Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет  
 информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина: Избранные главы информатики

ОТЧЕТ

к лабораторной работе

на тему

Система контроля версий Git

БГУИР КП 1-40 04 01

                                                Выполнила: студентка группы 253502,

                              Лишик Ксения Ярославовна

                                                 Проверила: Жвакина Анна Васильевна

Минск 2024

**Цель**: Познакомиться с возможностями и получить практические навыки работы с системой контроля версий Git.

**Задание 1.** Подготовьте рабочее окружение в соответствии с типом вашей операционной системы (Windows), установите Git, выполните базовую настройку.

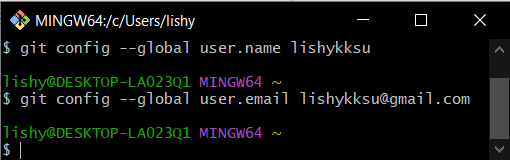


Рисунок 1 – Базовые настройки в Git Bash

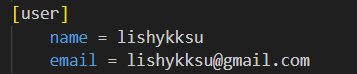


Рисунок 2 – Файл .gitconfig

**Задание 2.** Создайте структуру папок (использовать командную строку) и поместите в репозиторий Git на своем локальном компьютере:

253502\_LISHYK\_12

IGI

LR1 *(Папки и файлы в соответствии с индивидуальным заданием, отчет по ЛР1 со скринами действий и результатов)*

LR2

LR3

LR4

LR5

STRWEB

LR1

LR2

LR3

LR4

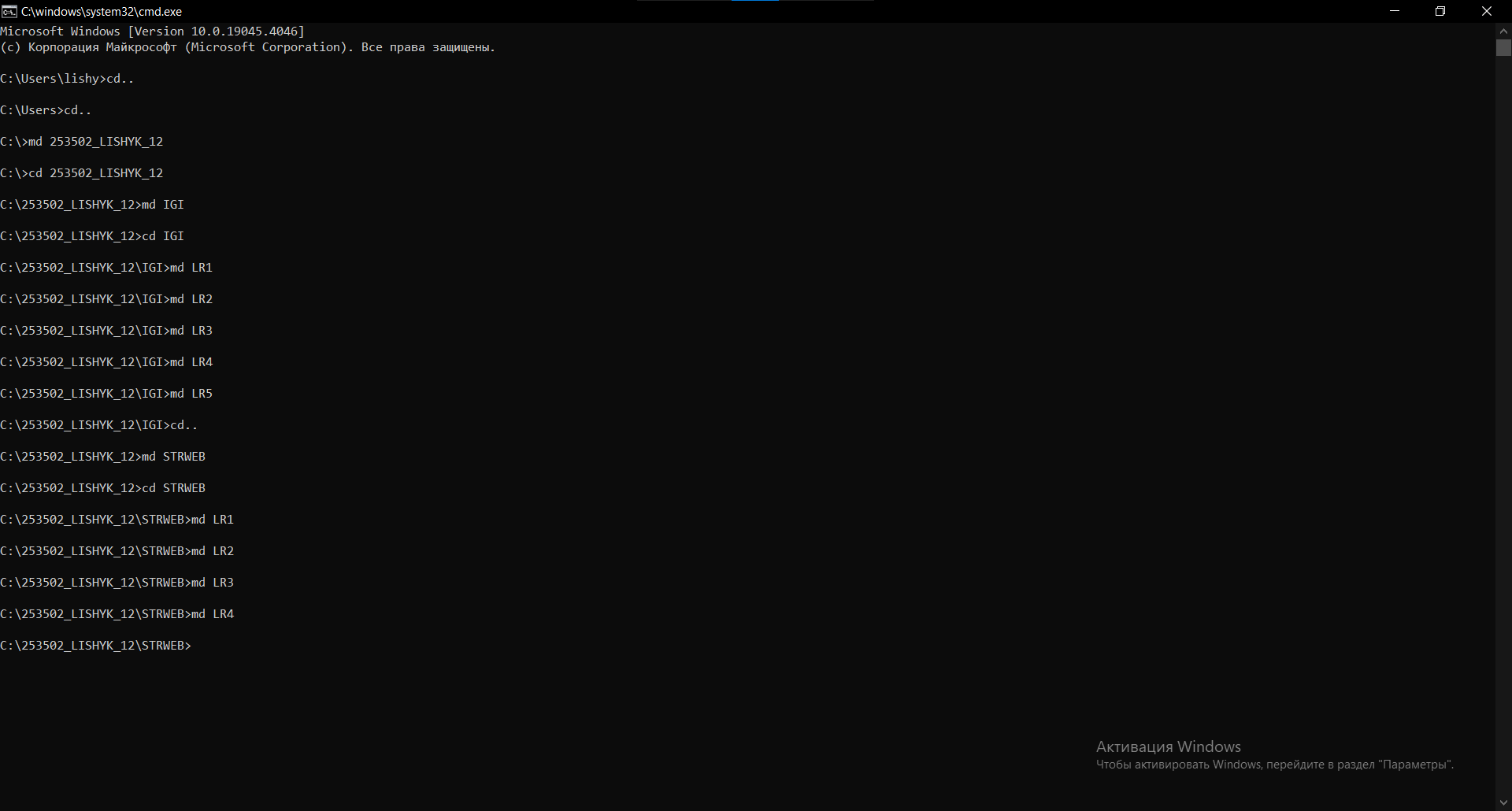


Рисунок 1 – Создание и переход в корневую папку

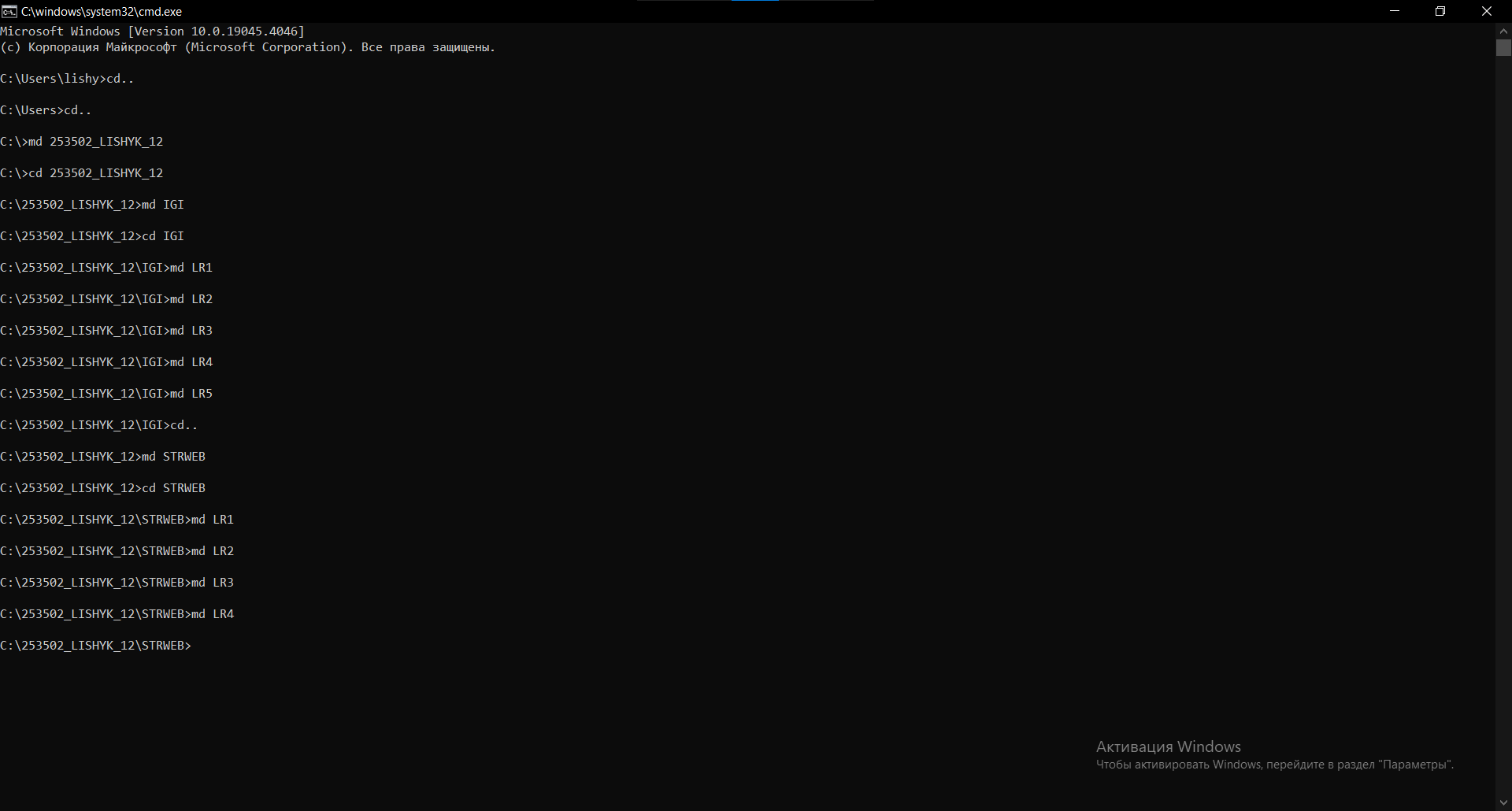


Рисунок 2 – Создание и переход в папку второго уровня

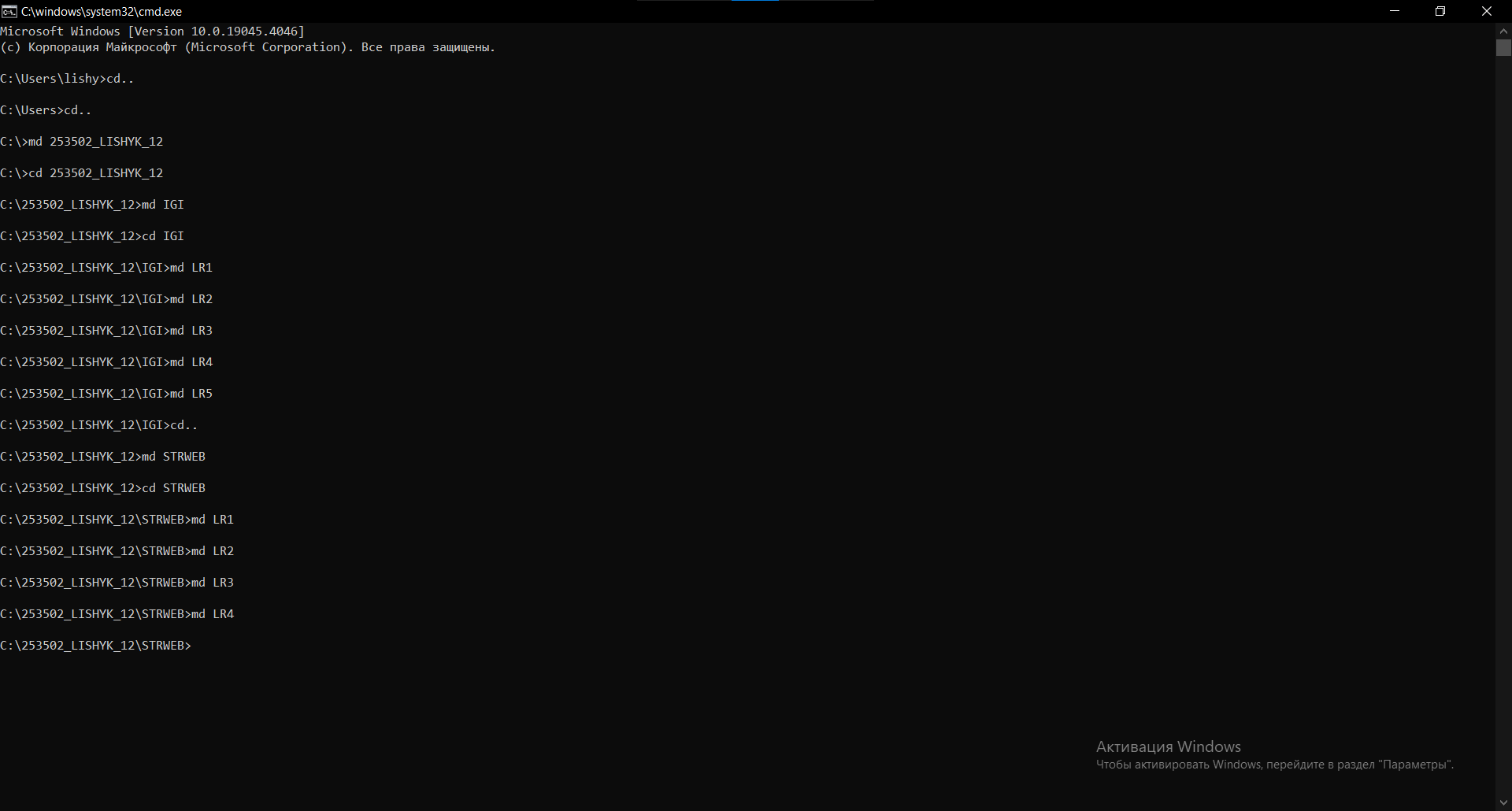


Рисунок 3 – Создание и переход в папки третьего уровня

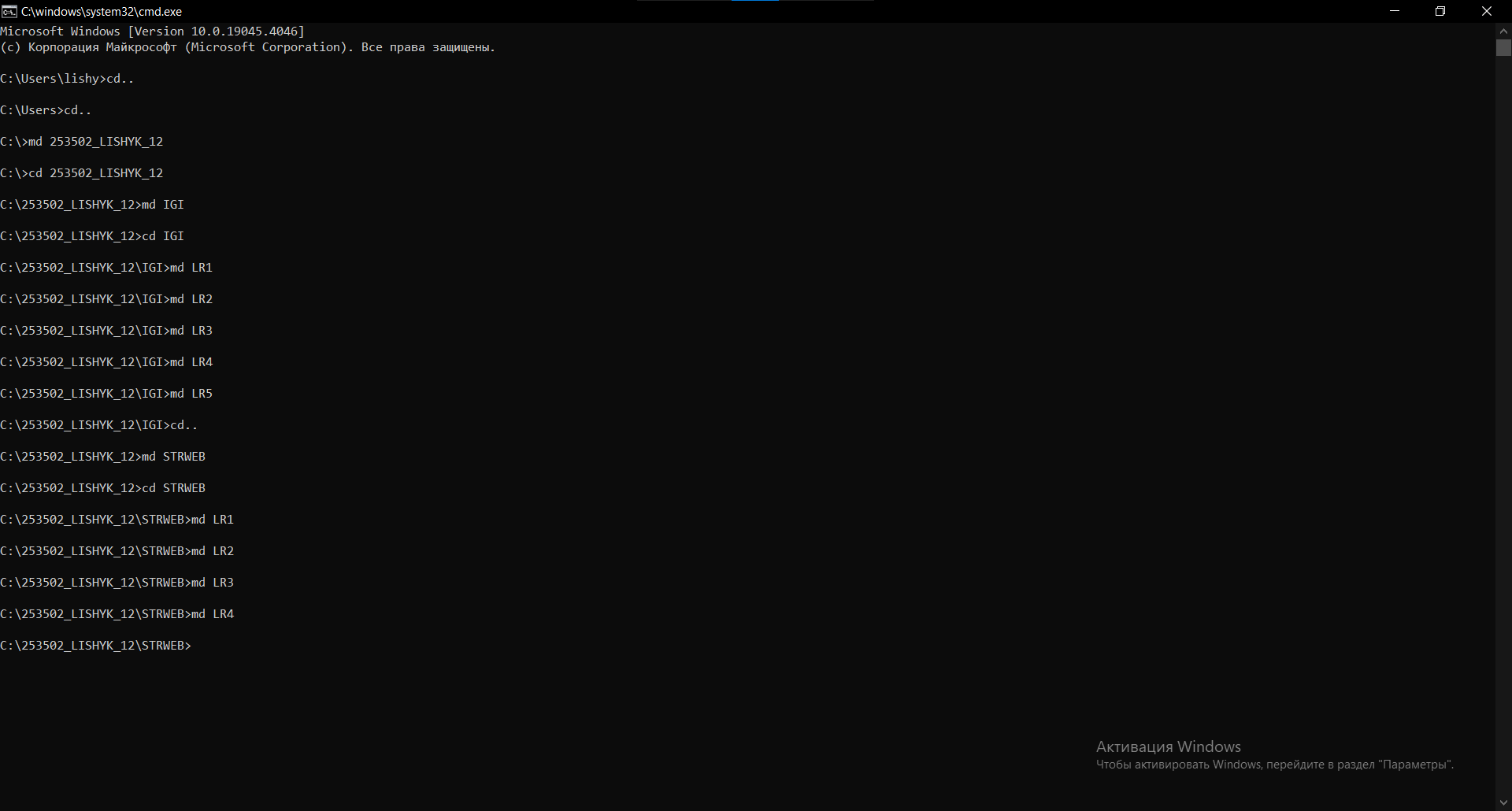


Рисунок 4 – Аналогичное создание папок второго и третьего уровня

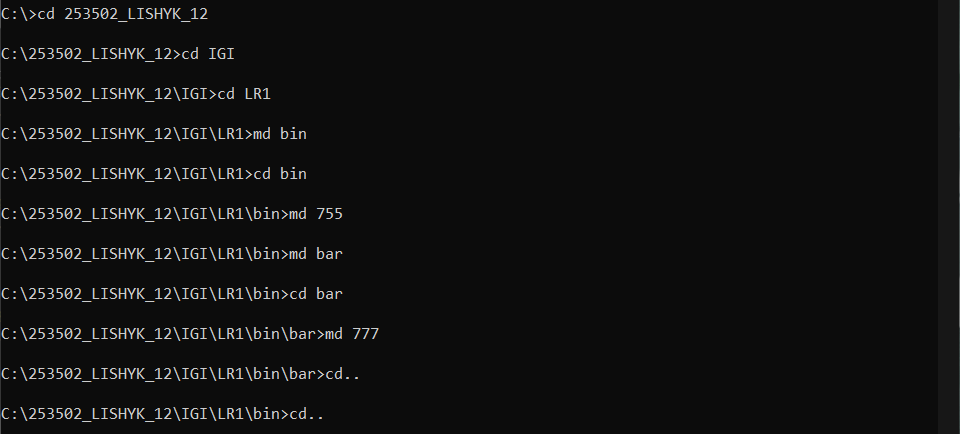


Рисунок 5 – Создание первого поддерева

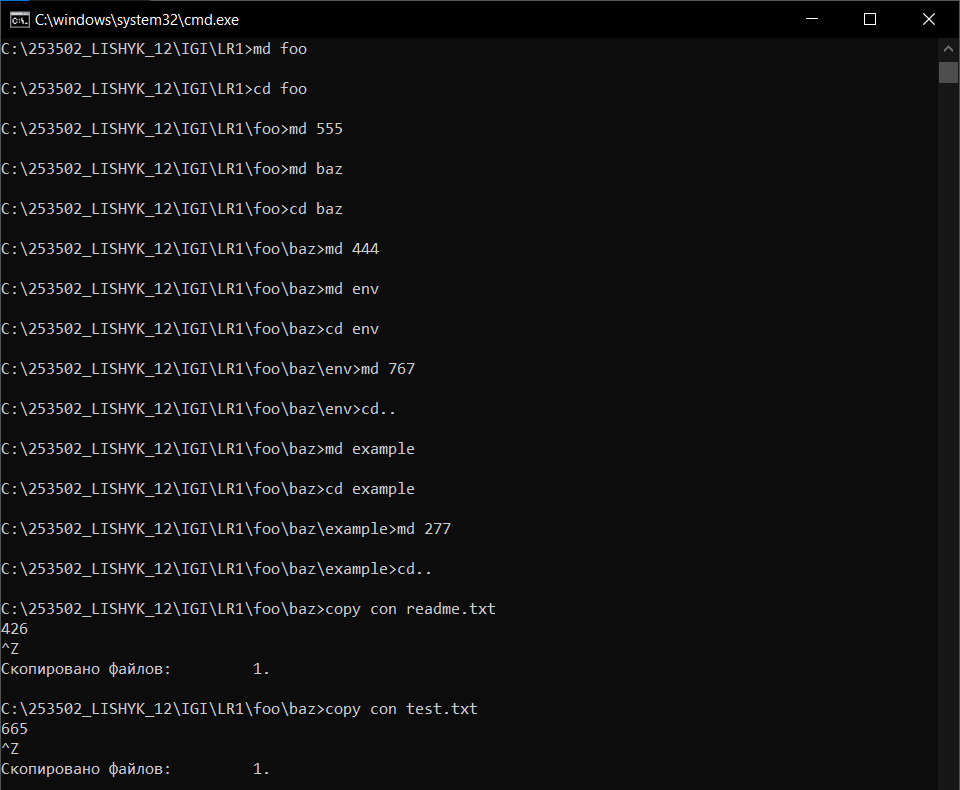
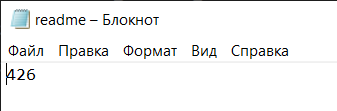


Рисунок 5 – Создание второго поддерева



Рисунок 6 – Cтруктура папок



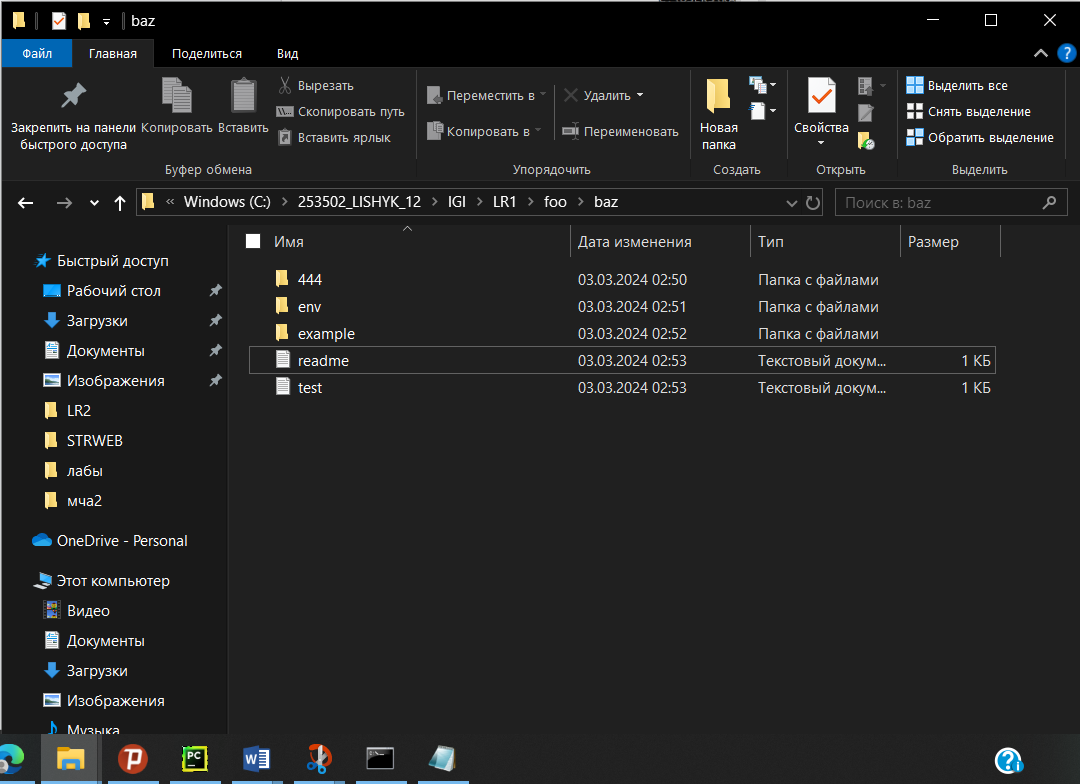




Рисунок 7 – Создание локального репозитория

**Задание 3.** Создайте репозиторий внутри папки LR1 с помощью клонирования проекта с сайта github.com.

<https://github.com/smartiqaorg/geometric_lib>

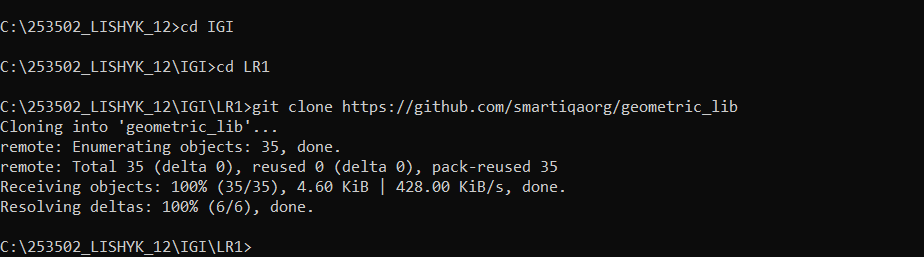


Рисунок 1 – Клонирование удаленного репозитория в папку LR1

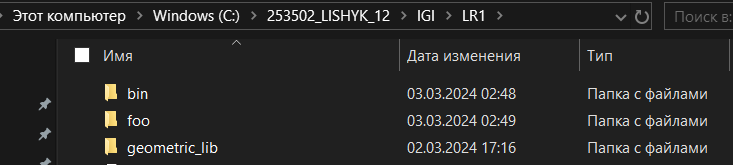
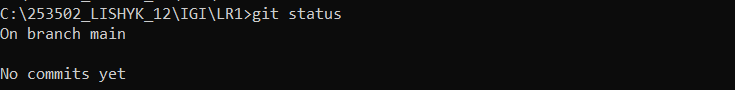


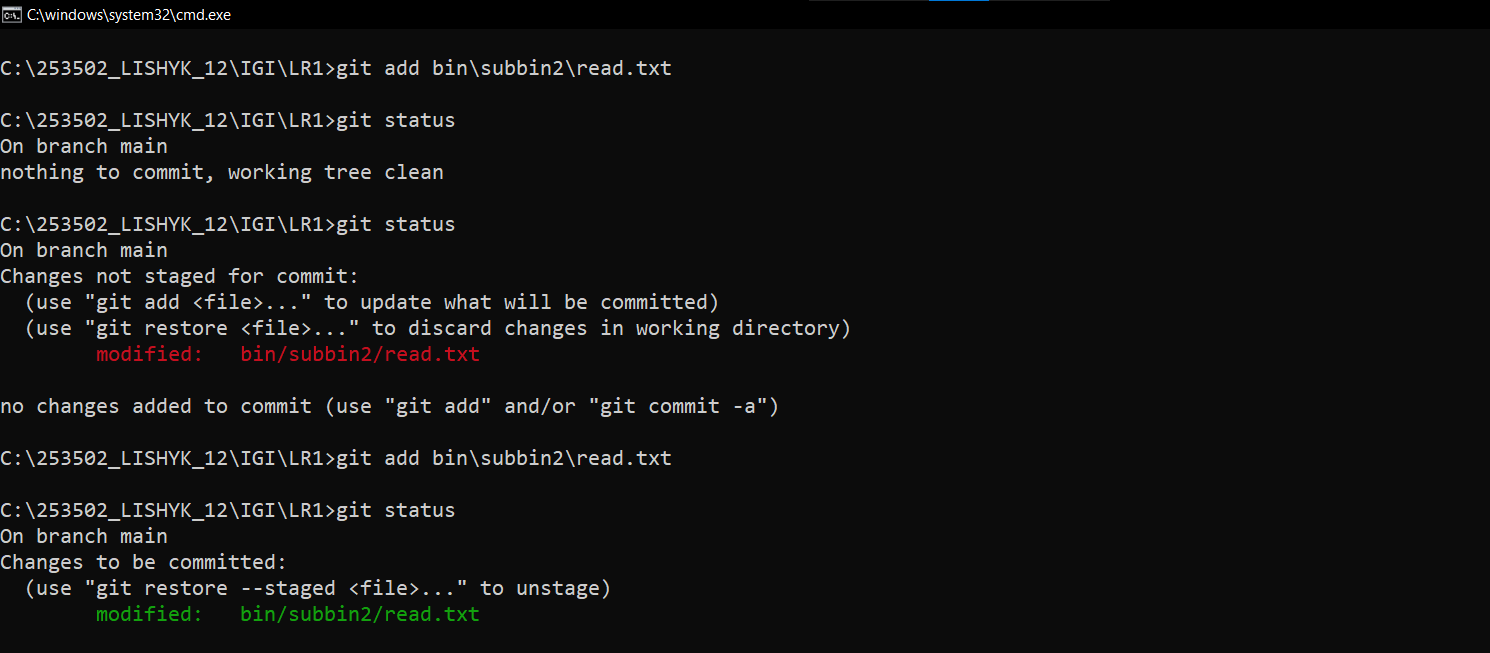
Рисунок 2 – Локальный репозиторий geometric\_lib

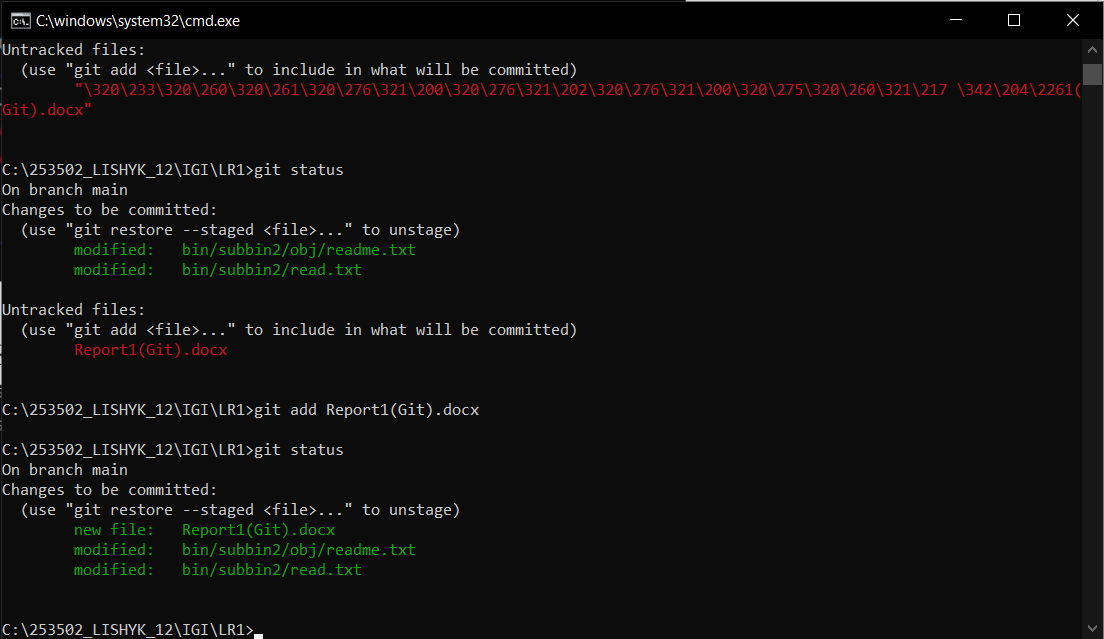


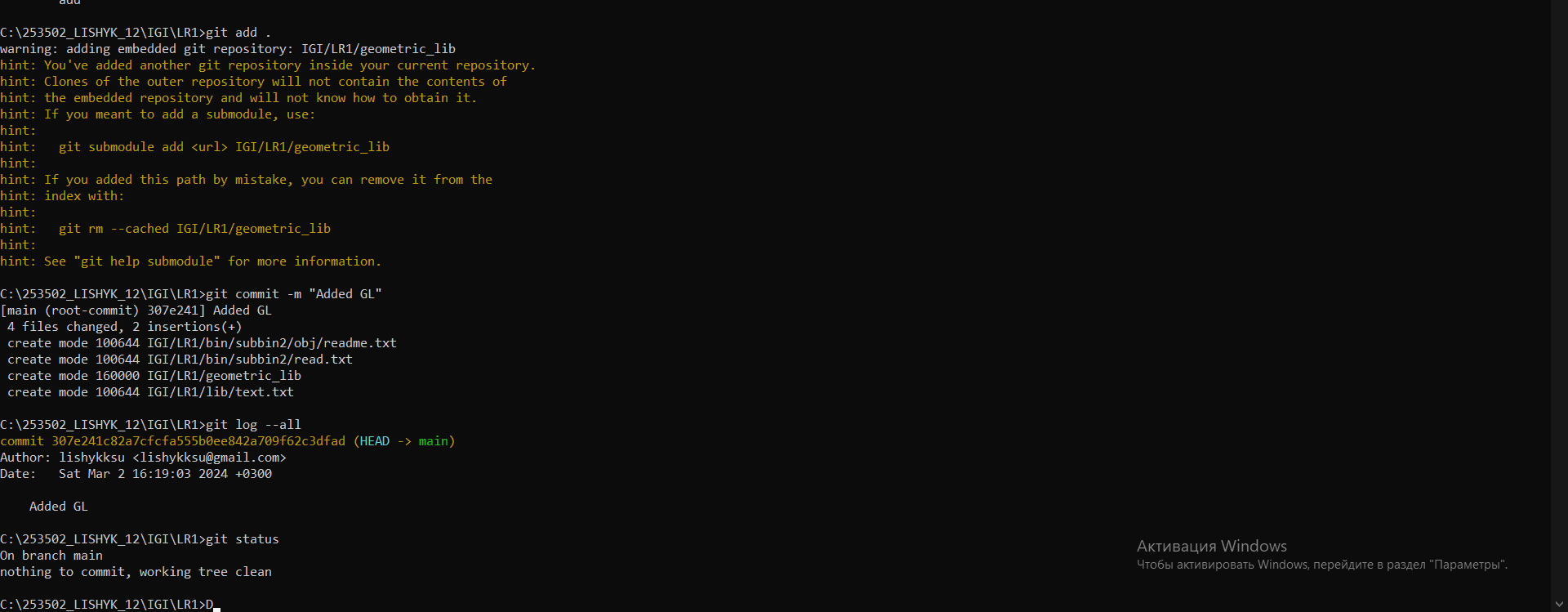
**Рисунок 3 – Состояние репозитория после git clone**

**Задание 4.** Выполните следующие действия, каждый раз проверяя состояние файлов в локальном репозитории до и после выполнения операции:

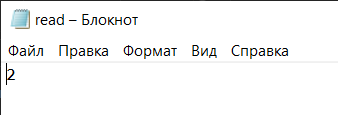
1. Сделайте индексацию нескольких файлов в вашем Git-репозитории (git add …)

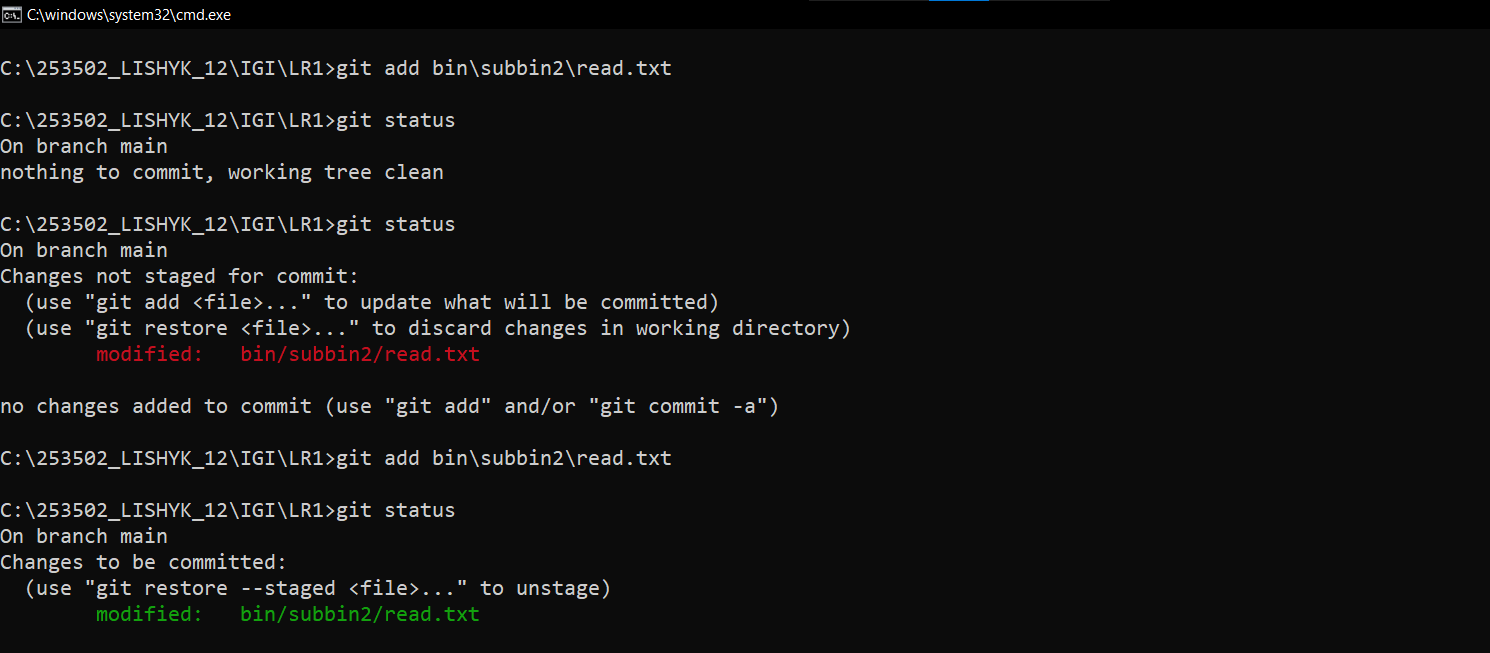
****

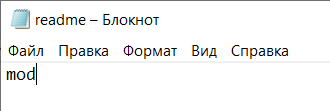


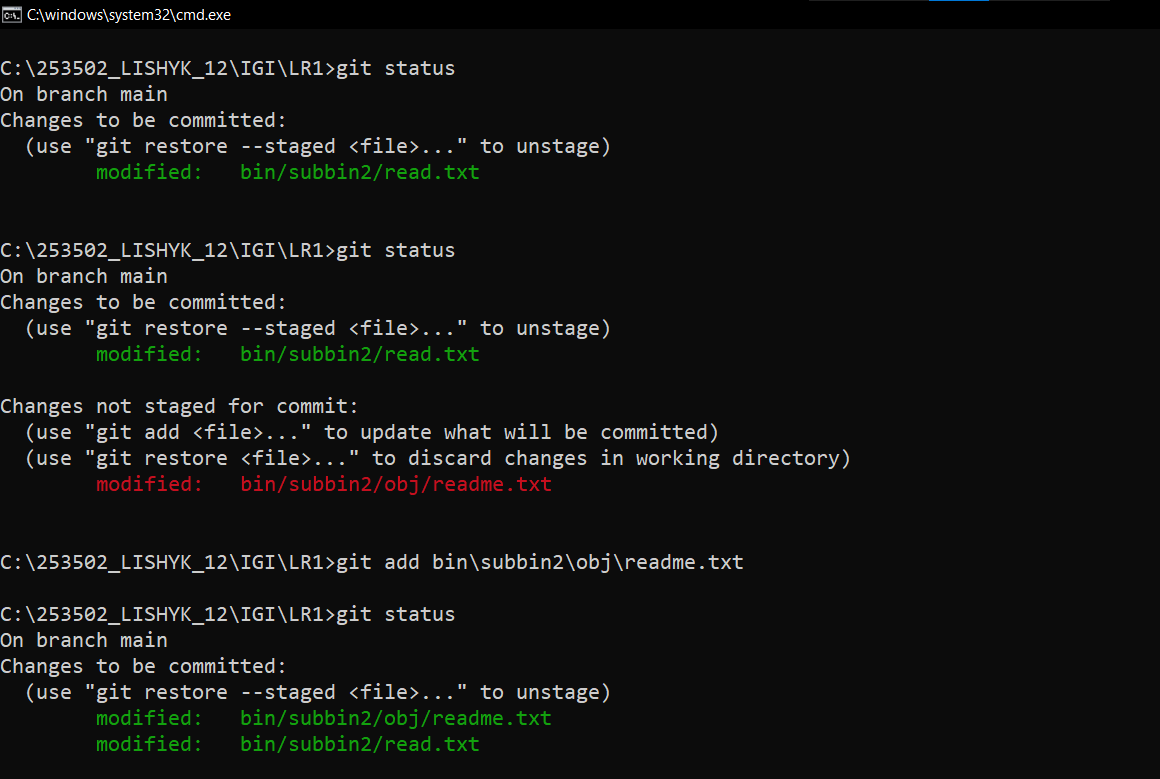


1. Внесите изменения в файлы

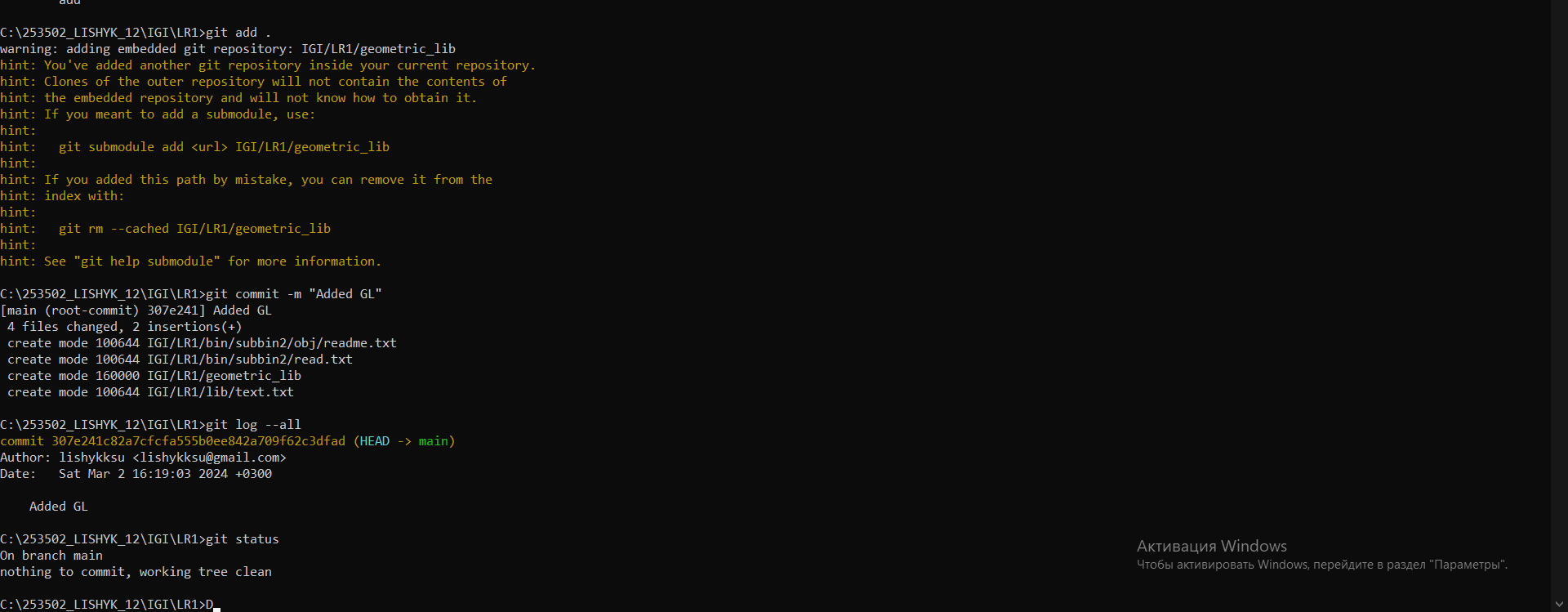


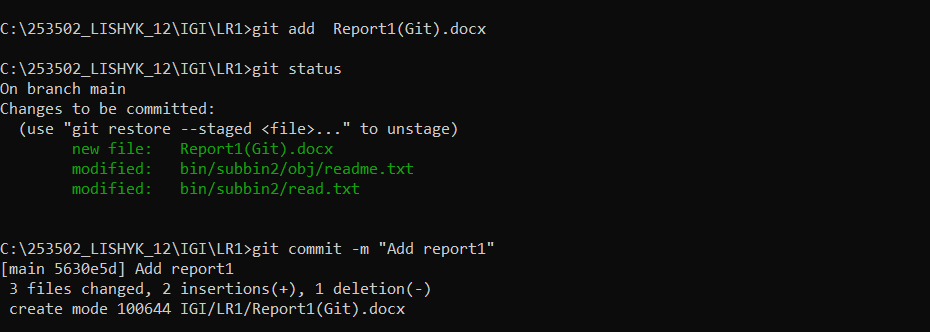
****

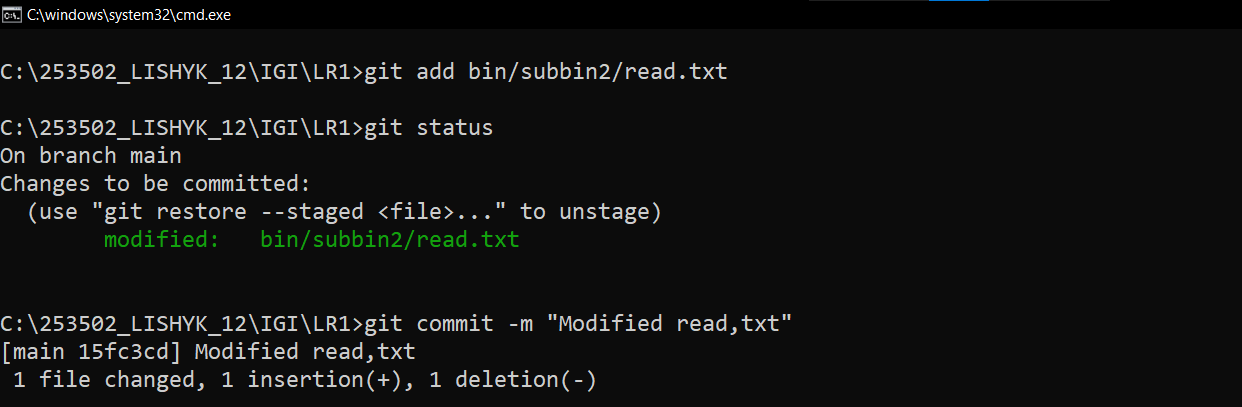
****

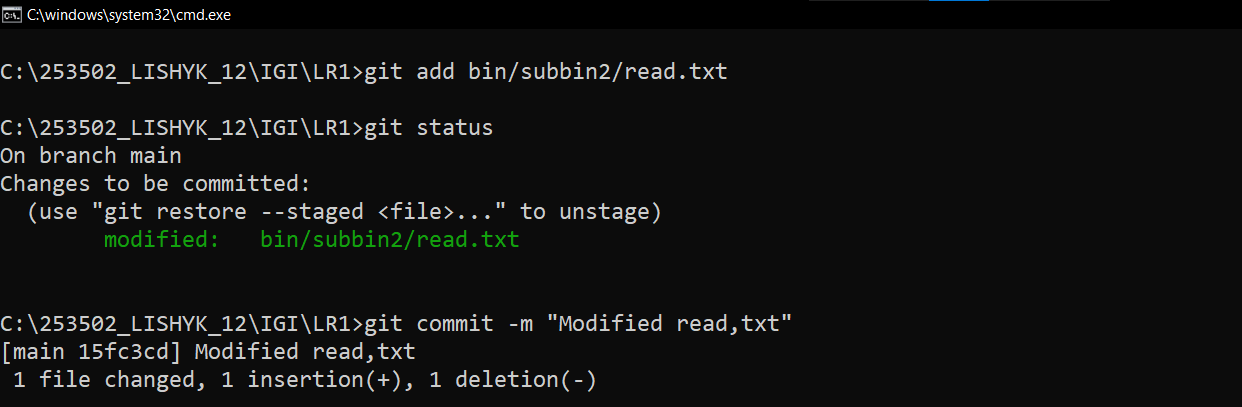
****

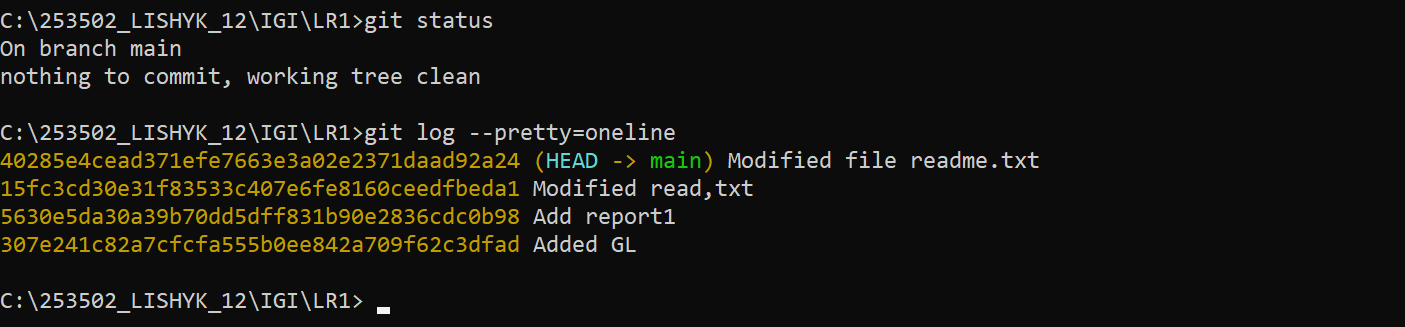
1. Выполните коммиты к проиндексированным  файлам



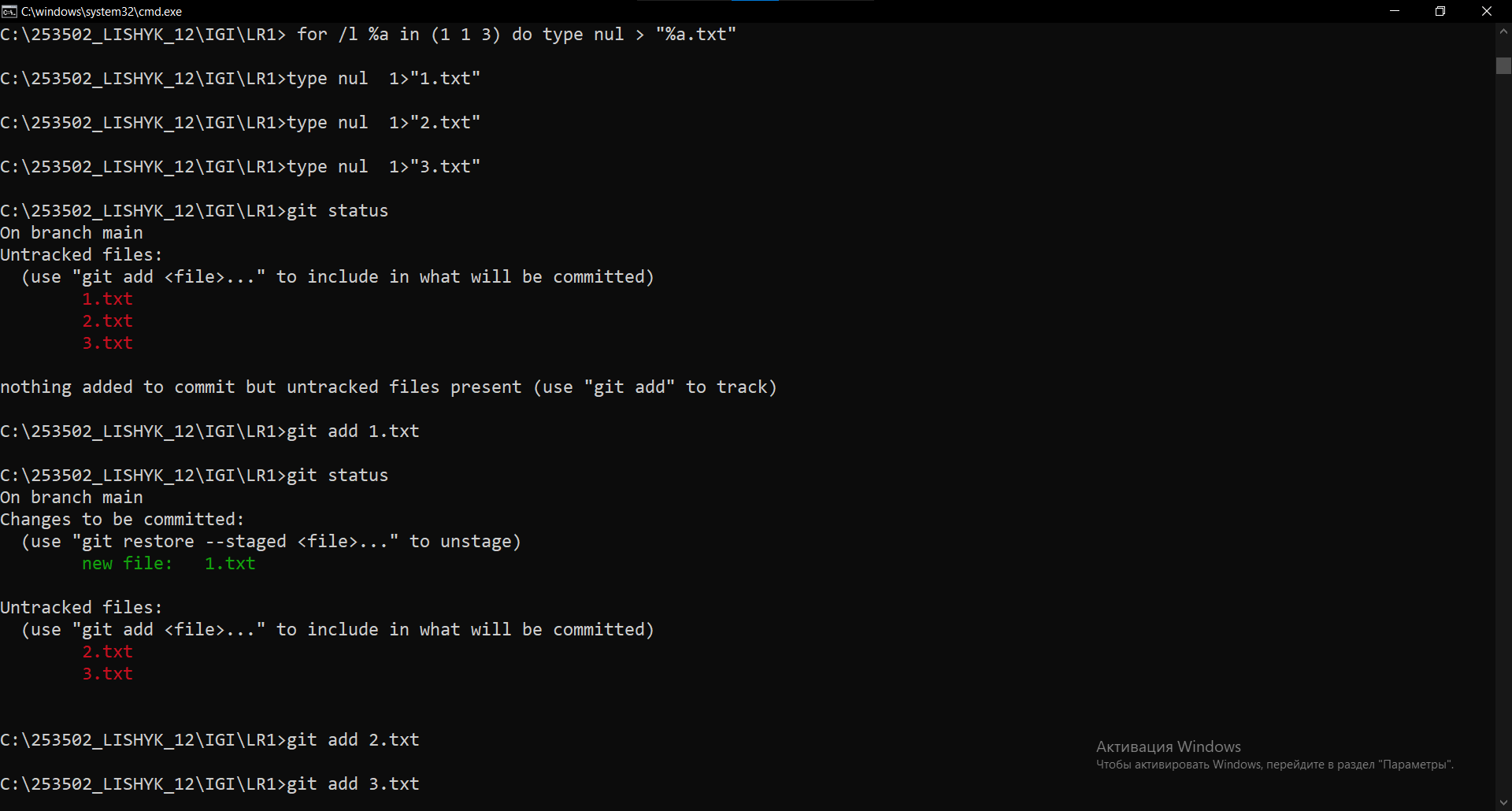


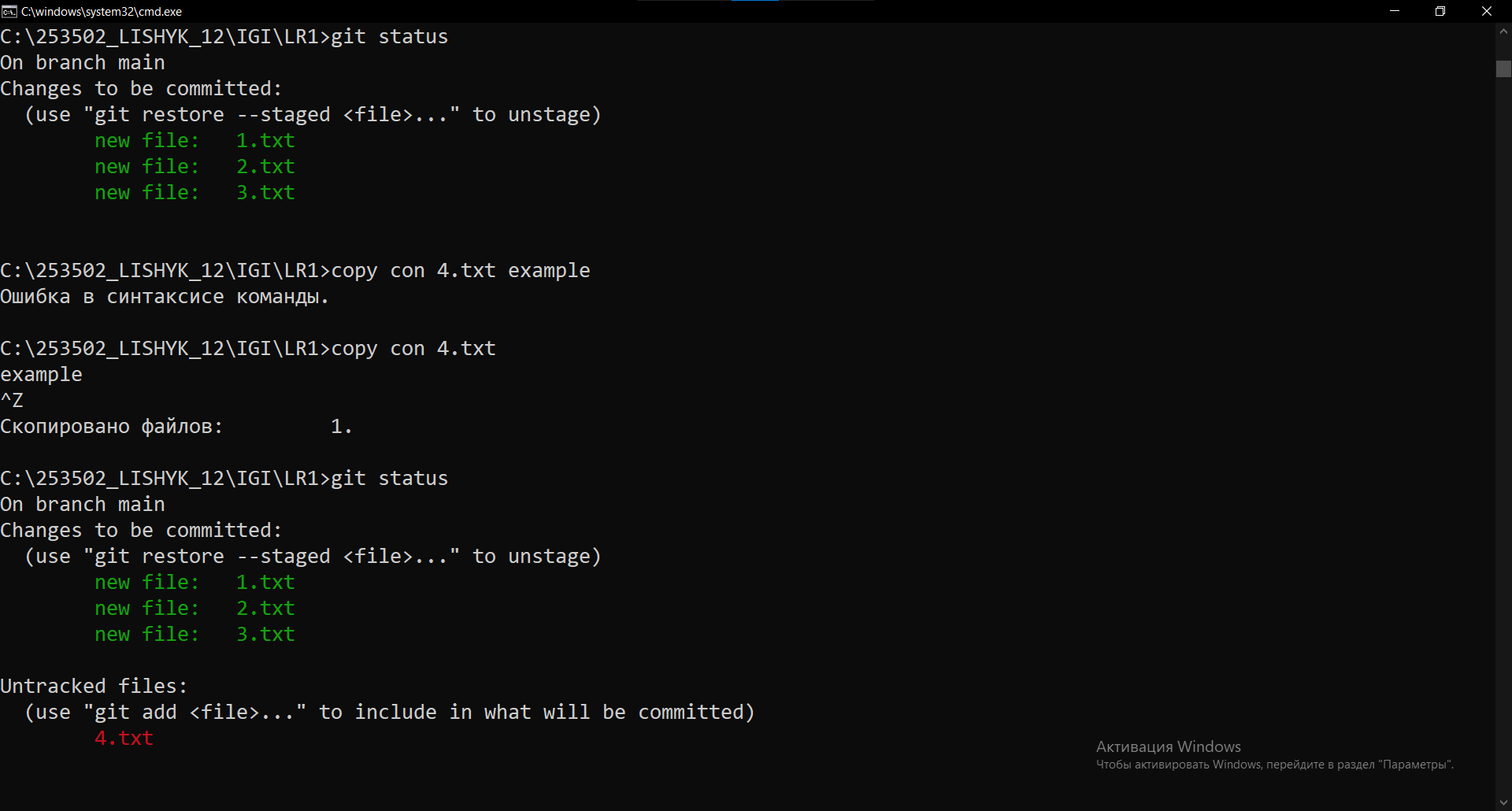


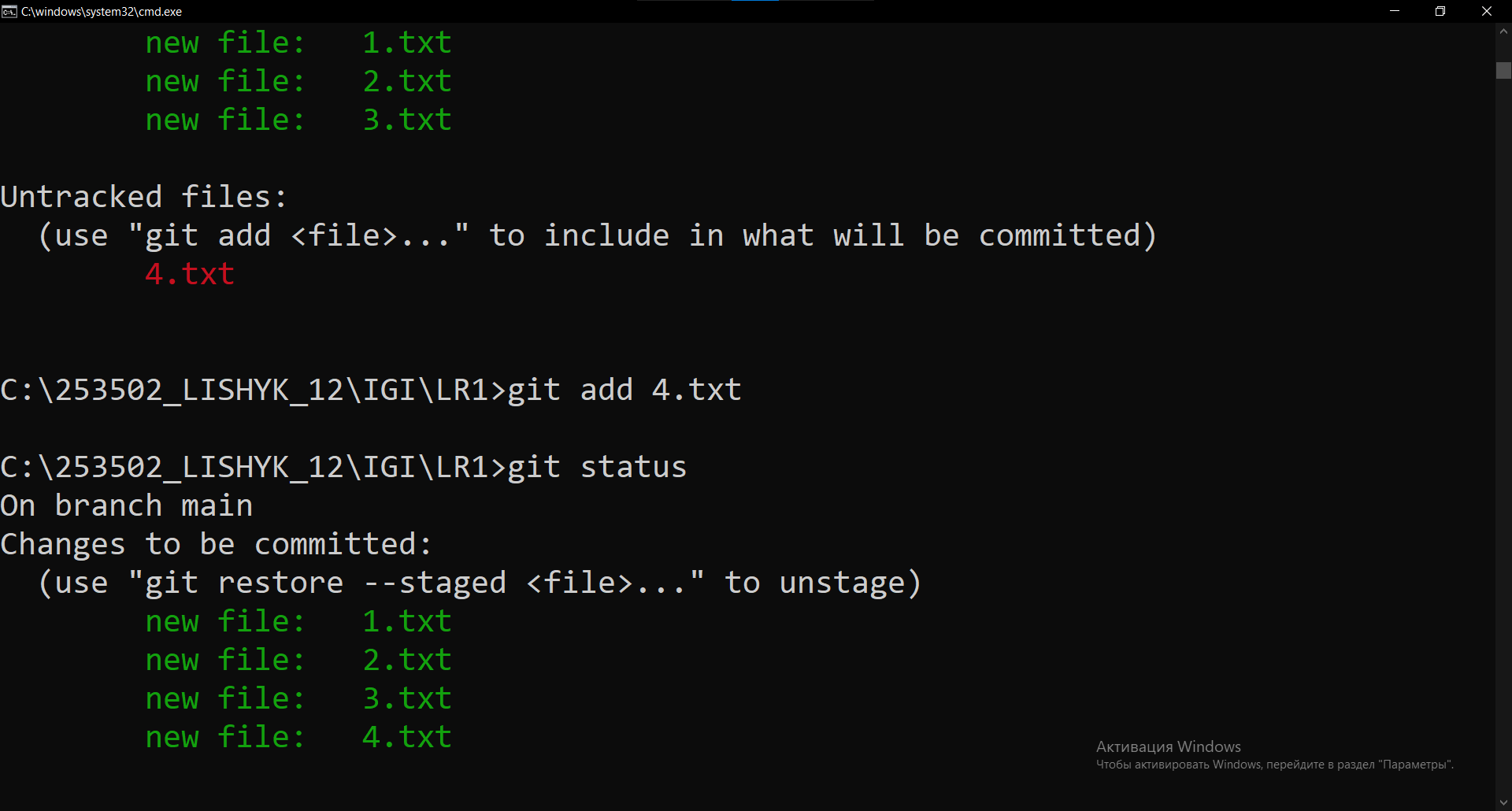


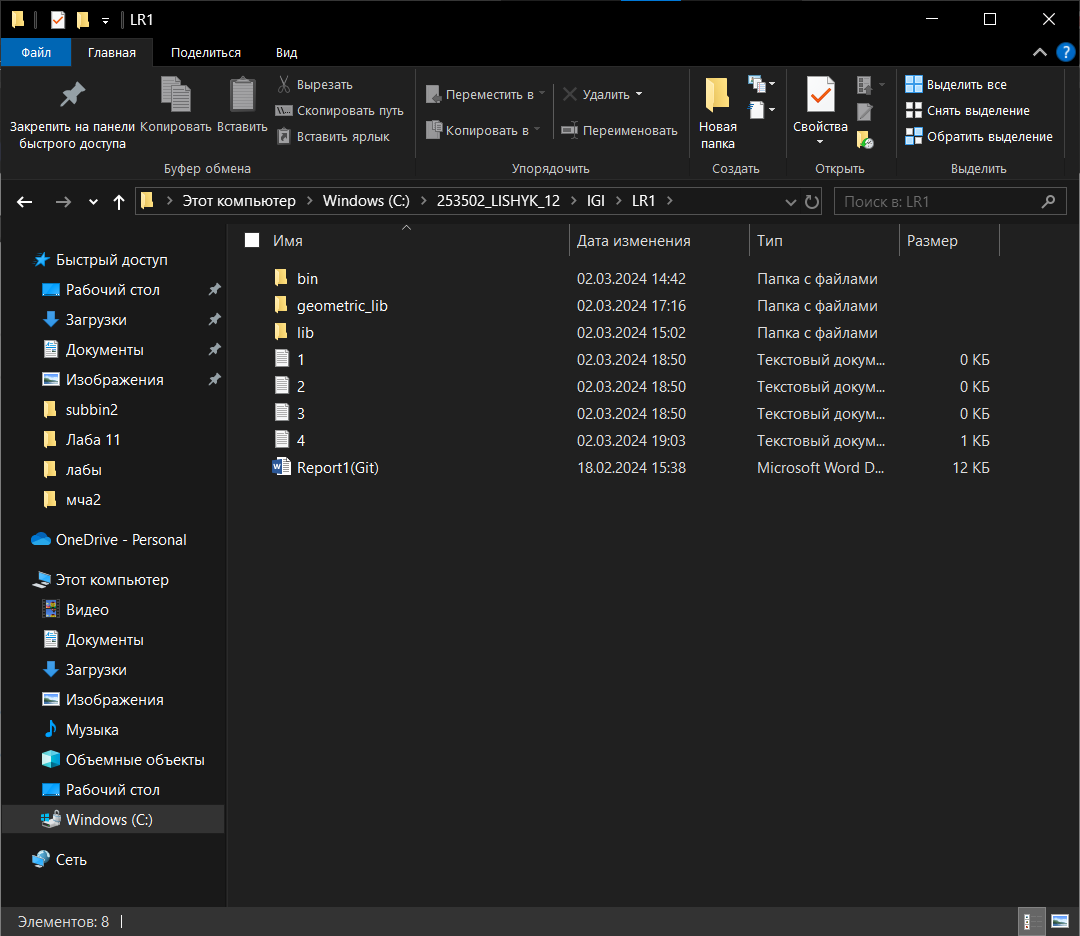


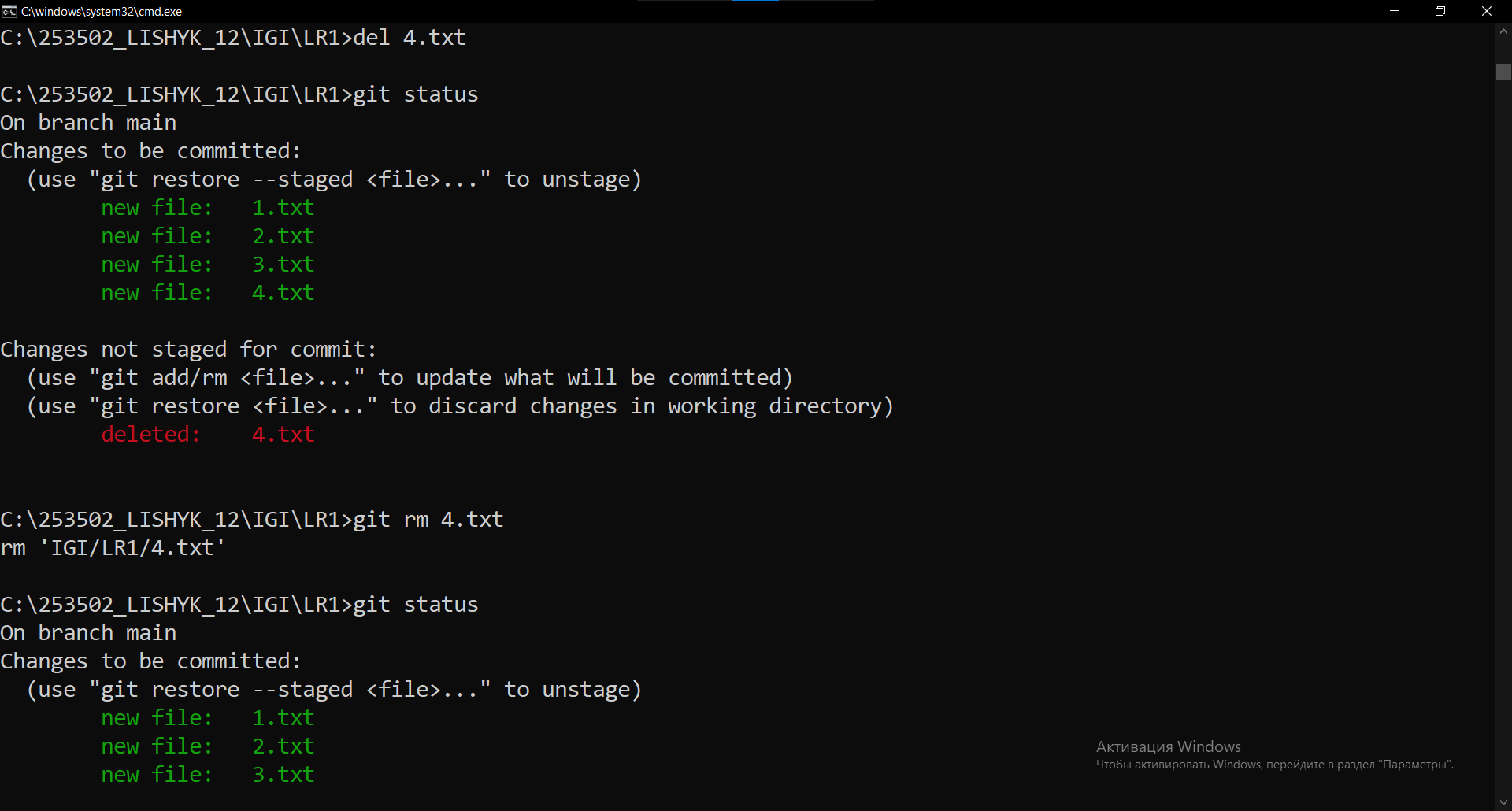
1. Добавьте и удалите файлы (один и группу файлов с определенным расширением)

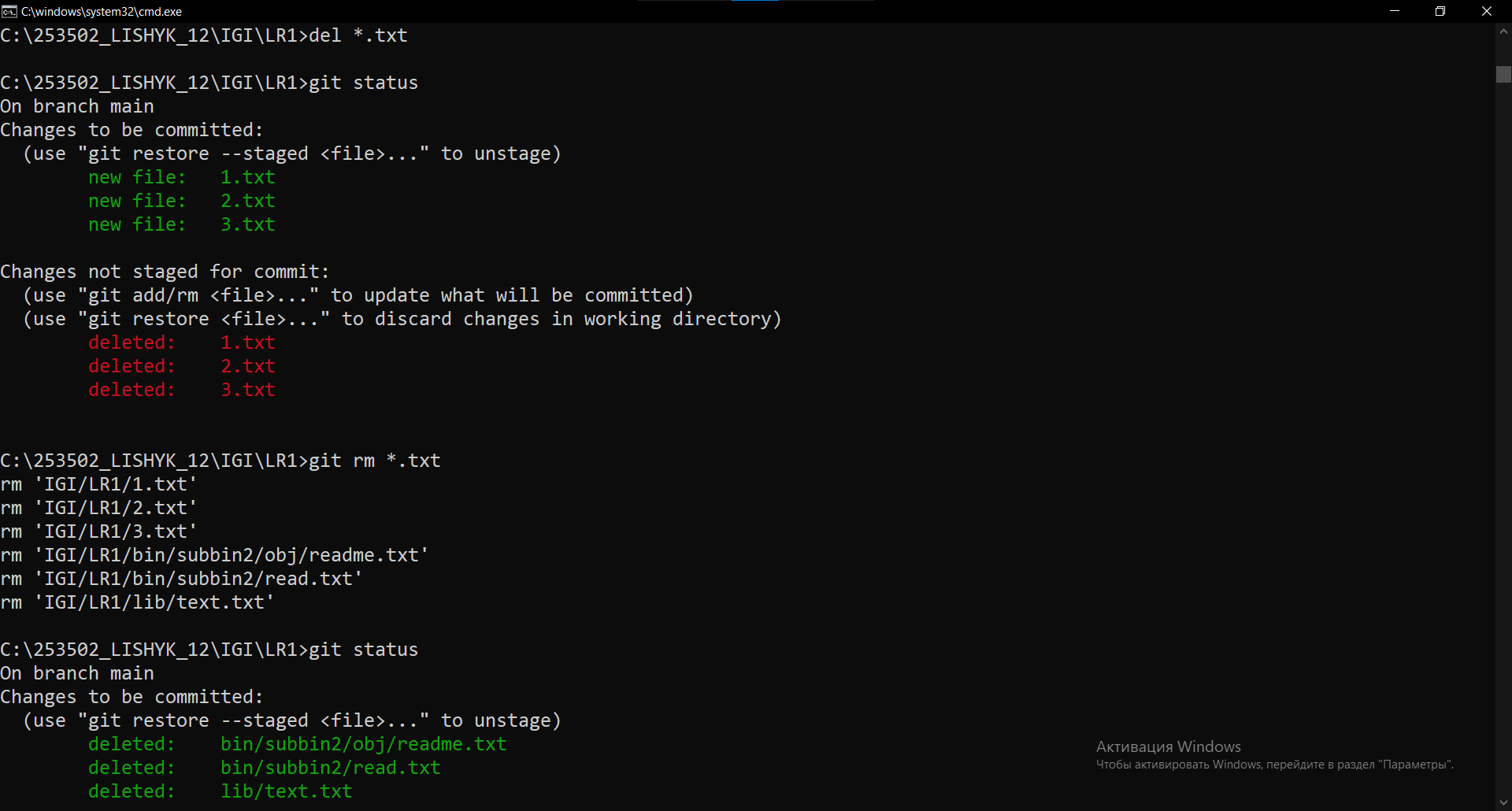


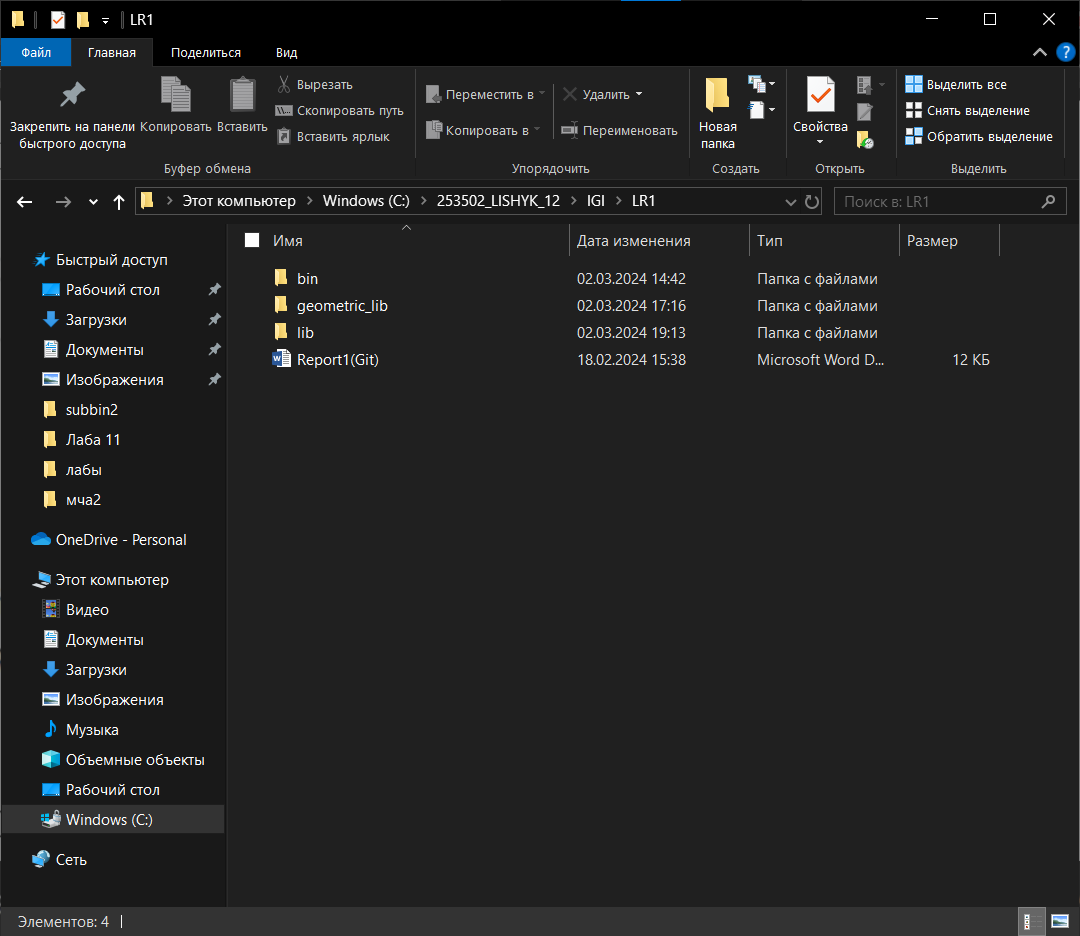




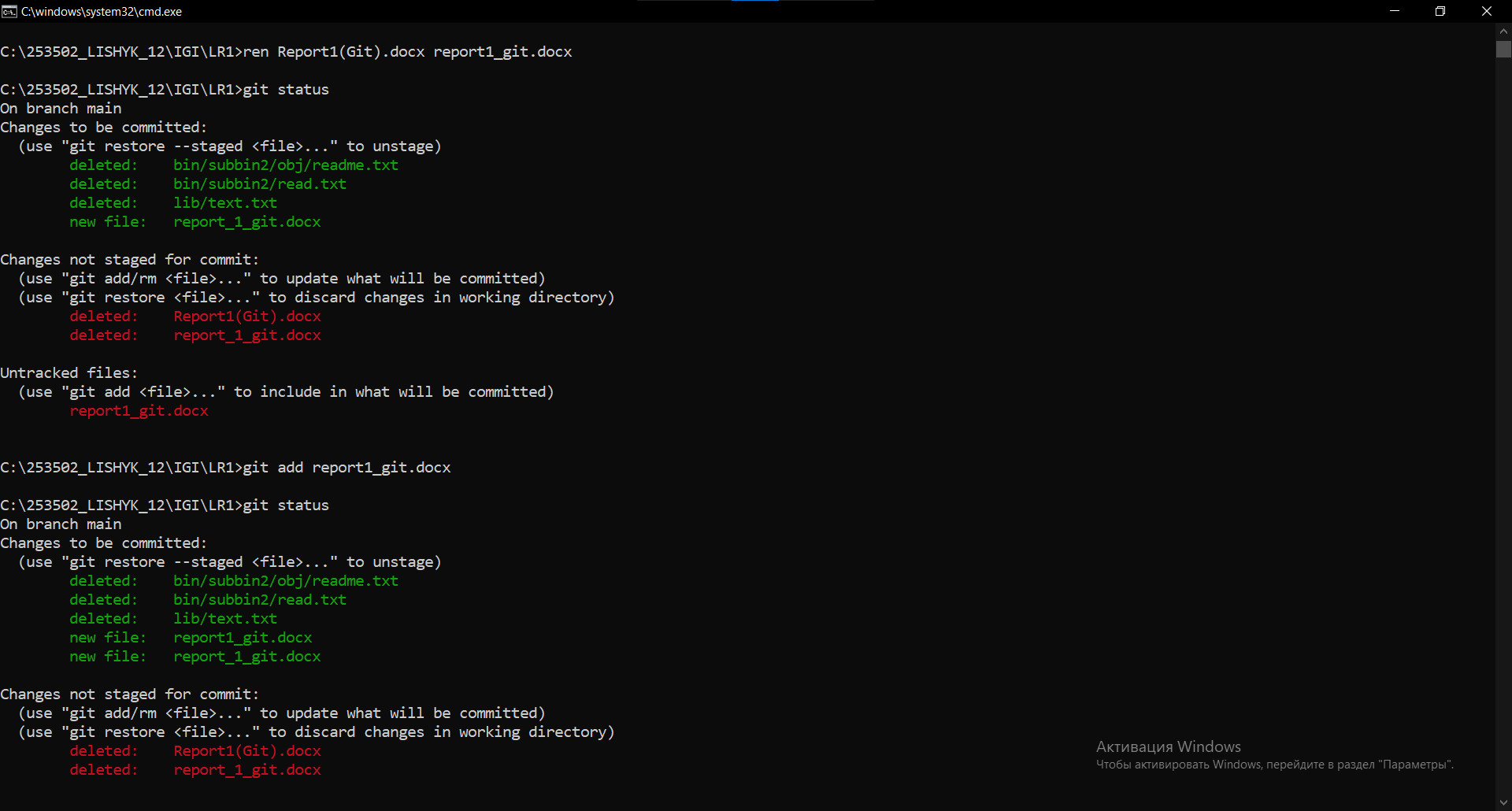


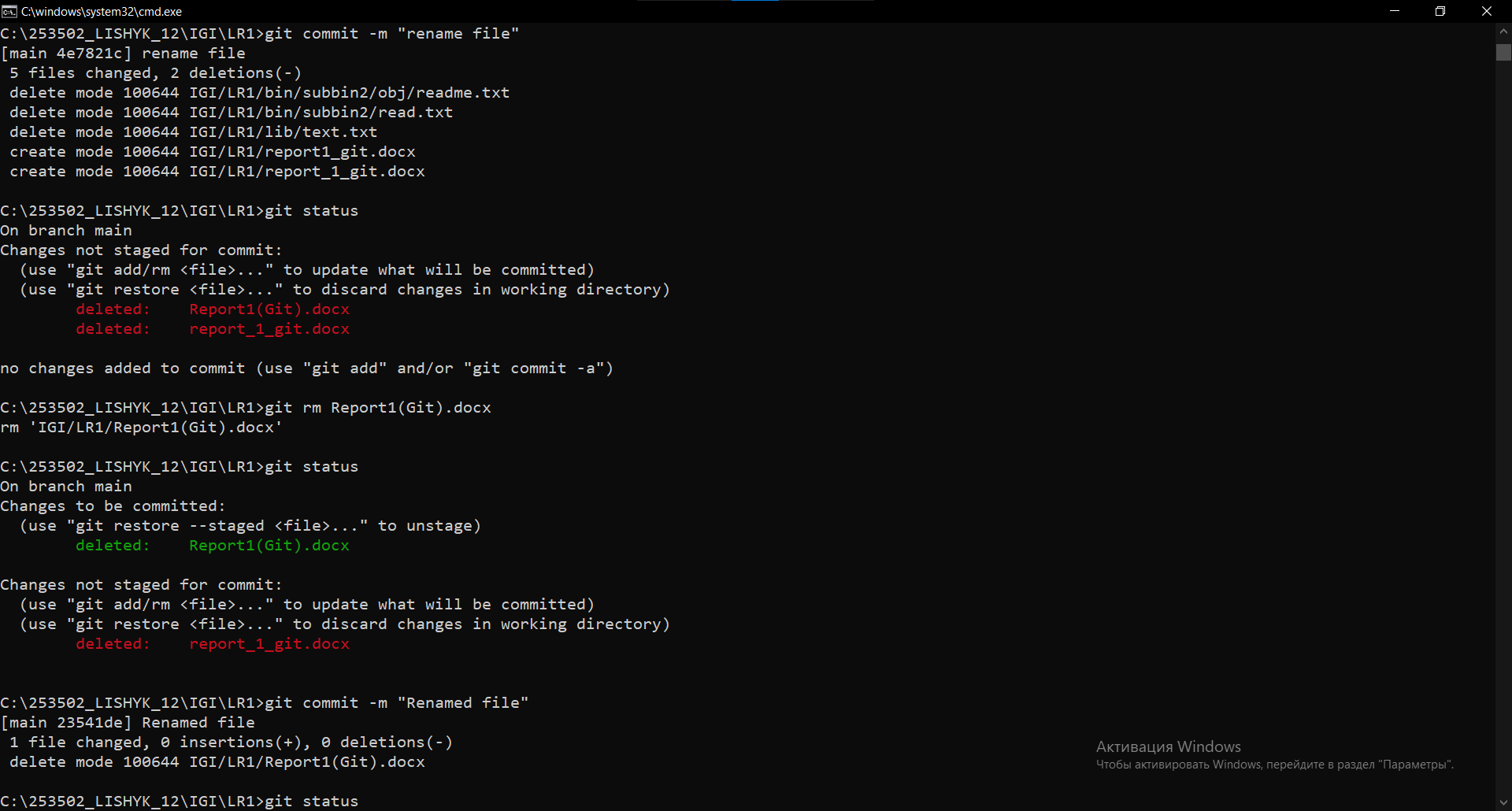


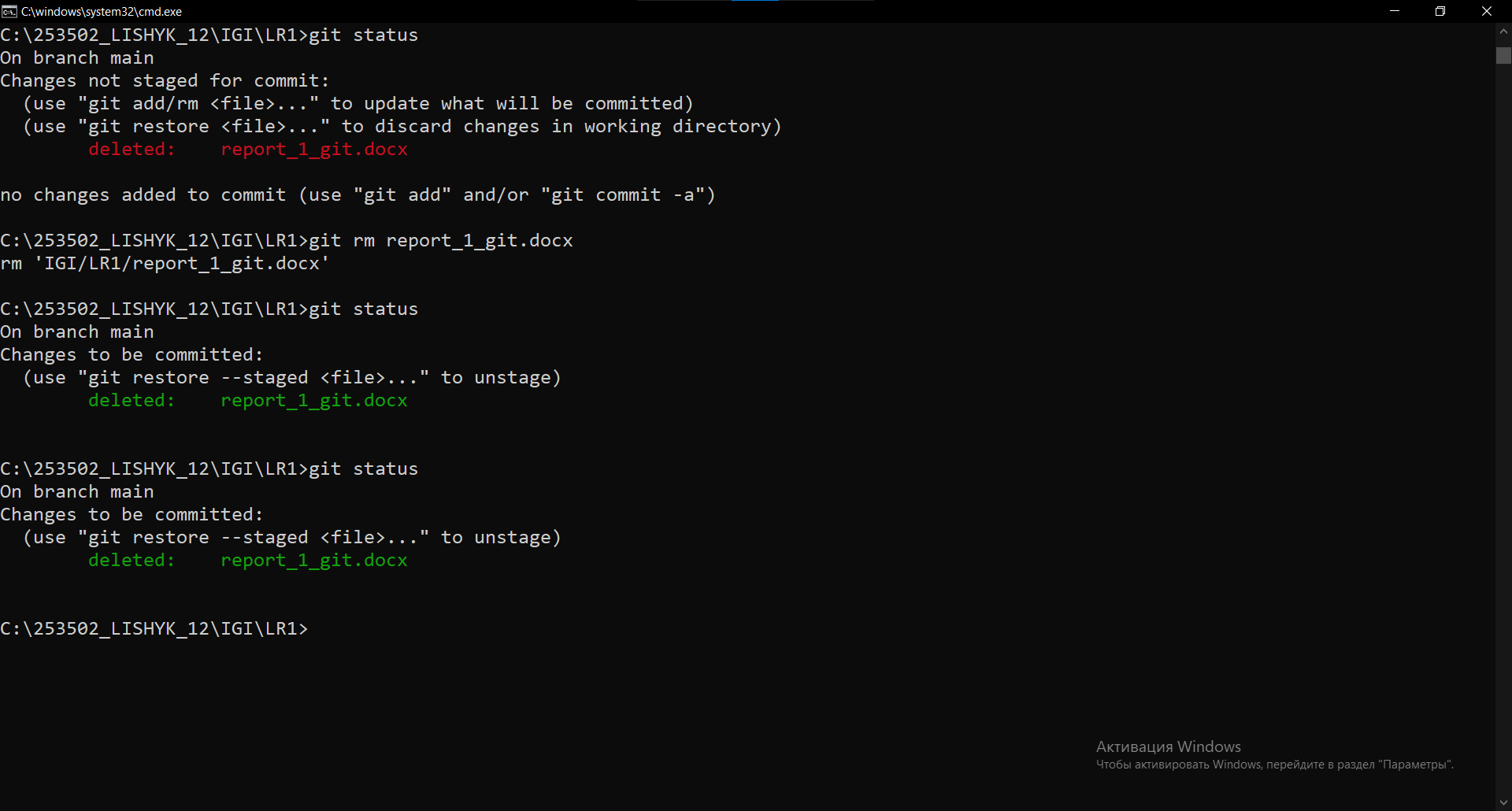




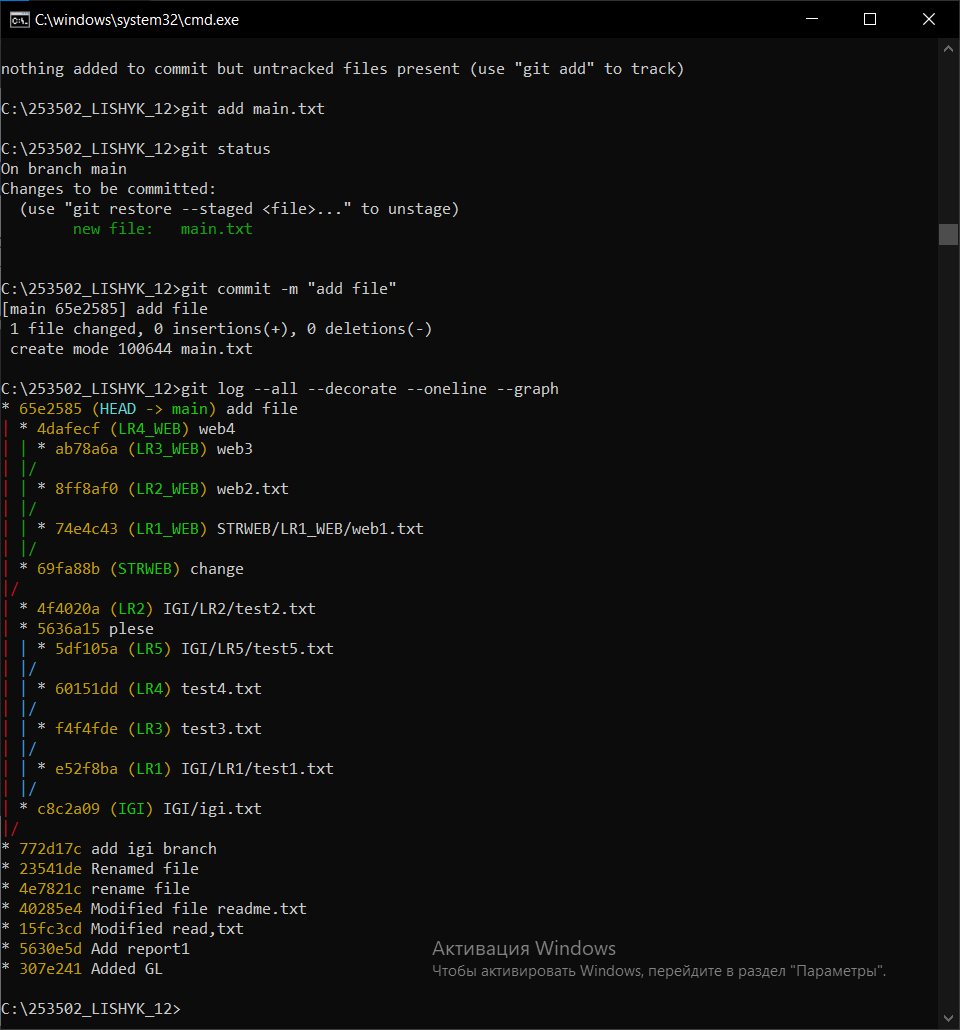
1. Переименуйте файлы

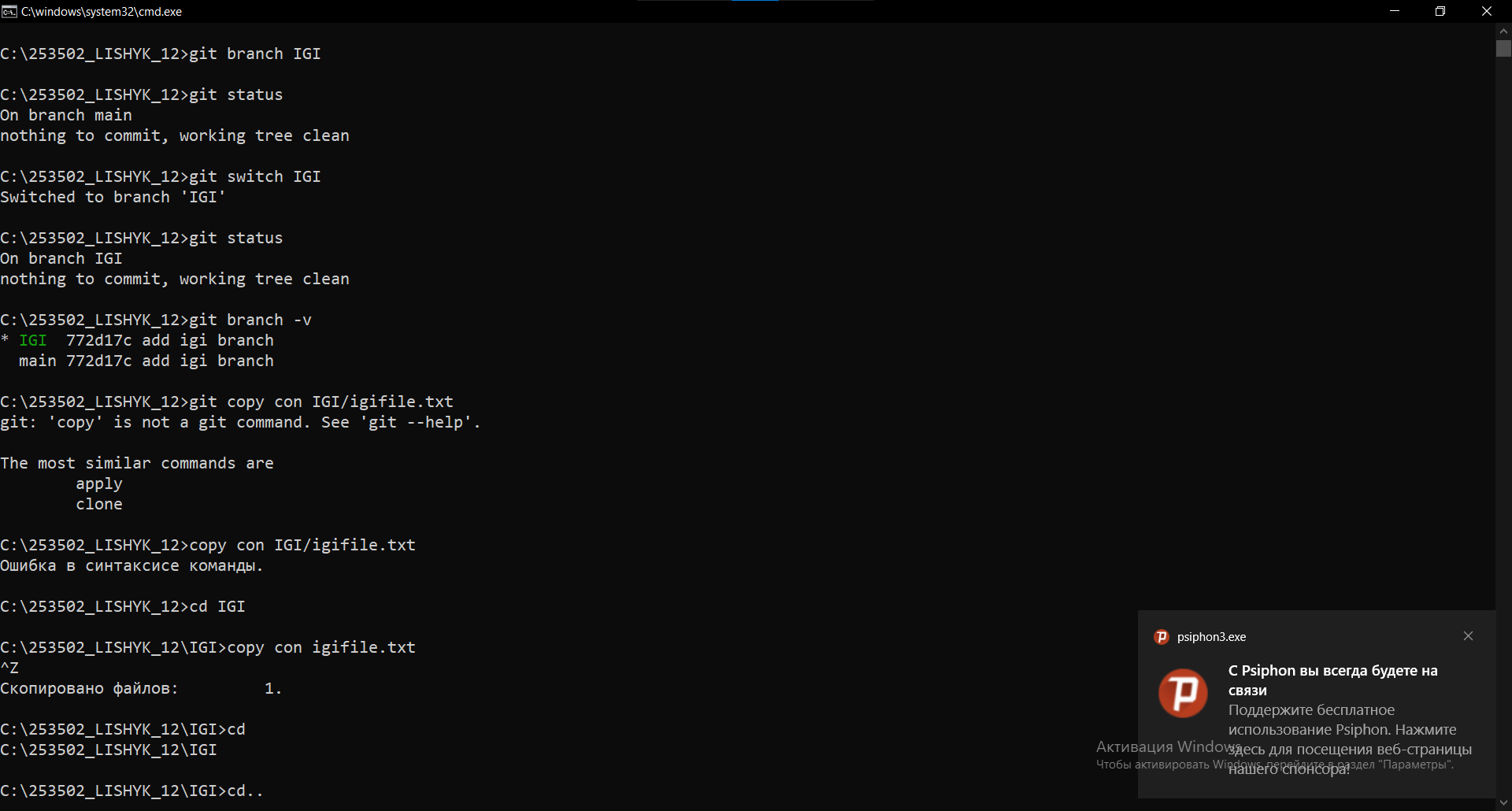


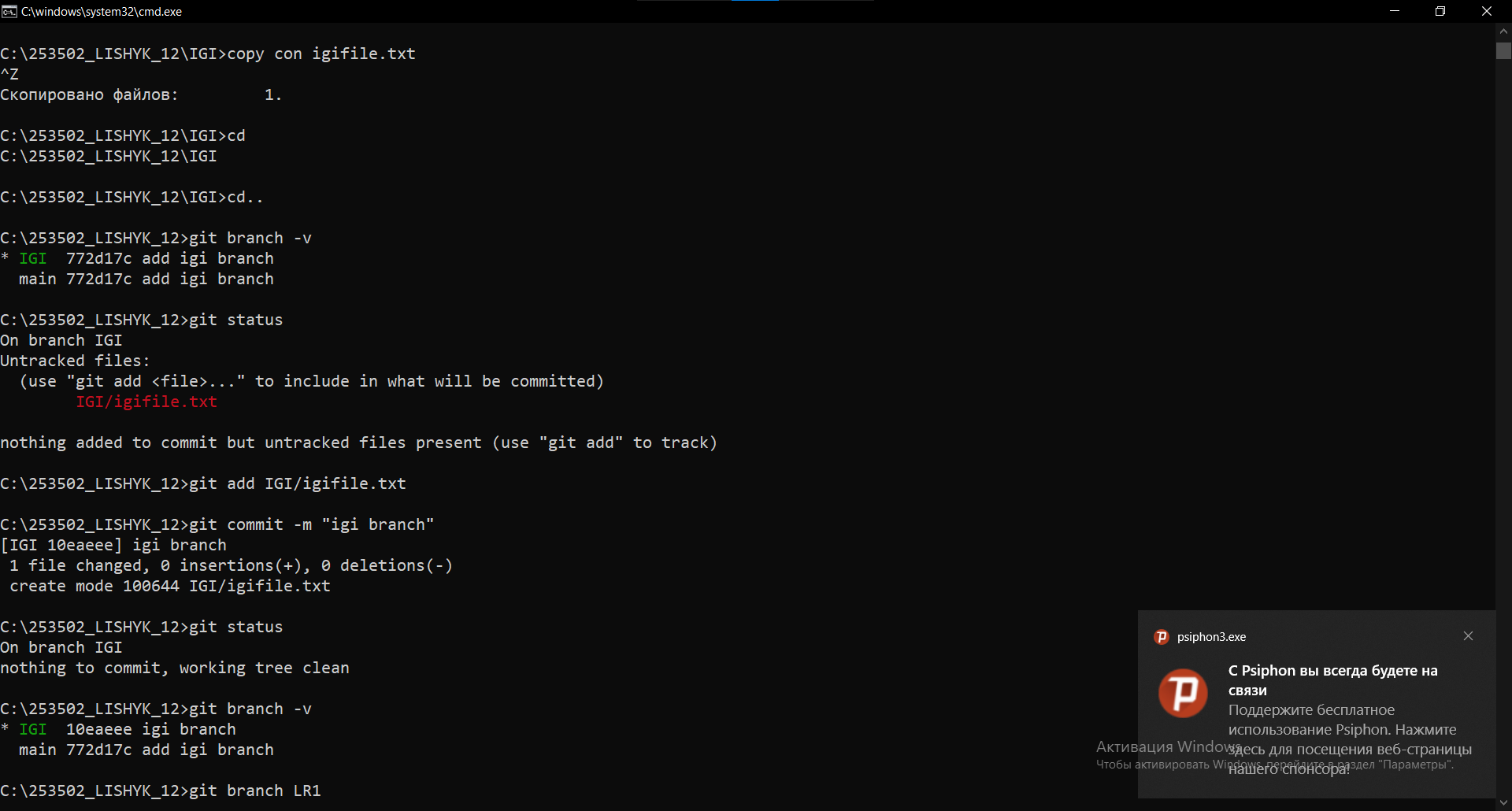


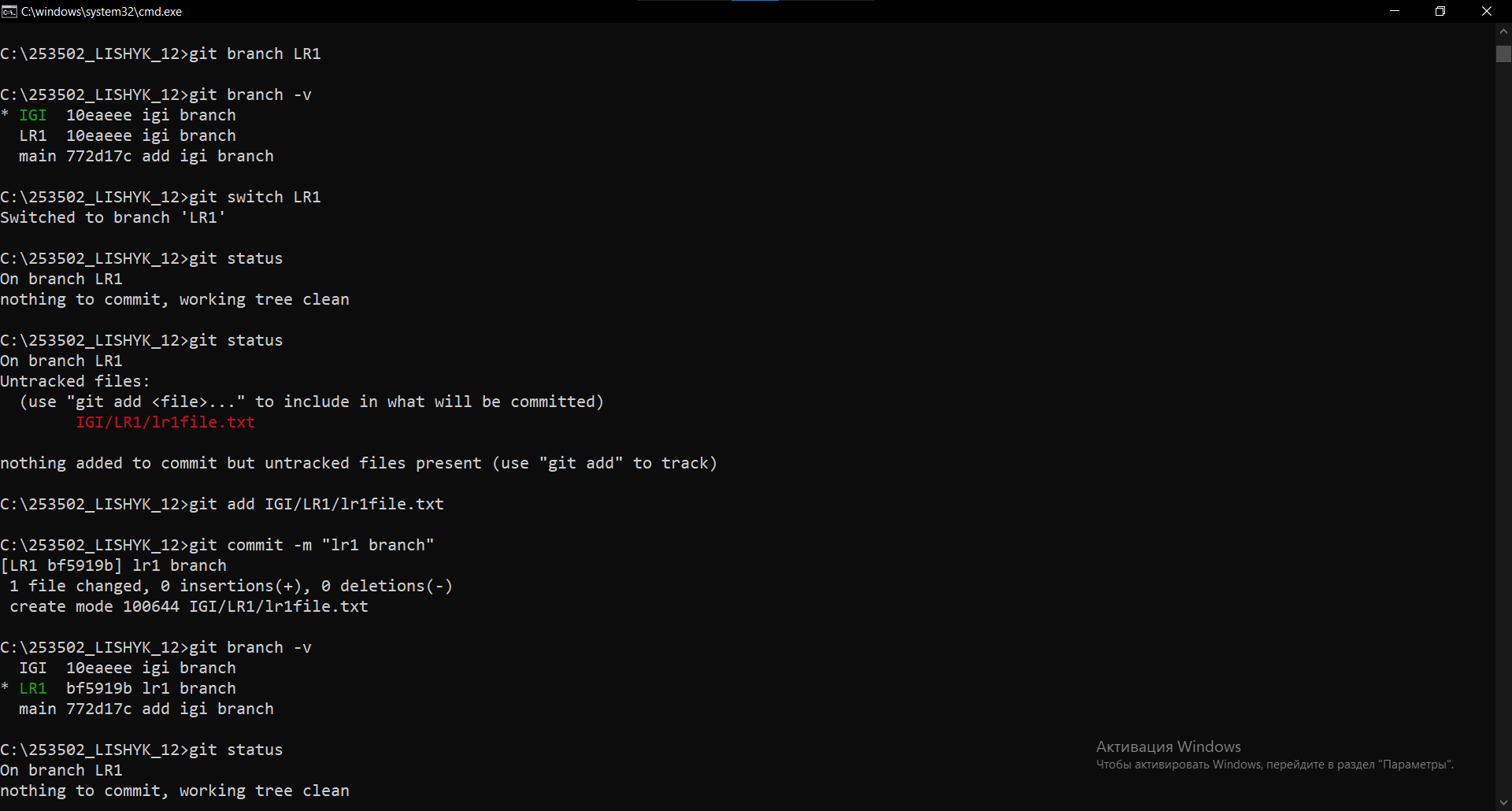


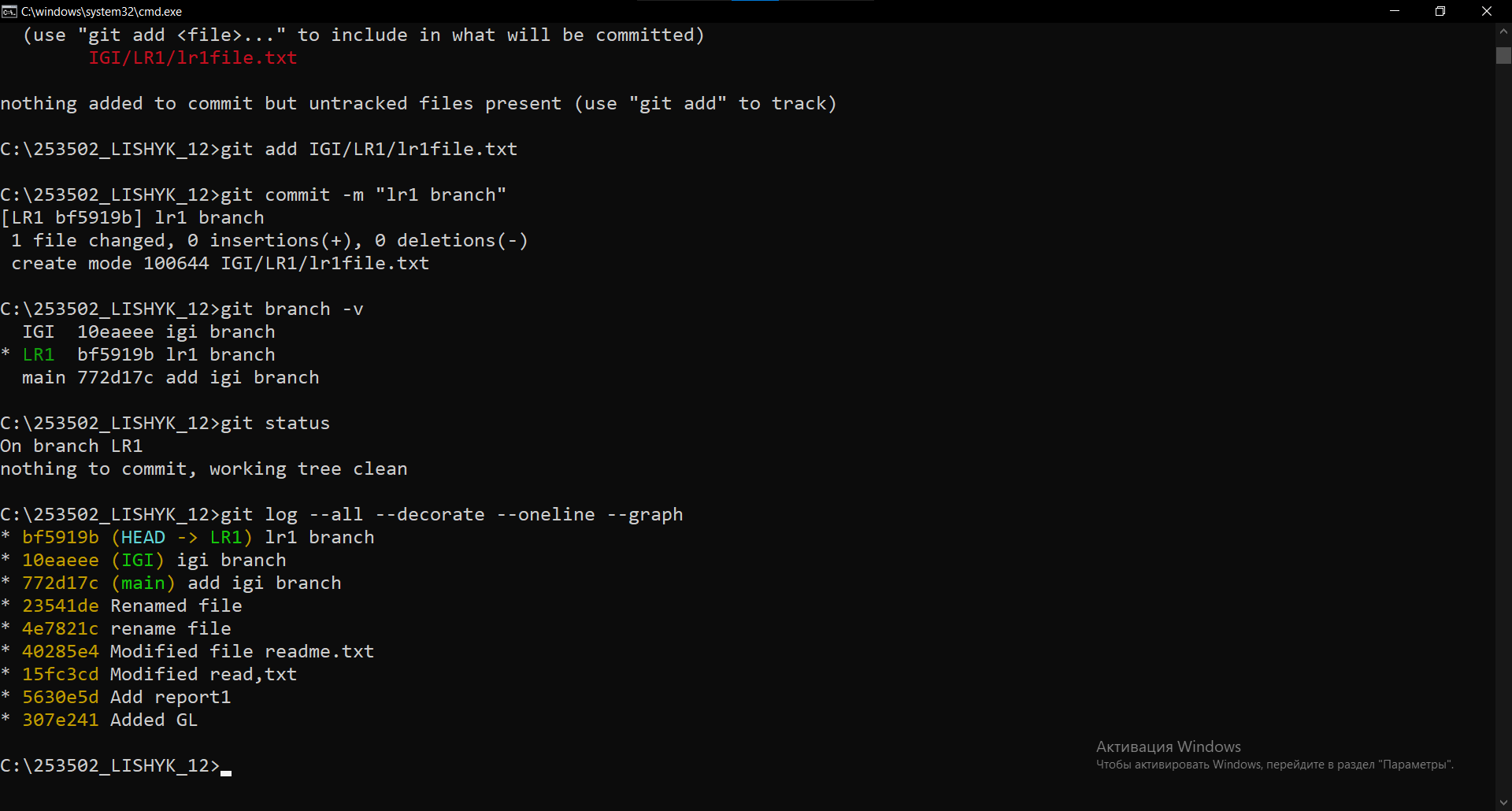
1. Создайте для IGI и STRWEB, а также для каждой ЛР, вложенной в них, отдельные ветки, переместите папки и файлы на соответствующие ветки.



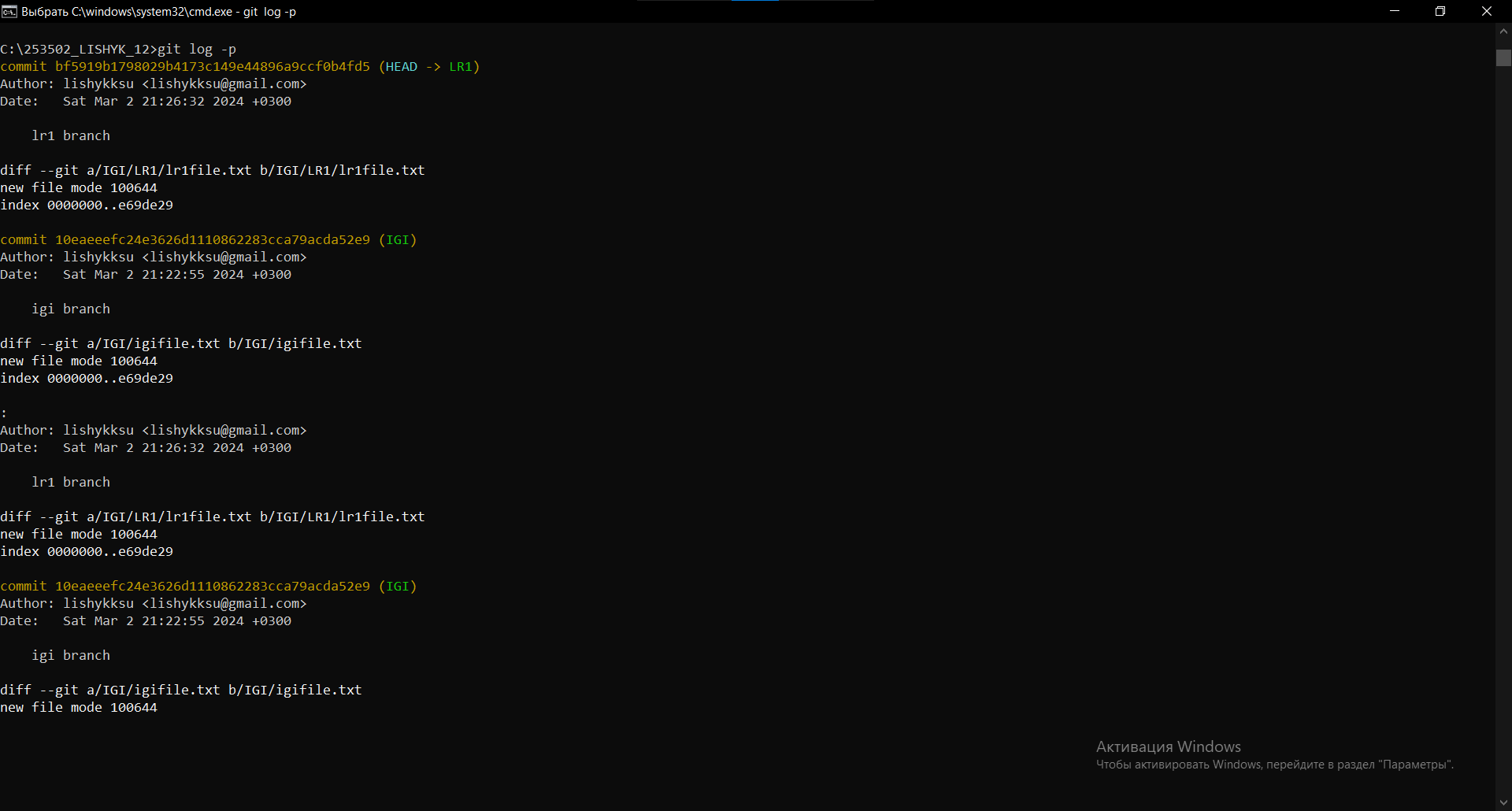


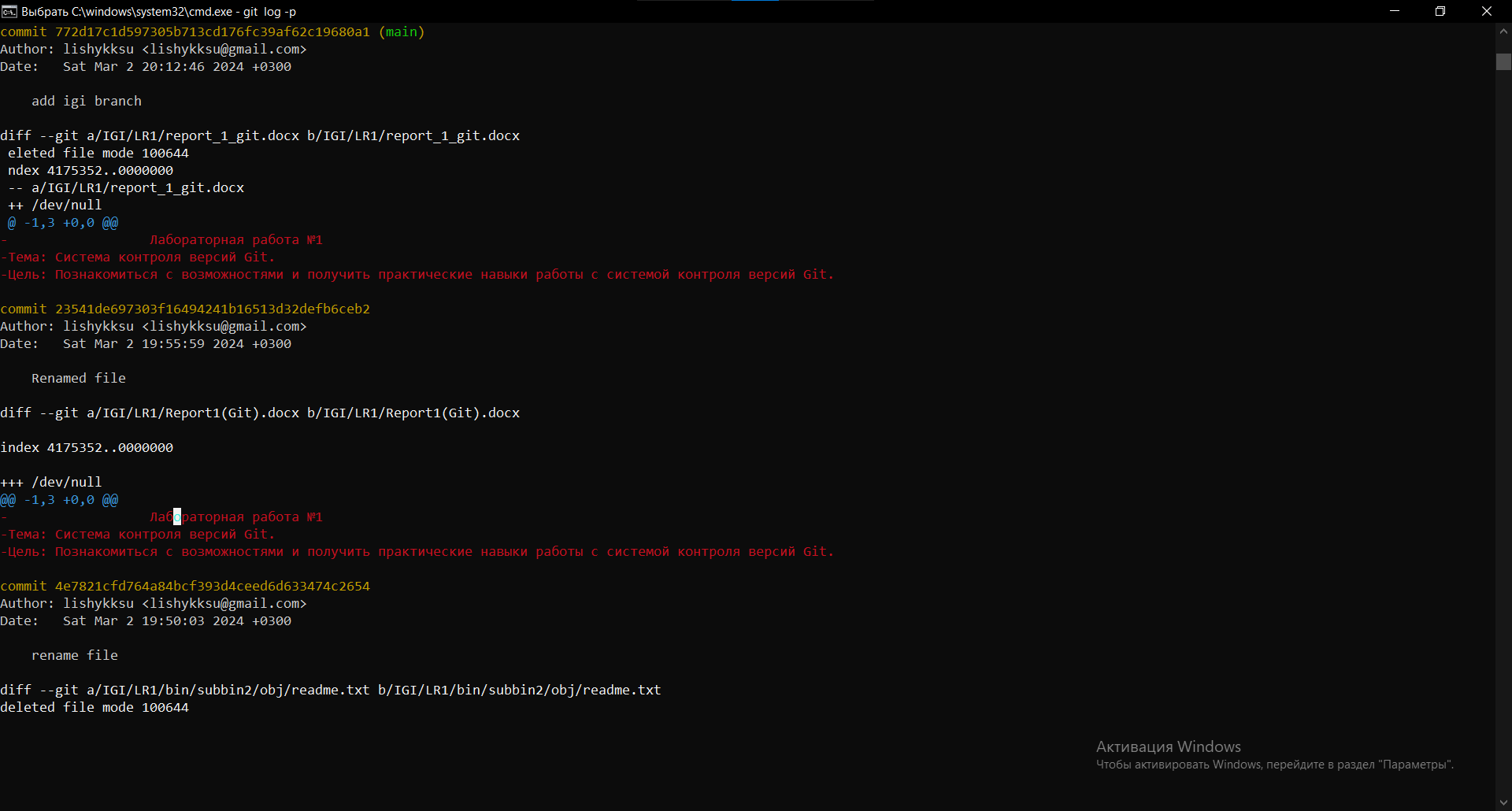


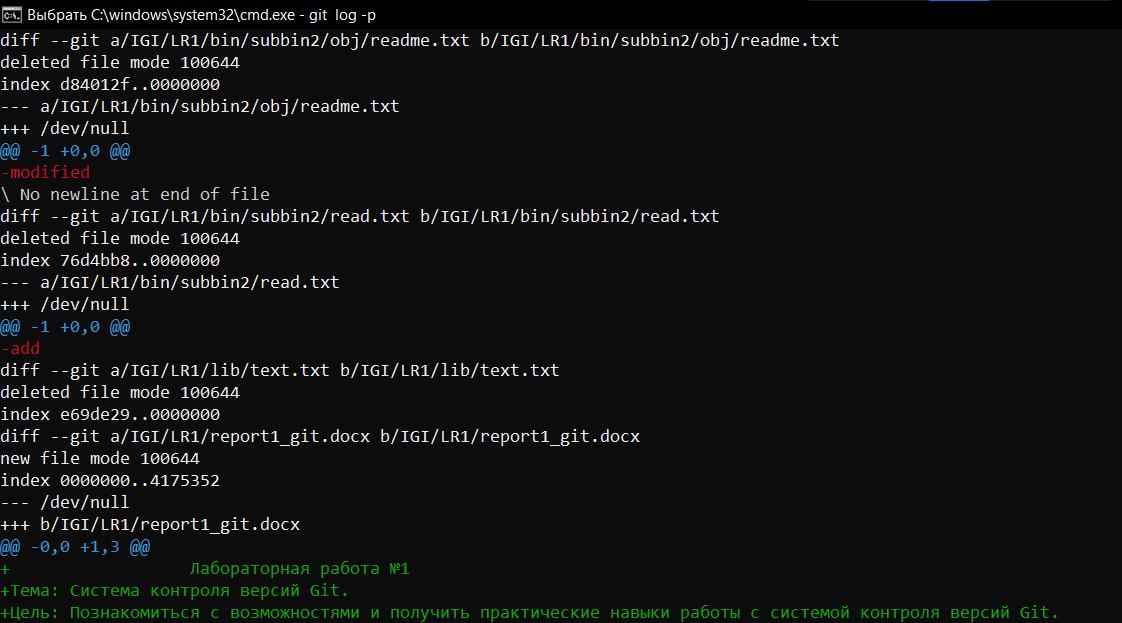


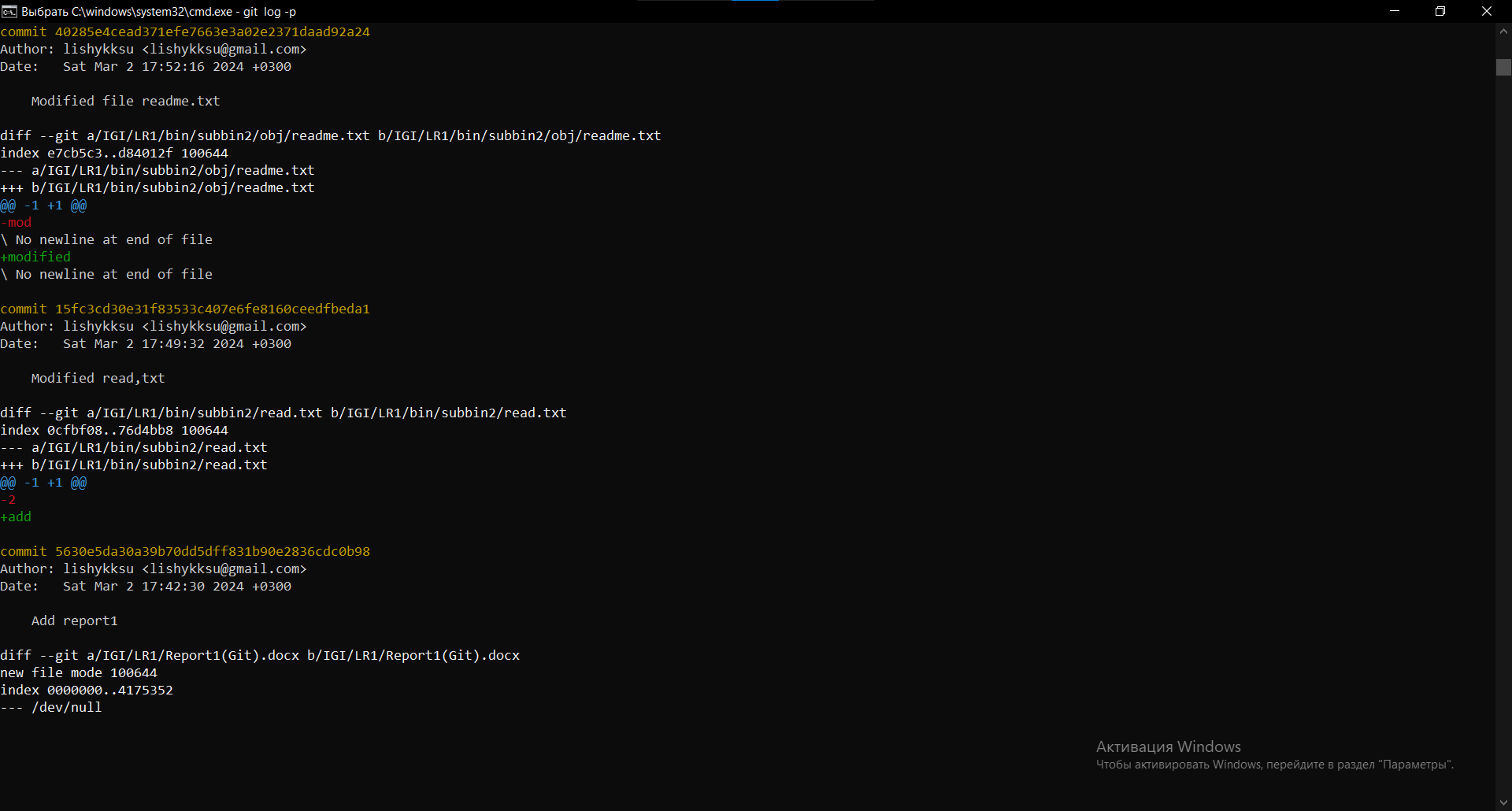


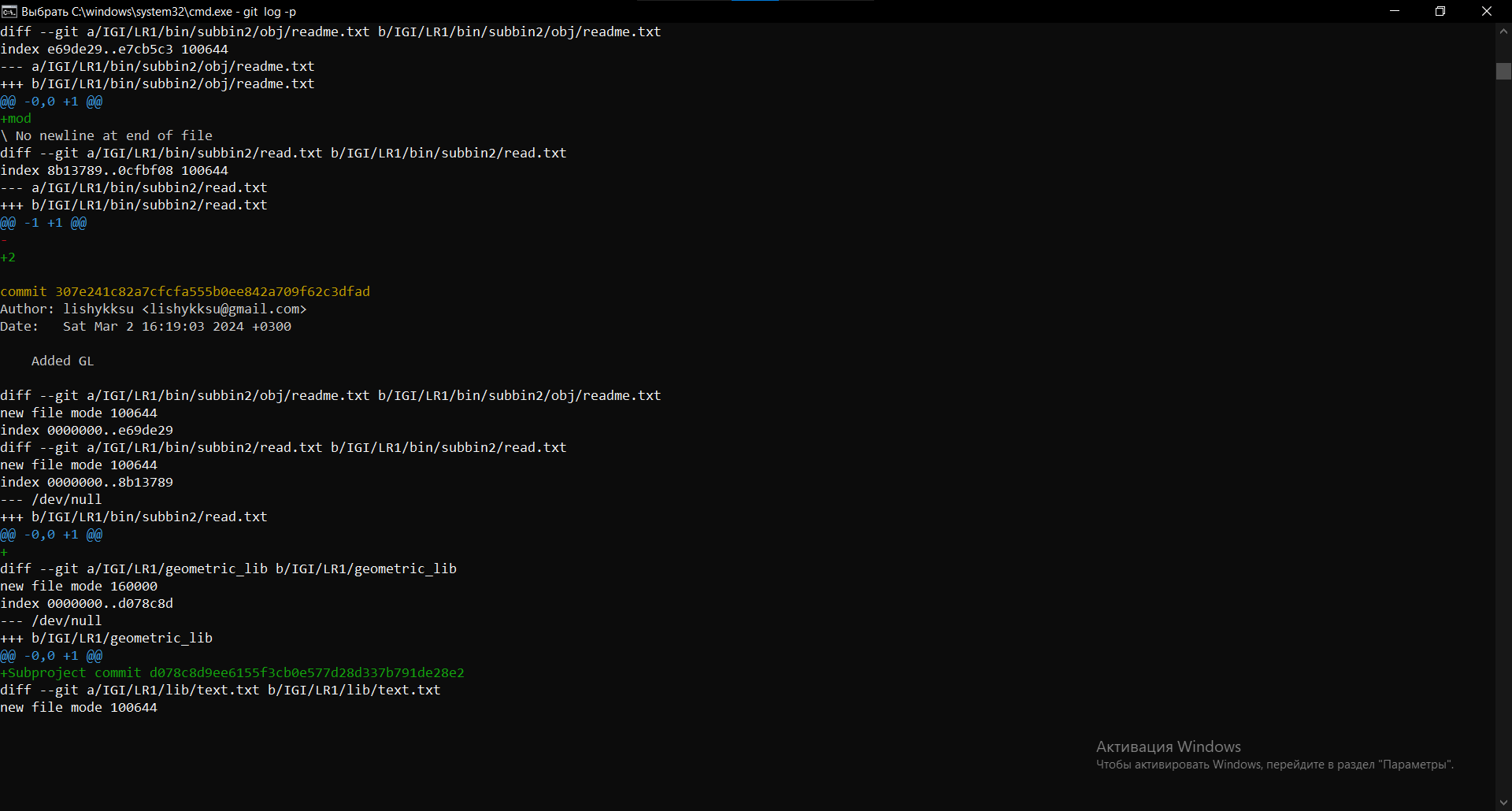
1. Исследуйте флаги для git log: p , stat ,shortstat, name-only, name-status, relative-date, pretty, graph:
2. -p – Разница в изменениях каждого коммита



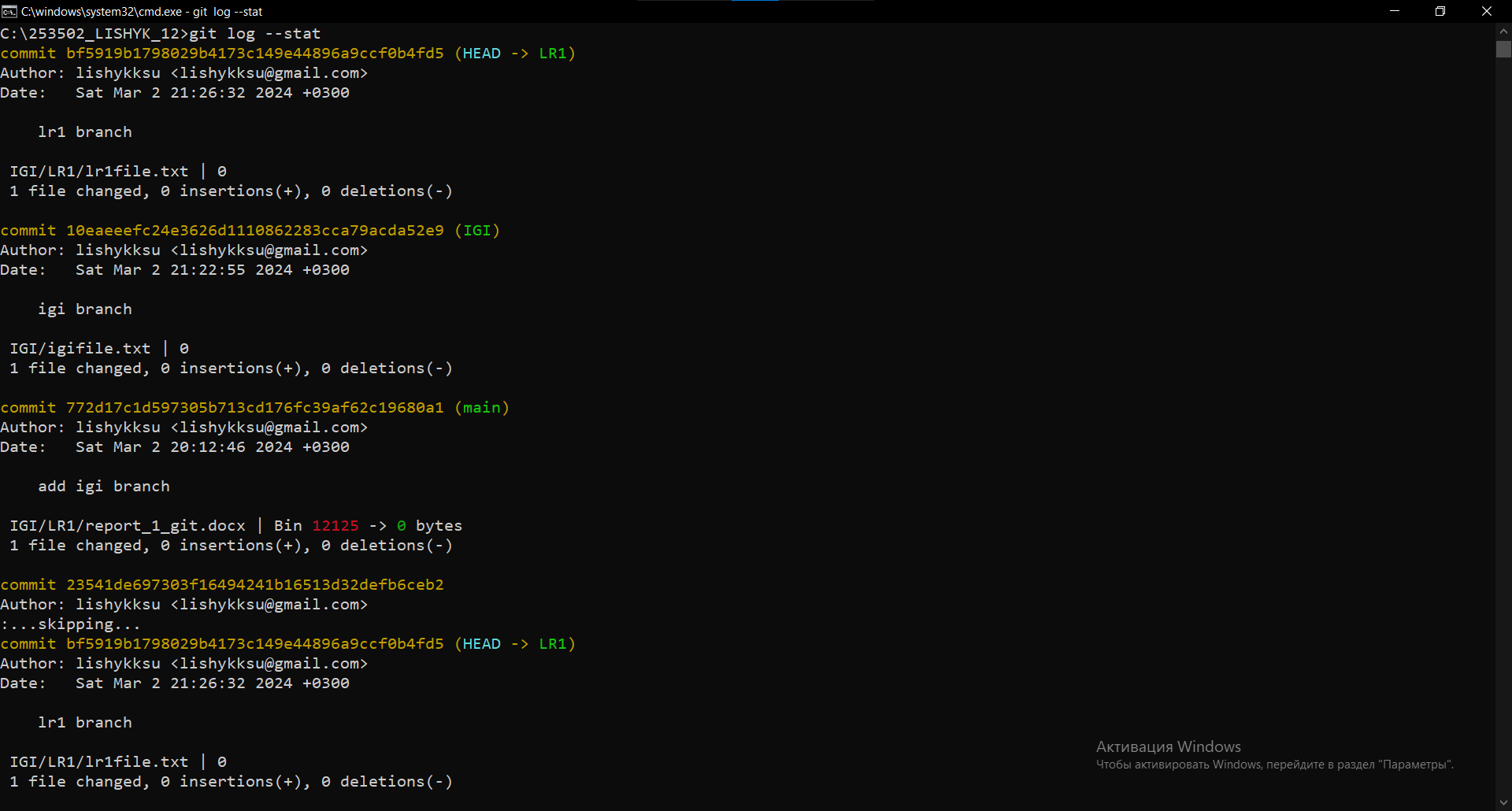


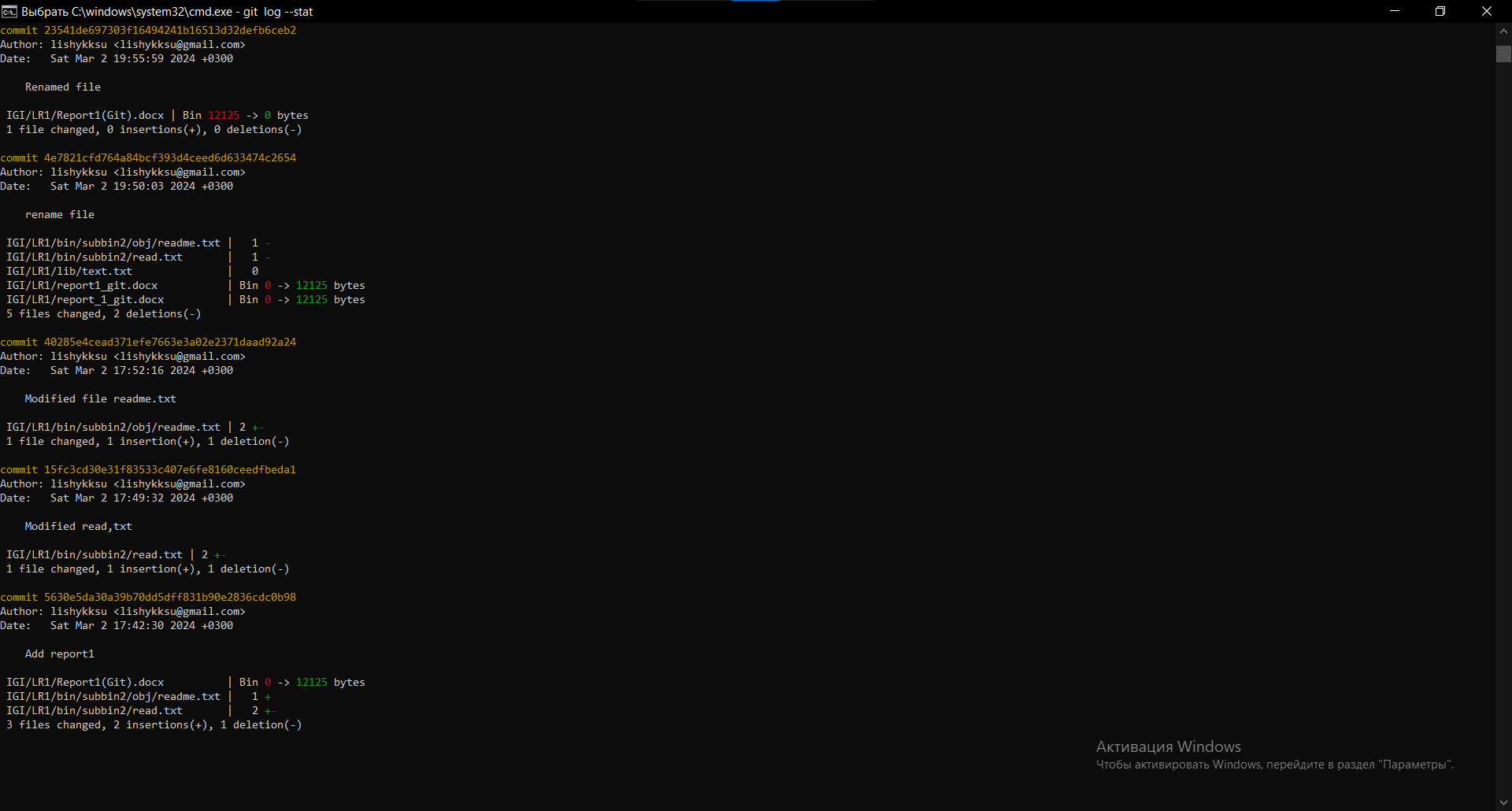


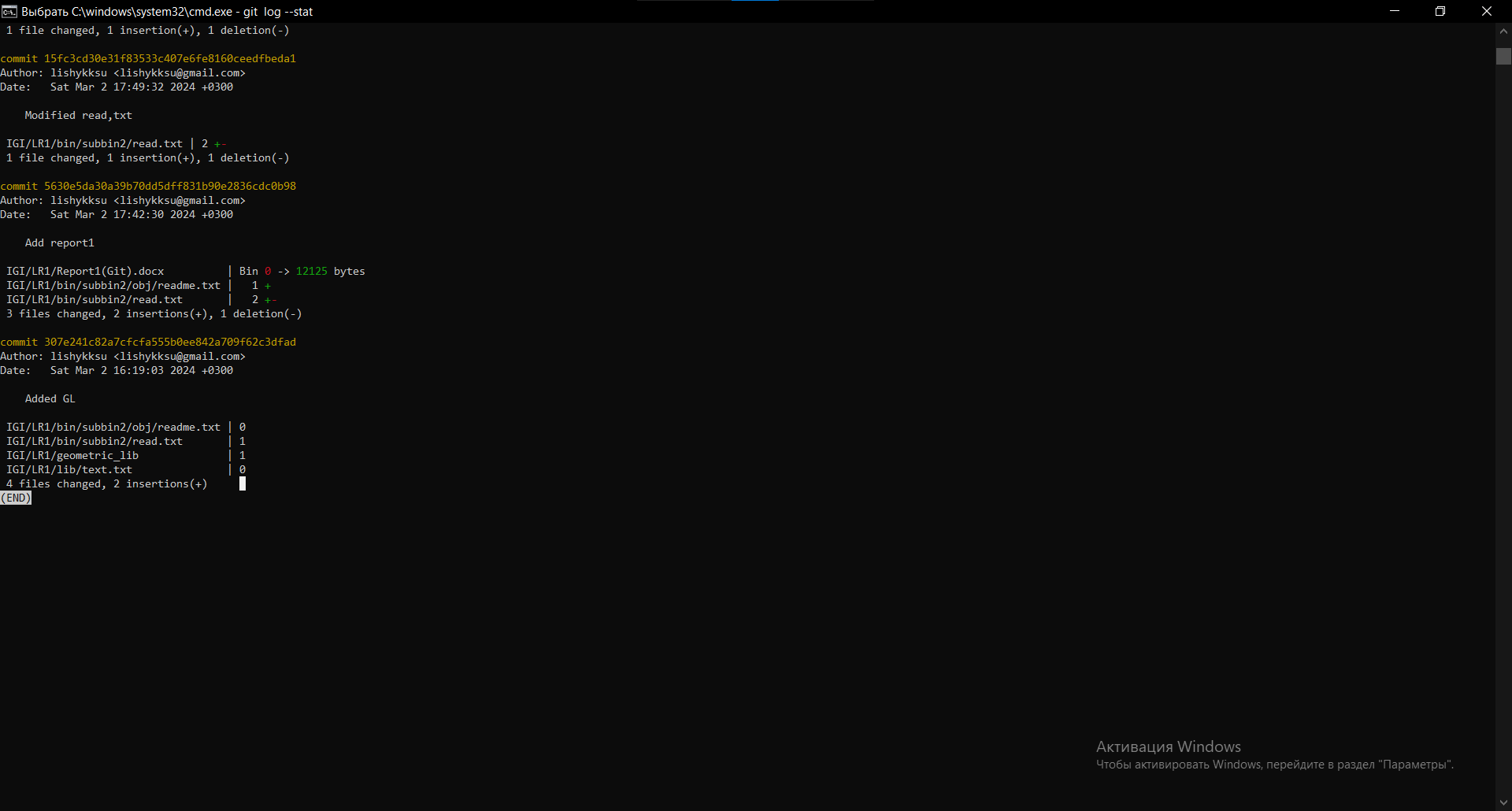




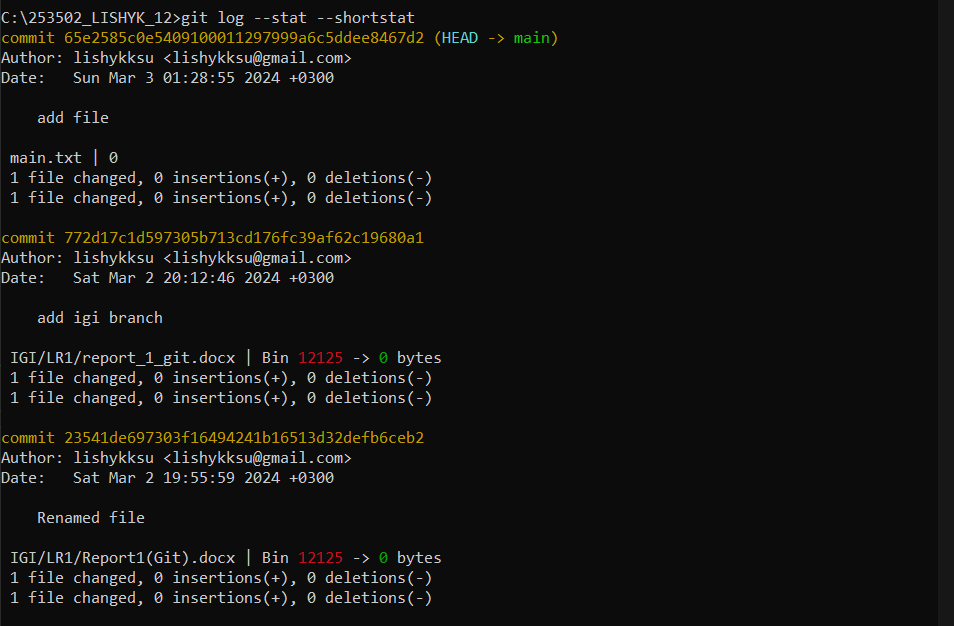
1. -- stat – Статистика для коммита

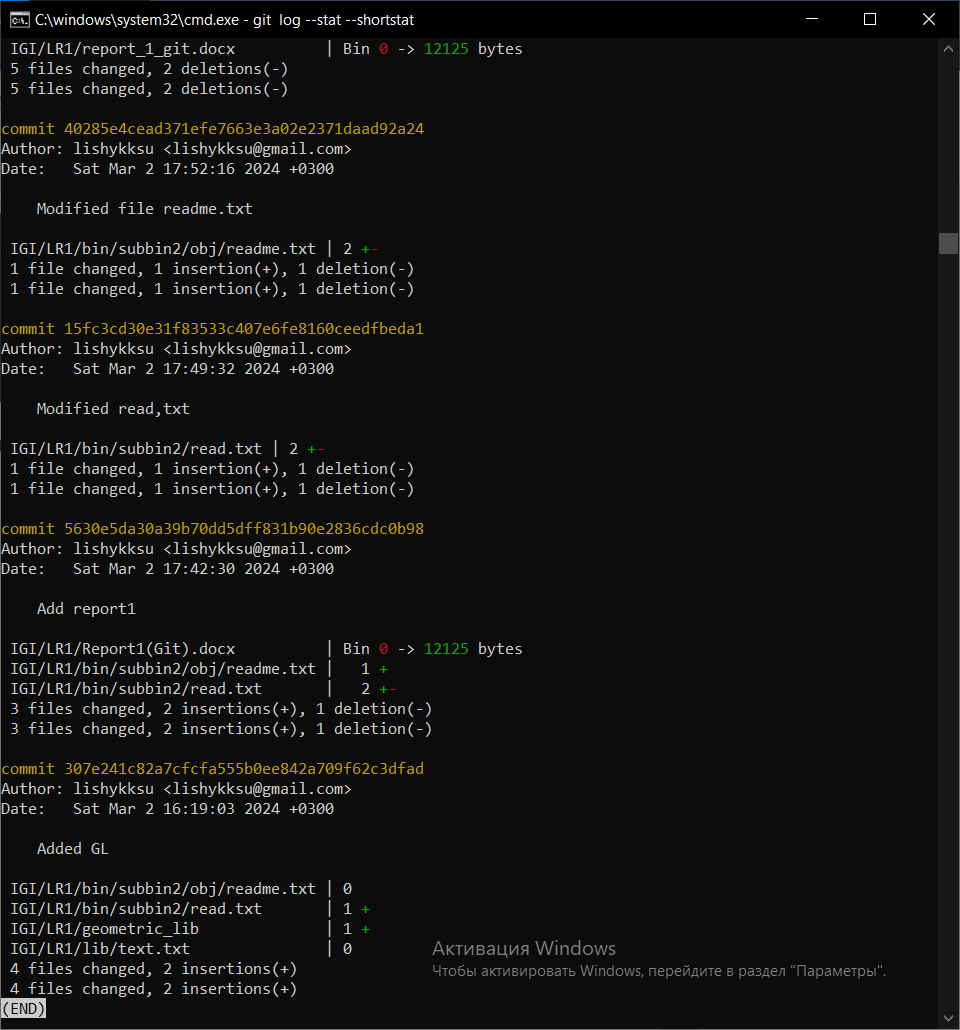




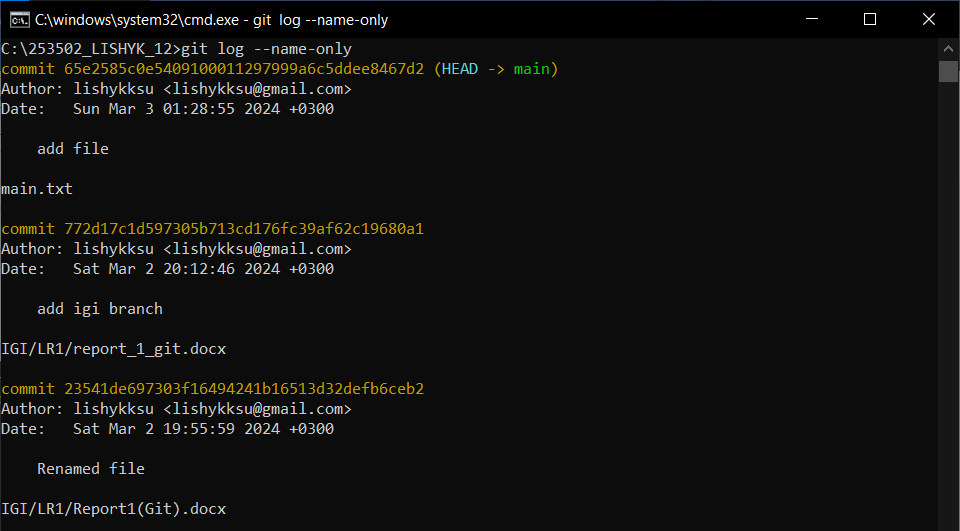


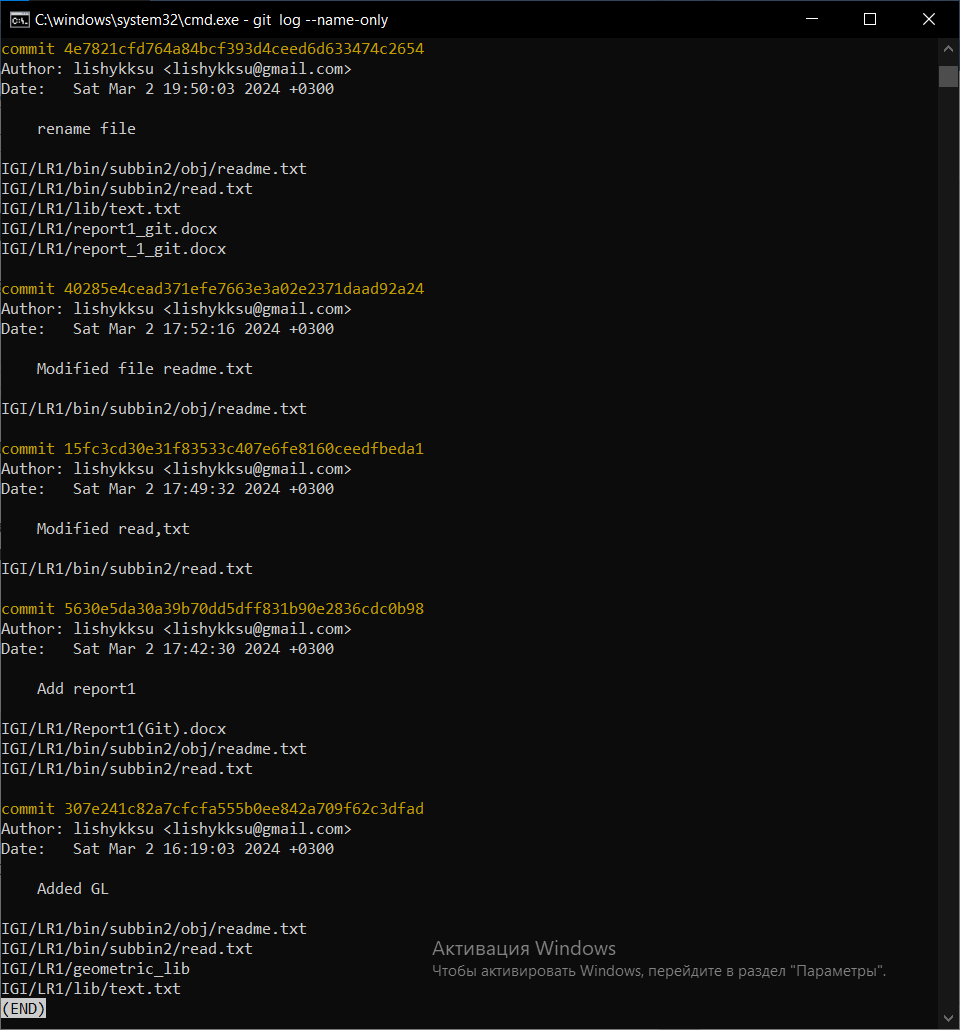
1. --shortstat – Количество изменений/вставок/удалений при применении команды –stat



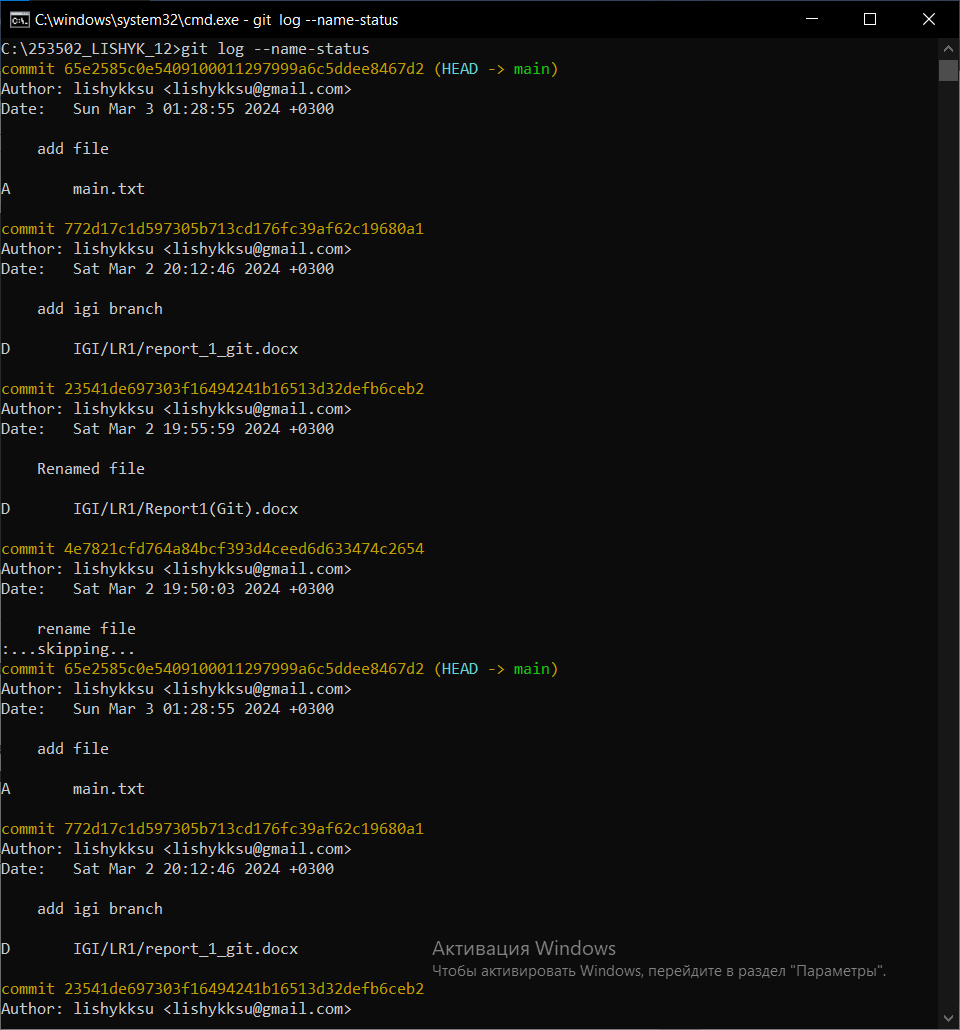


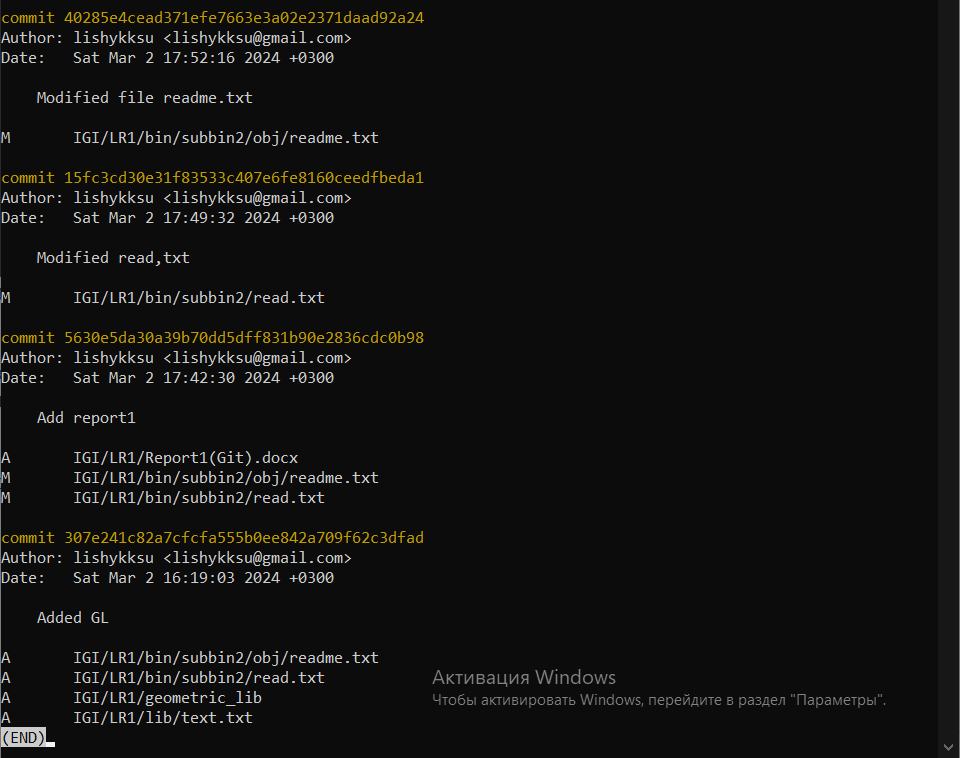
1. --name-only – Отображает информацию о коммите и имена измененных файлов



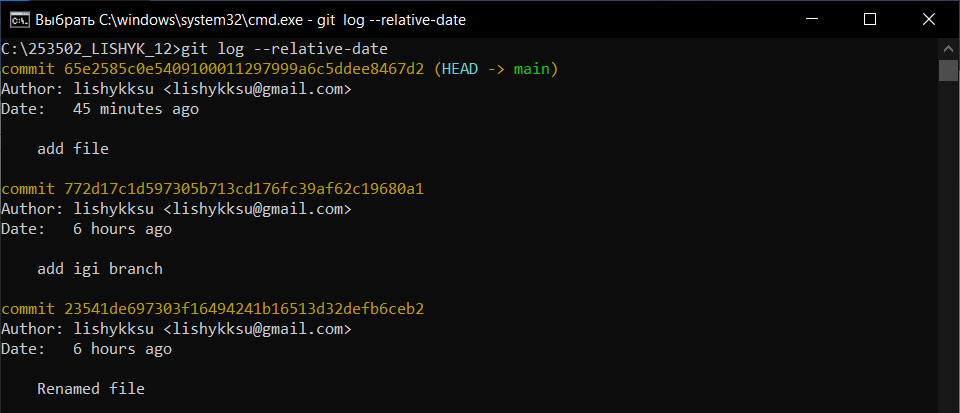


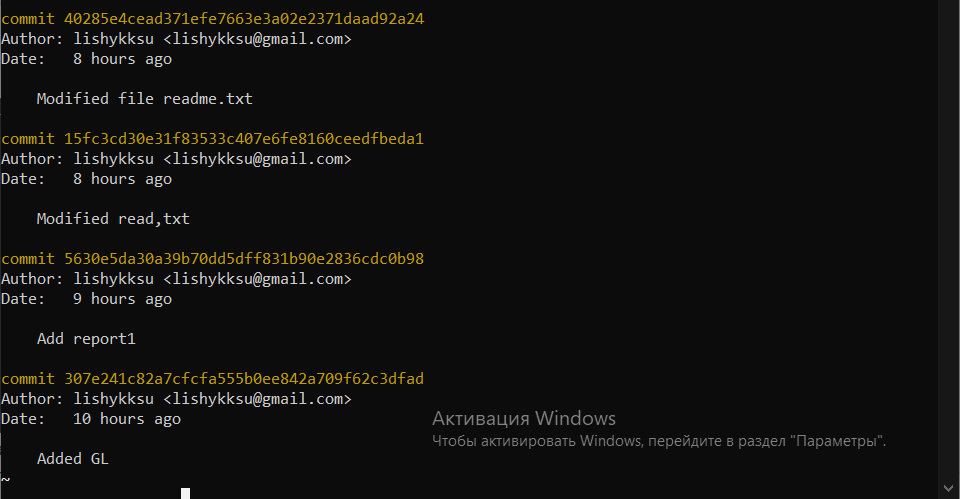
1. --name-status – Выводит сообщение о списке файлов, которые были изменены, в т.ч. добавлены и удалены.



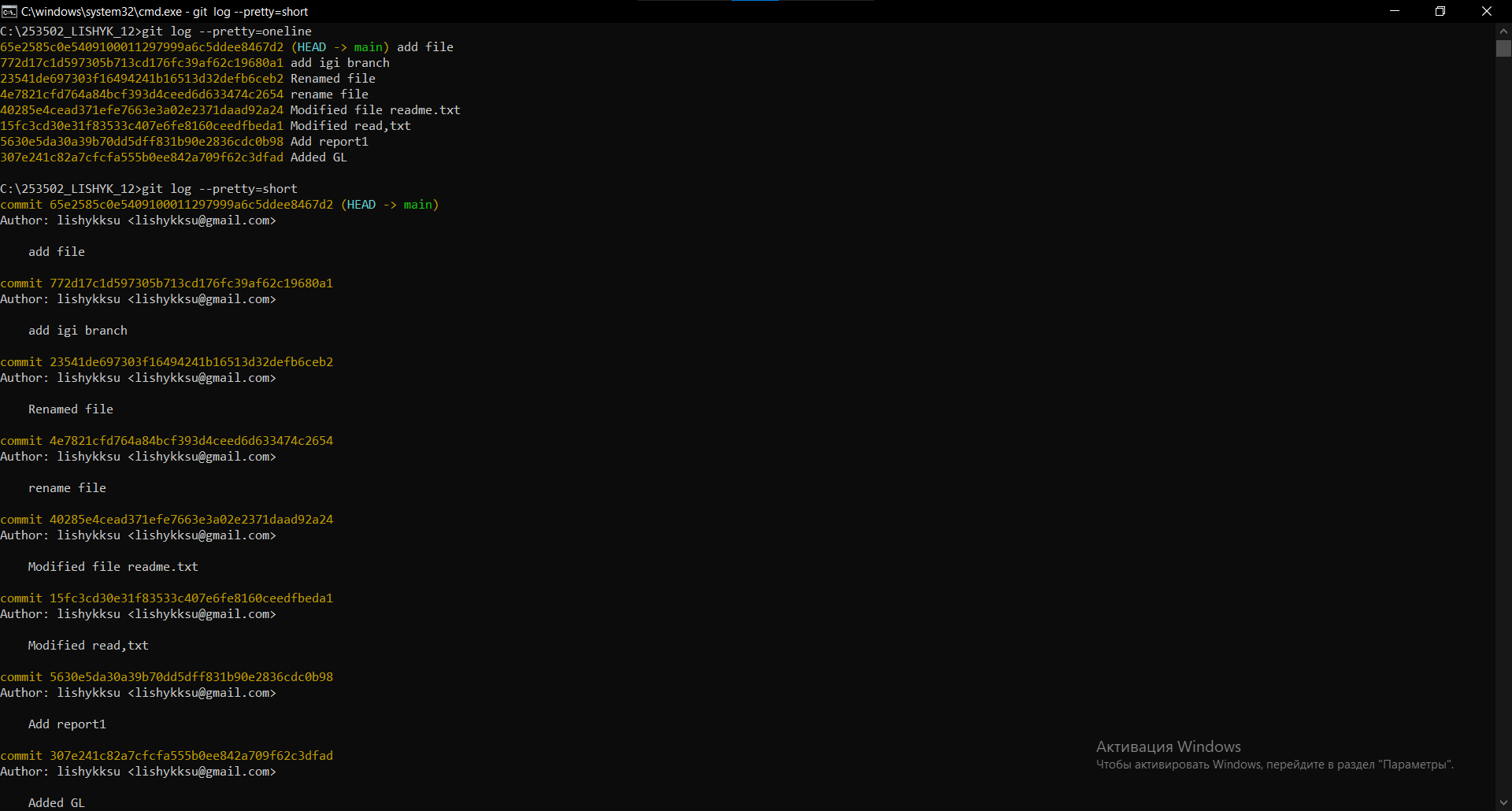


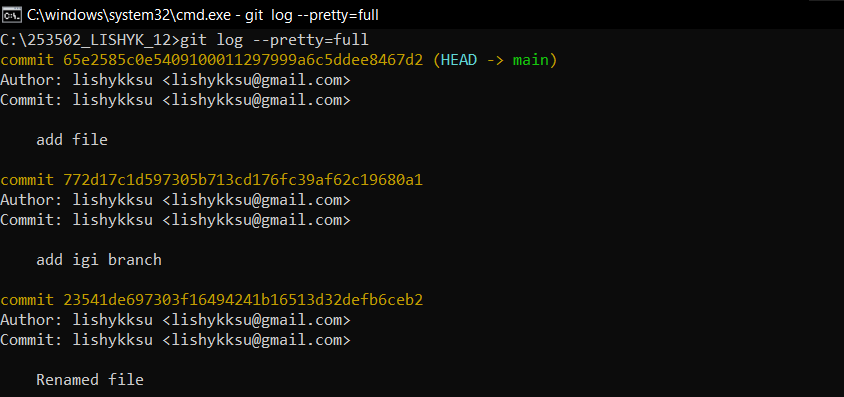
1. --relative-date – Отображает дату применения изменений в относительном формате

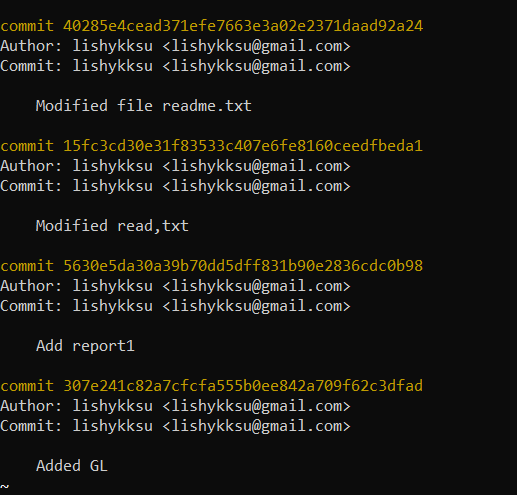




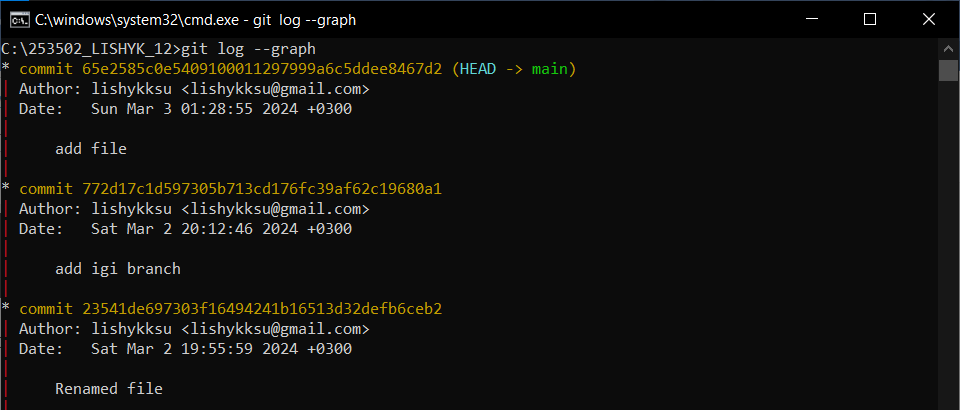
1. --pretty=oneline/full/short – Меняет формат вывода истории коммитов (в одну строку, полный вывод, краткий вывод)

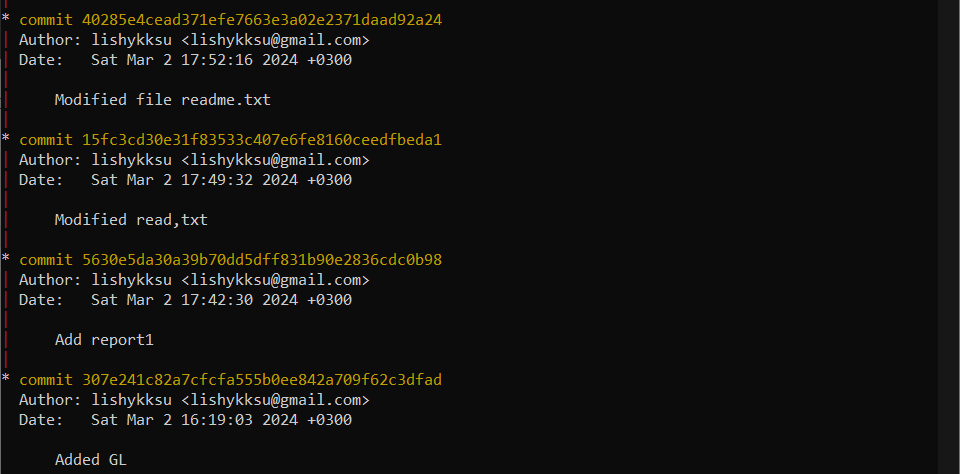






1. --graph – Показывает ветвление веток Git и историю





1. Выведите историю коммитов: полностью и с ограничением вывода – отобрать только последние,  до и после указанной даты, по автору файлов, коммитов (флаги n, since, after, until, before, author, committer, grep)
2. –n
3. --since, --after
4. --until, --before
5. –author
6. –committer
7. –grep
8. Найдите разницу в изменениях для каждого коммита
9. Создайте новый файл test.txt и вставьте в него строчку 'номер варианта'. затем сделайте коммит и отмените его.
10. Проиндексируйте файл в вашем Git-репозитории, затем отмените его индексацию.
11. Внесите изменения в файл в вашем Git-репозитории и затем верните его в исходное состояние.

**Задание 5.** Выполните следующие действия с удаленным репозиторием:

1. создайте удаленный репозиторий (на github.com)
2. Просмотрите информацию о нем, внесите изменения и опять просмотрите (remote, fetch)
3. Измените несколько файлов в вашем проекте и перенесите их в удаленный репозиторий.
4. Переименуйте свой удаленный репозиторий.
5. Получите данные из удаленного репозитория
6. Просмотреть историю изменений.