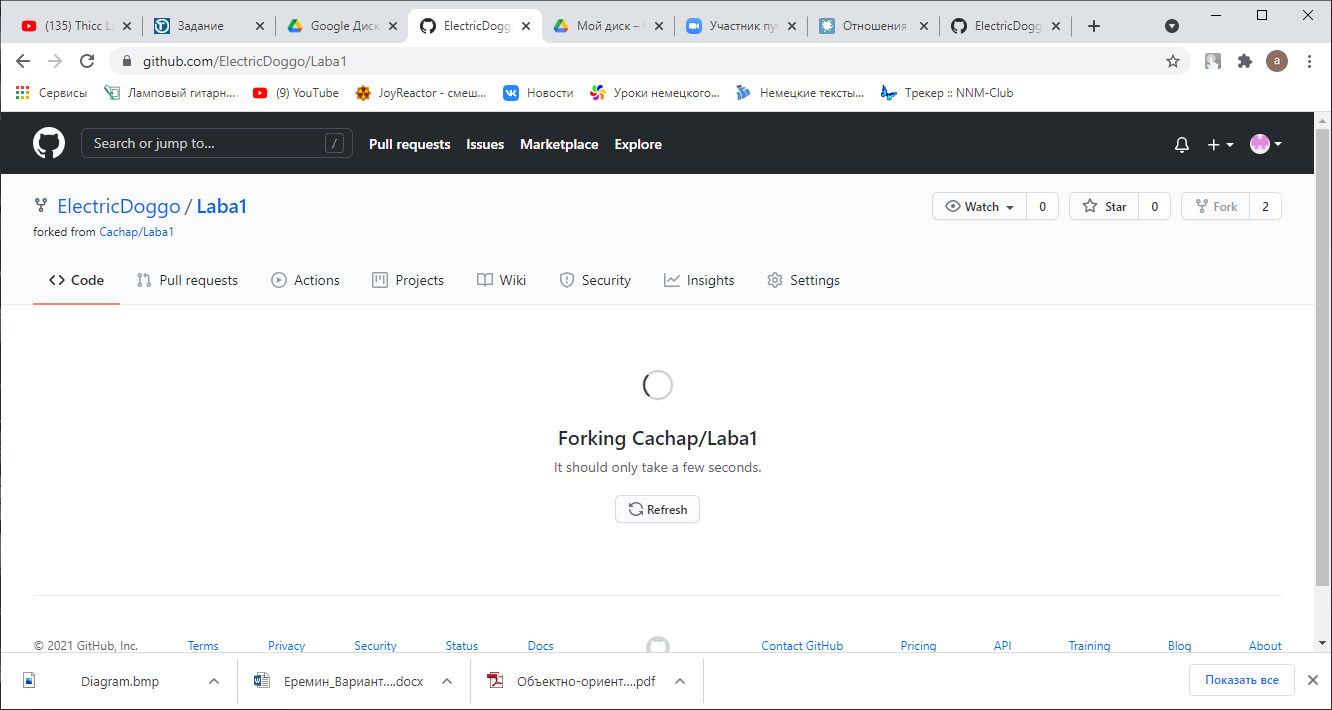


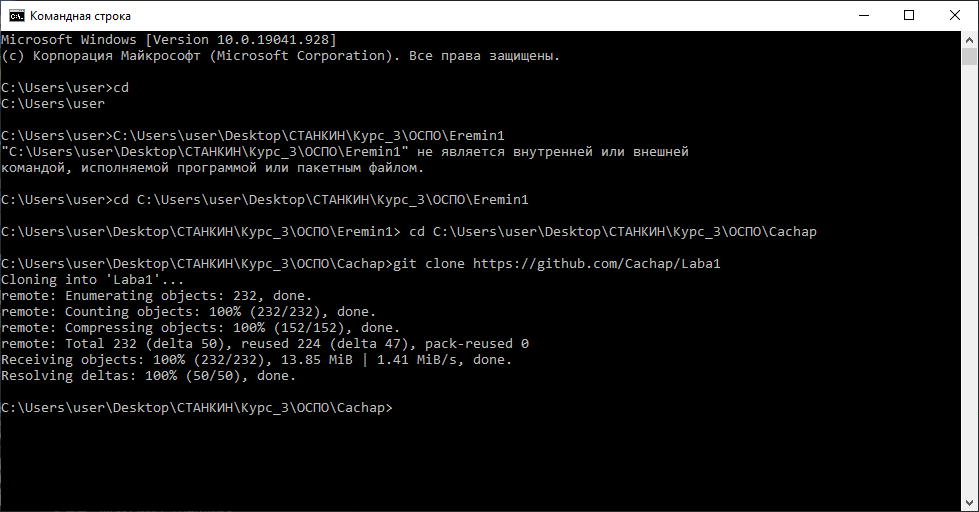
Как видно из рисунка, нужный файл игнорируется



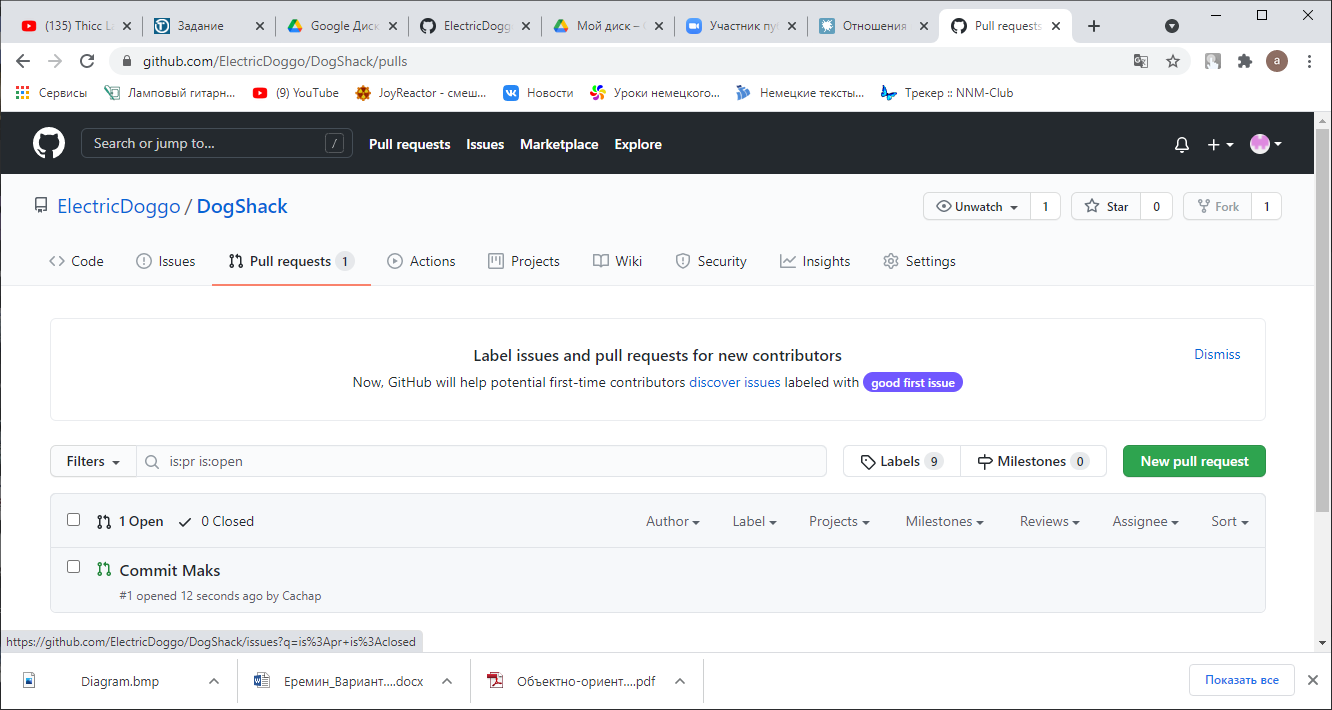


Далее займемся совместной работой с напарником. Отправим друг другу адреса репозиториев и нажимаем на кнопку fork.



После этого создаем папку и клонируем туда репозиторий напарника

После, даем доступ напарнику к репозиторию. Теперь он может вносить изменения в репозиторий. После того, как он запушил изменения, нужно принять pull request



На слайде ниже видны изменения

