**Лабораторная работа №1**

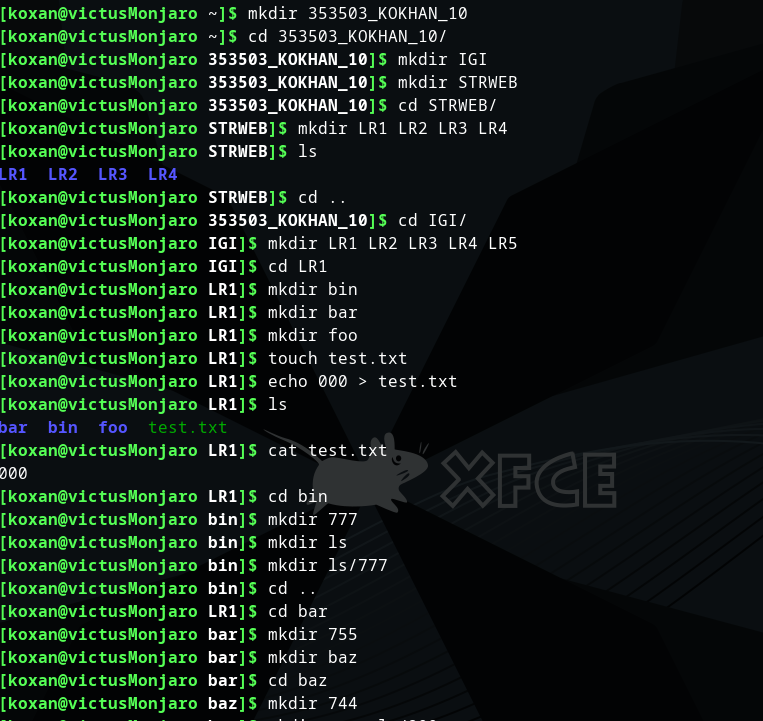
**Выполнил**: Кохан Артём 253503. Вариант 10.

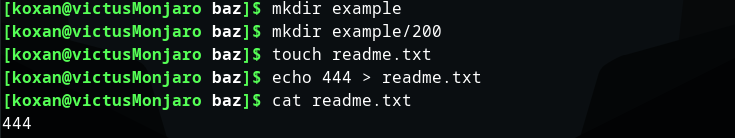
**Тема**: Система контроля версий Git.

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

**Задание:**

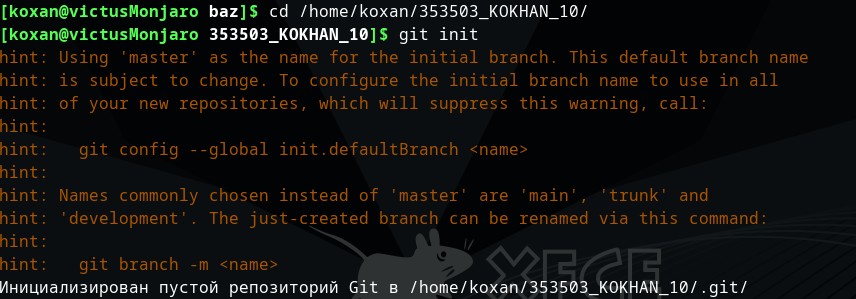
1. *Создание структуры папок (использовать командную строку) и поместить в репозиторий Git на своем локальном компьютере*Создание структуры папок:



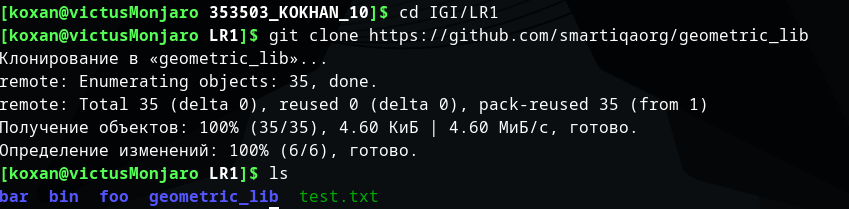




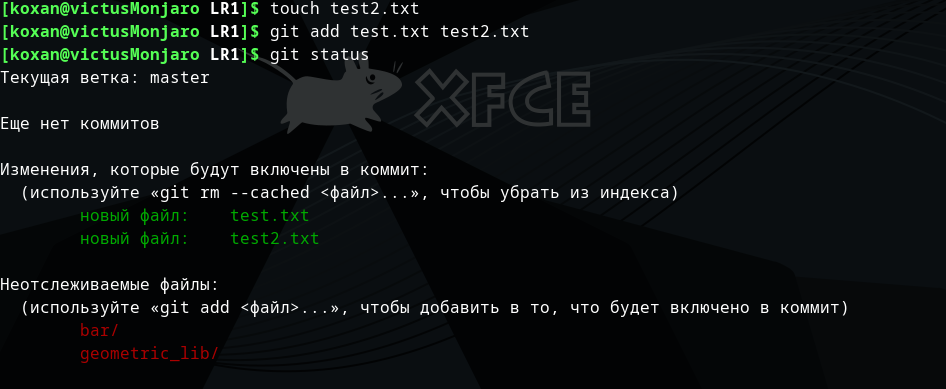
Создание локального репозитория:

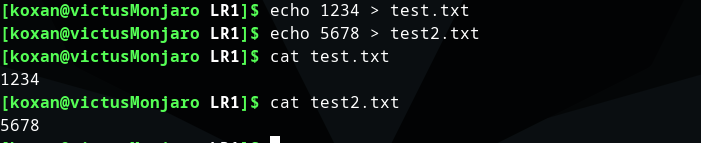


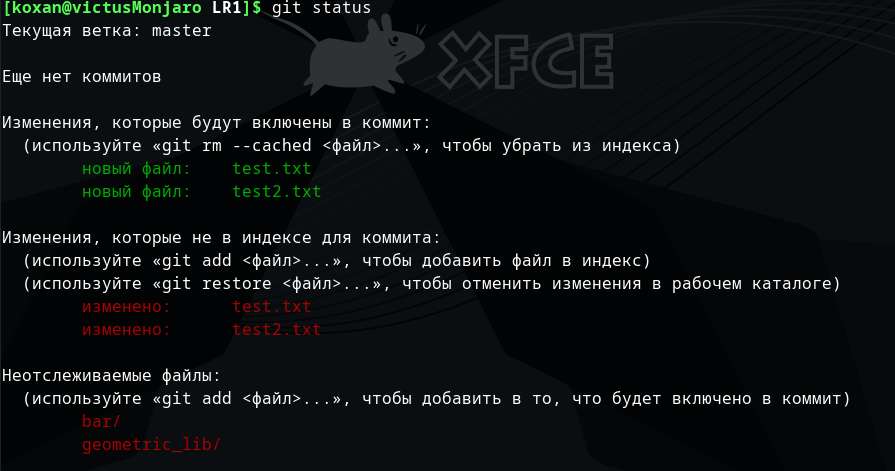
1. *Создайте репозиторий внутри папки LR1 с помощью клонирования проекта с сайта github.com (*[*https://github.com/smartiqaorg/geometric\_lib*](https://github.com/smartiqaorg/geometric_lib)*)*

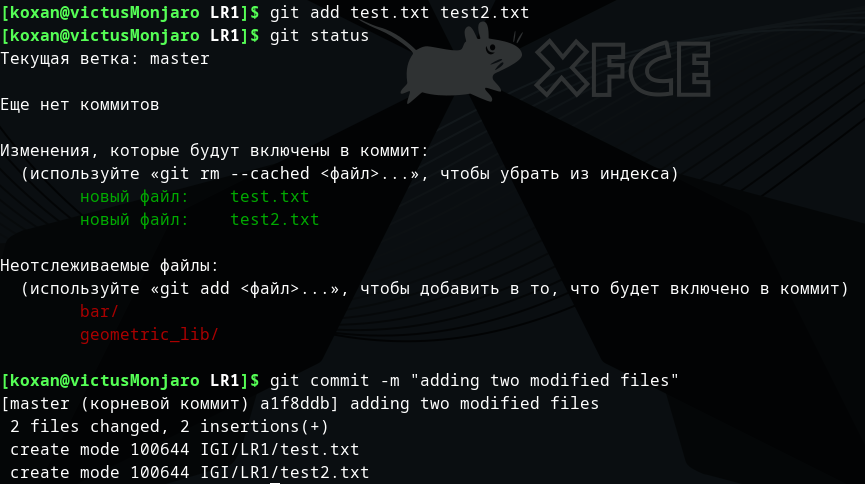


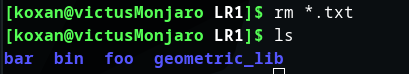
1. *Выполните следующие действия, каждый раз проверяя состояние файлов в локальном репозитории до и после выполнения операции:*
   1. сделать индексацию нескольких файлов в вашем Git-репозитории (git add …)



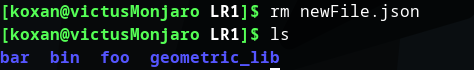
* 1. внесите изменения в файлы



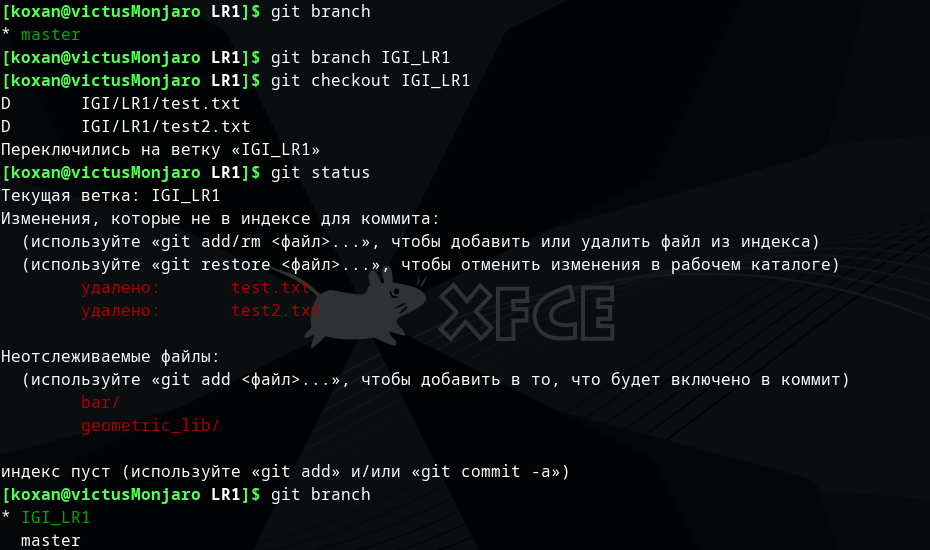
* 1. выполните коммиты к проиндексированным файлам
  2. добавьте и удалите файлы (один и группу файлов с определенным расширением)

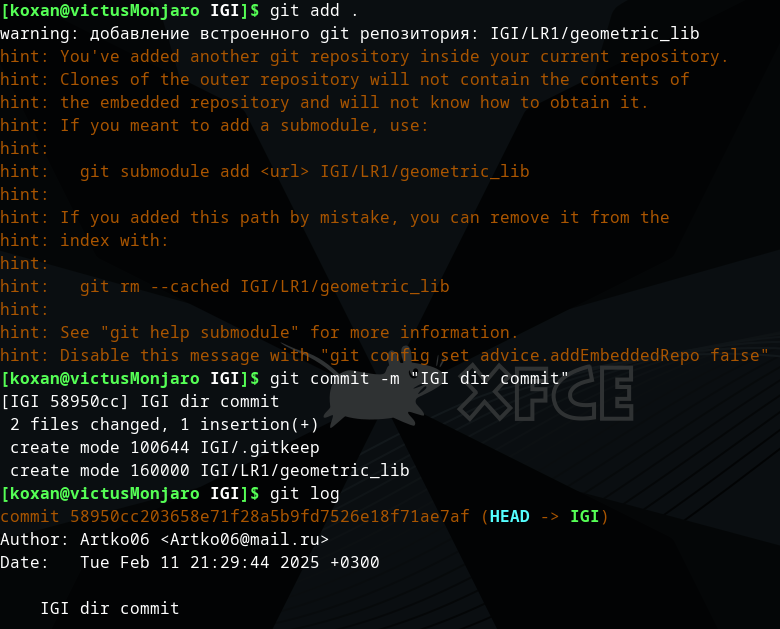
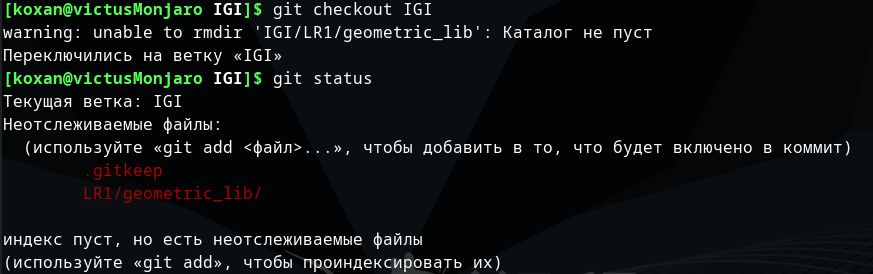
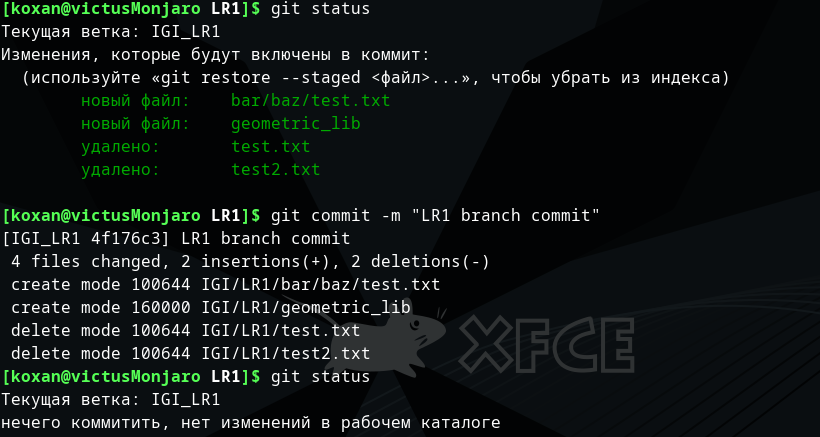
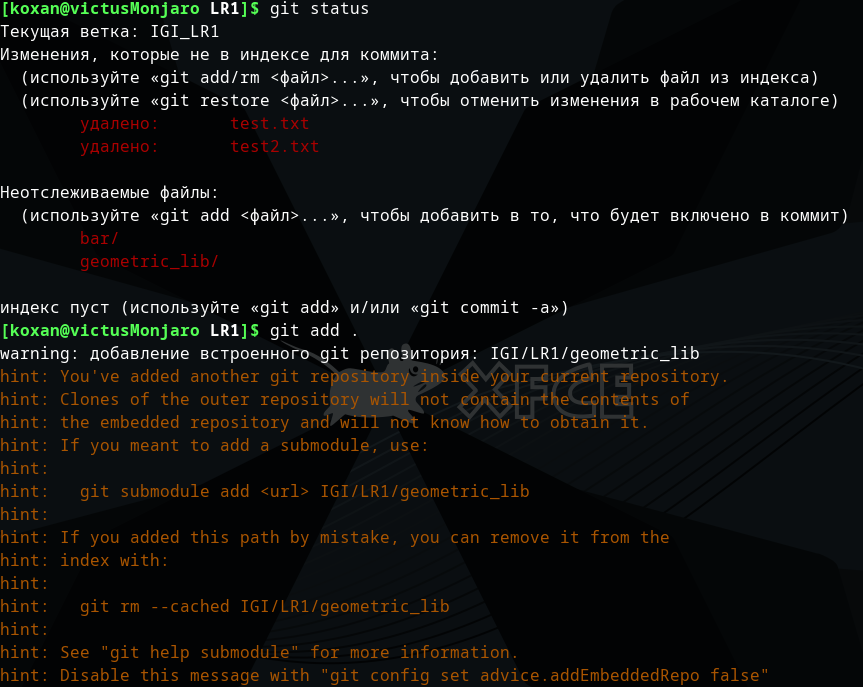
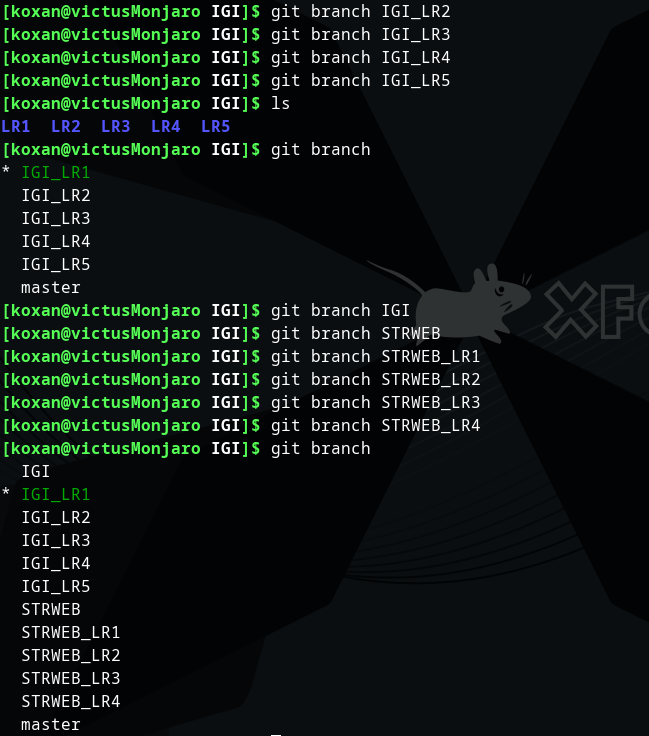






* 1. переименуйте файлы
  2. создайте для IGI и STRWEB, а также для каждой ЛР, вложенной в них, отдельные ветки, переместите папки и файлы на соответствующие ветки



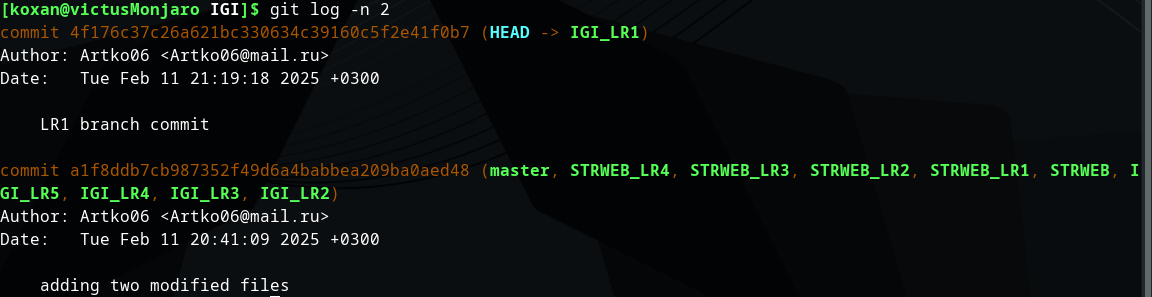


* 1. выведите историю коммитов: полностью и с ограничением вывода – отобрать только последние, до и после указанной даты, по автору файлов, коммитов (флаги n, since, after, until, before, author, committer, grep)

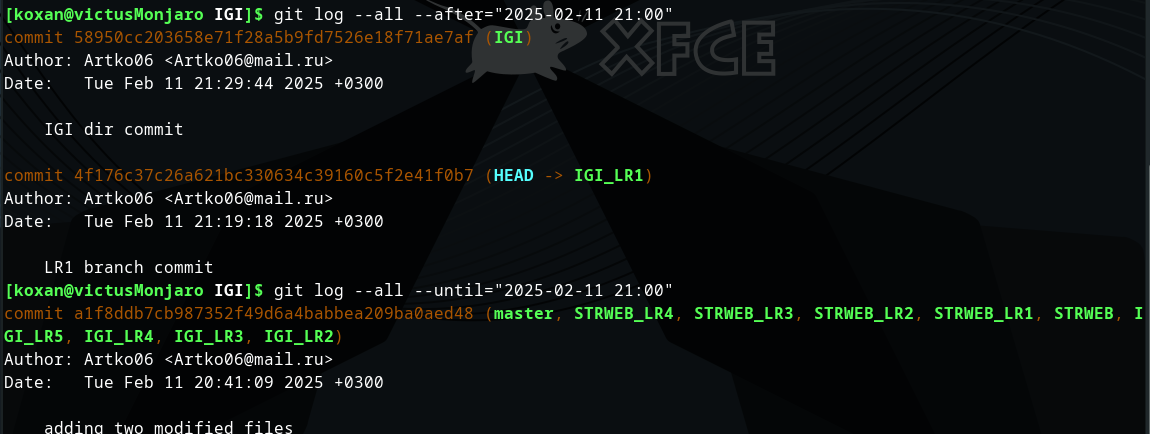
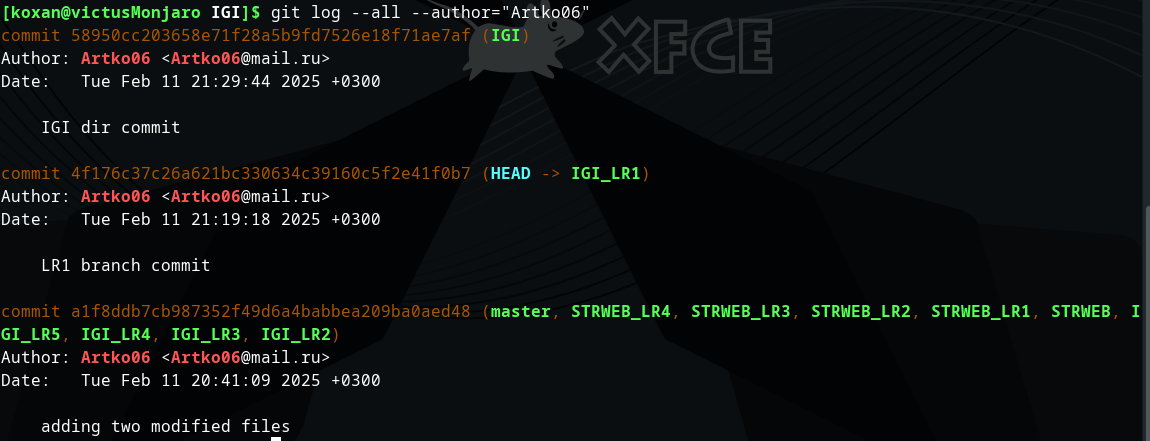
Все:

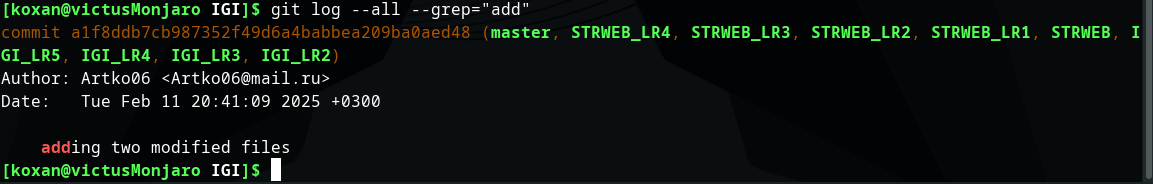


Флаг n:



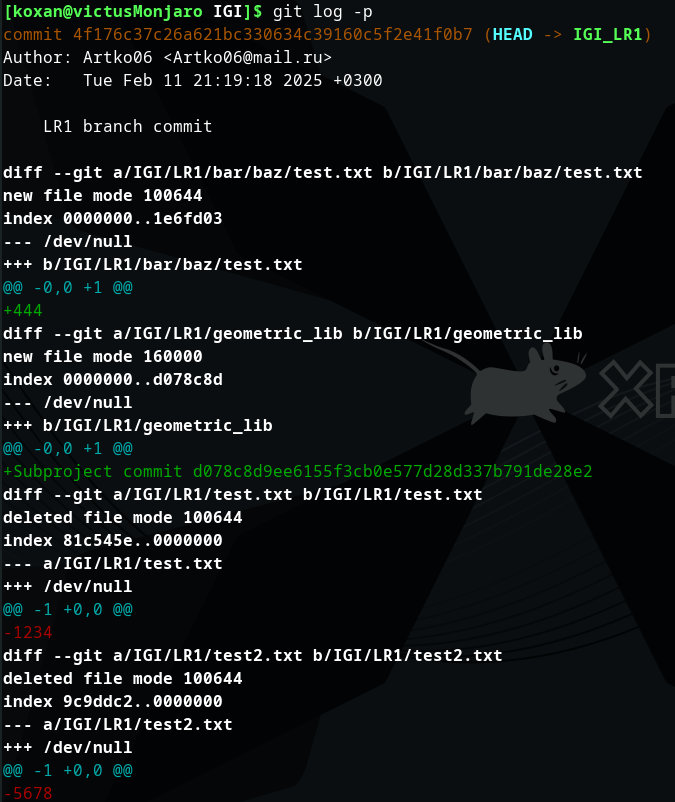
Since/after и until/before:

Author/committer:

Grep:

* 1. исследуйте флаги для git log: p , stat ,shortstat, name-only, name-status, relative-date, pretty, graph,

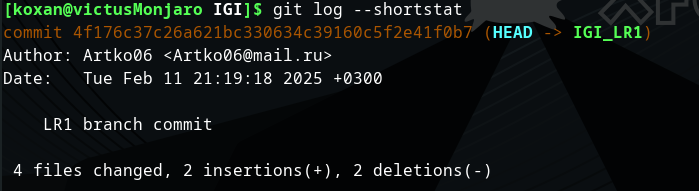
p или patch (изменения в каждом коммите):



stat (Показывает статистику изменений):



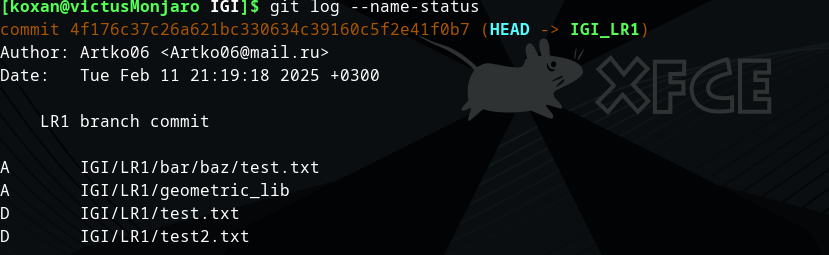
shortstat (показывает только итоговую статистику изменений):



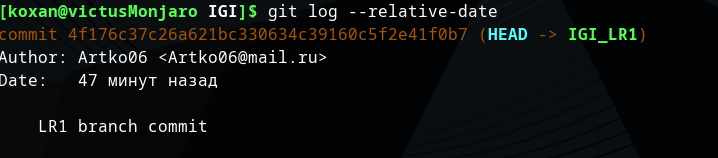
name-only (Выводит только названия файлов, изменённых в каждом коммите):



name-status (Выводит названия файлов и тип изменений для каждого файла):



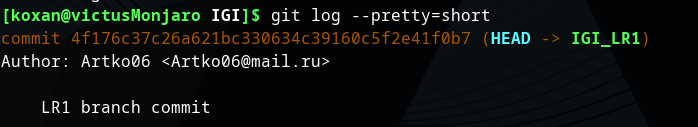
relative-date (Отображает дату в формате, относительно текущего времени):



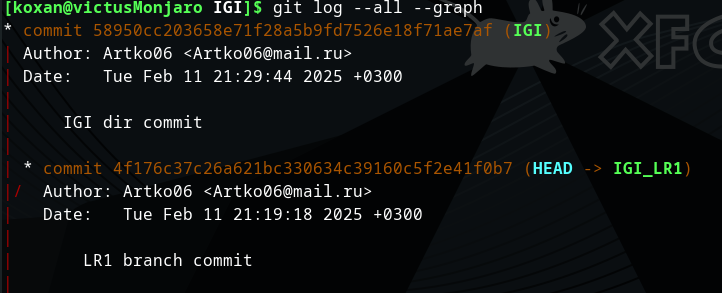
pretty (Позволяет кастомизировать формат вывода коммитов):



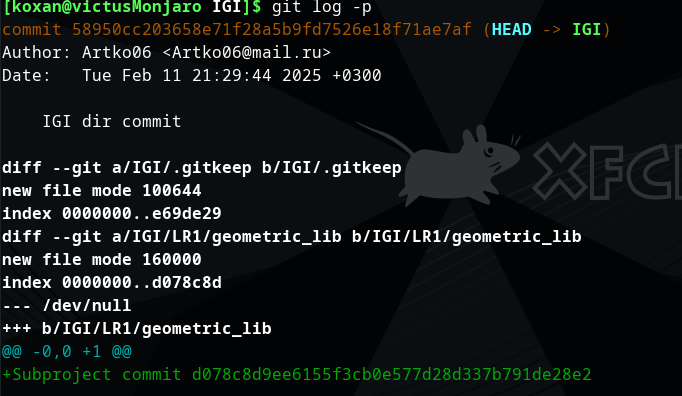




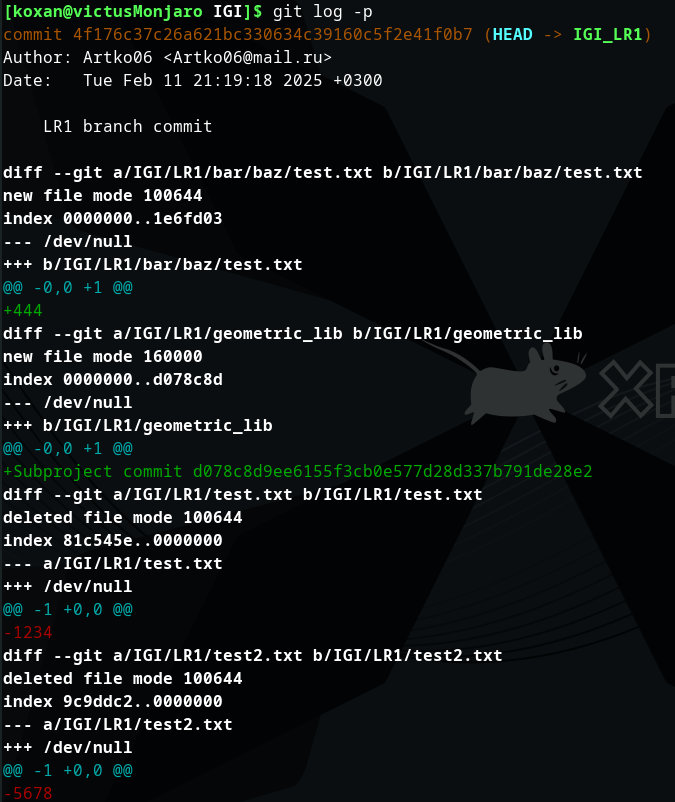
graph (Выводит графическое представление дерева коммитов, показывающее связи между ветками):



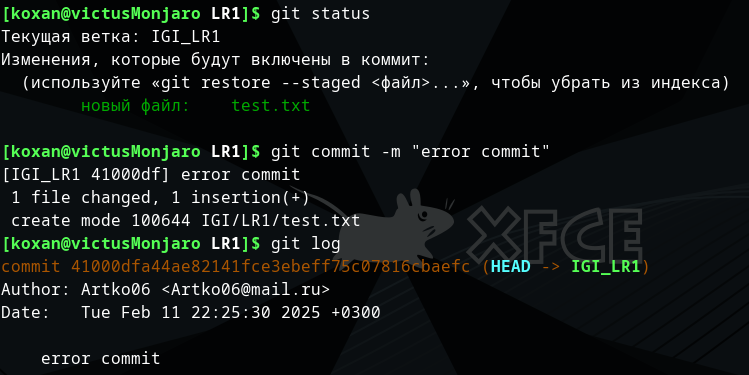
* 1. найдите разницу в изменениях для каждого коммита.

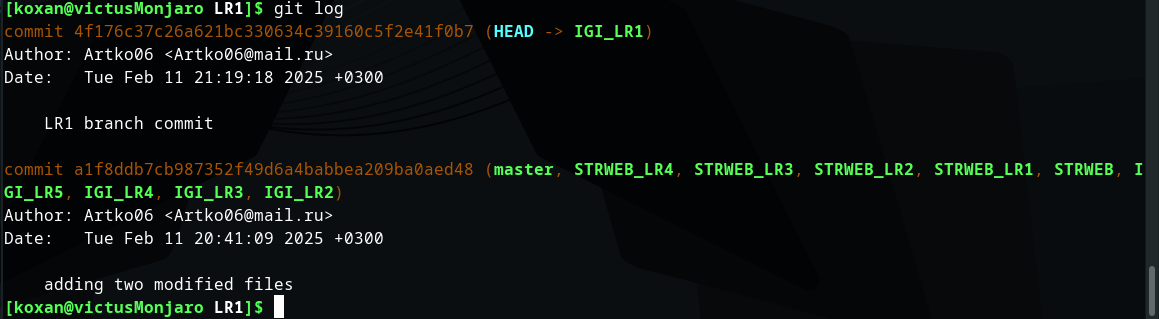




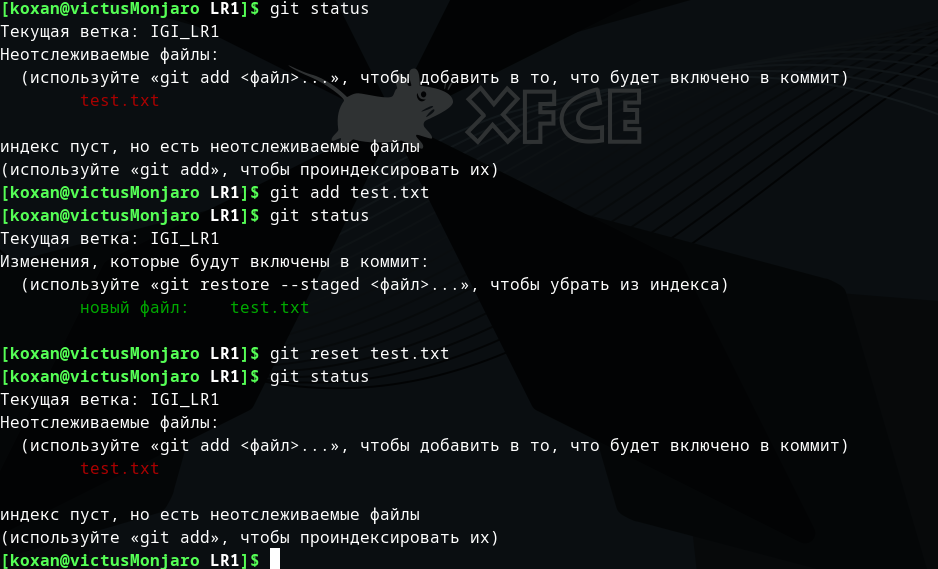


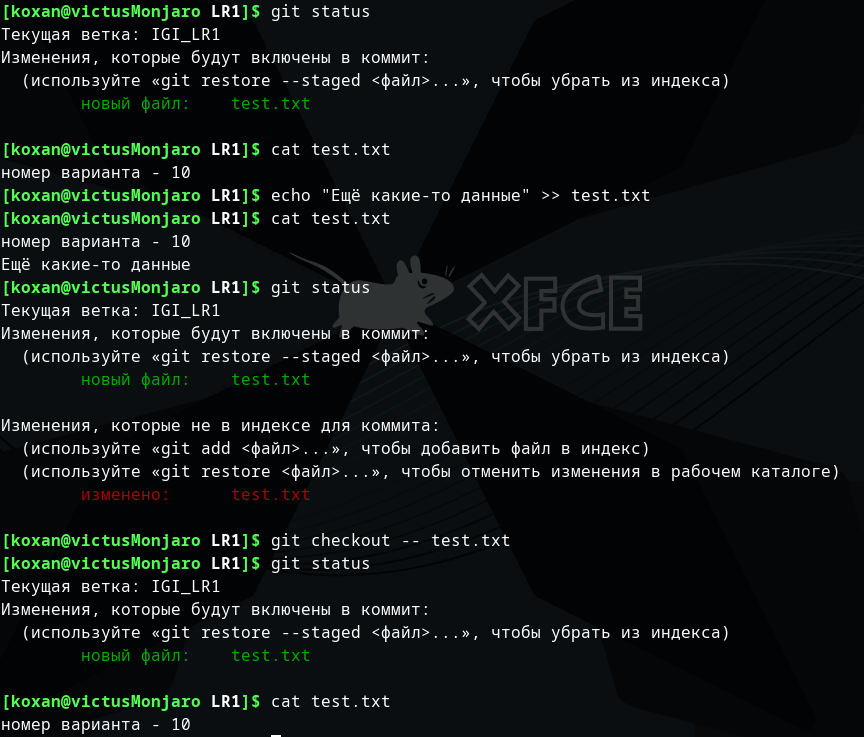
* 1. создайте новый файл test.txt и вставьте в него строчку 'номер варианта'. затем сделайте коммит и отмените его



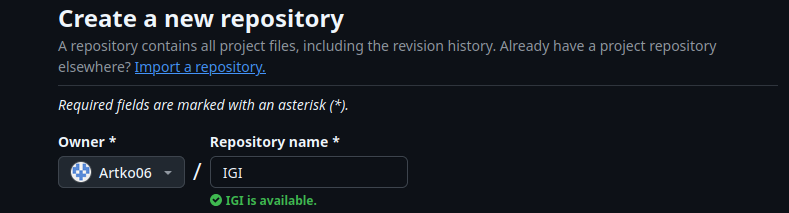


* 1. проиндексируйте файл в вашем Git-репозитории, затем отмените его индексацию

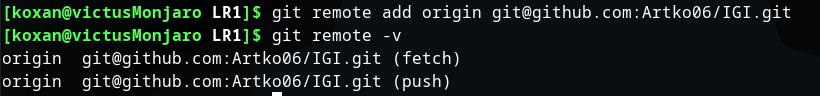


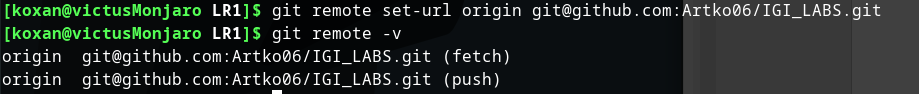
* 1. внесите изменения в файл в вашем Git-репозитории и затем верните его в исходное состояние.

1. *Выполните следующие действия с удаленным репозиторием:*
   1. создайте удаленный репозиторий (на github.com)

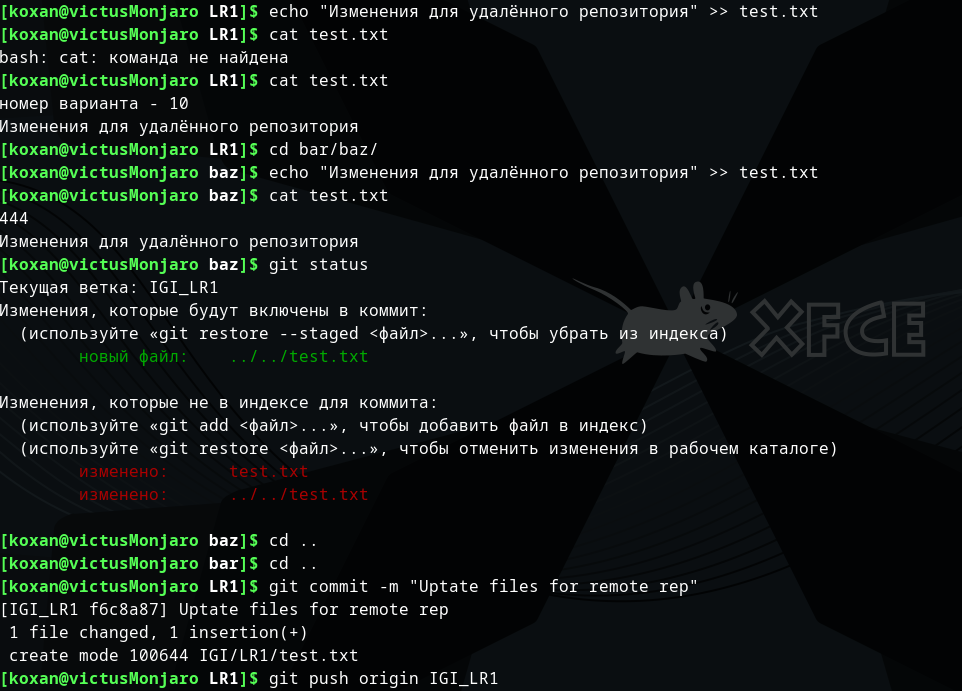


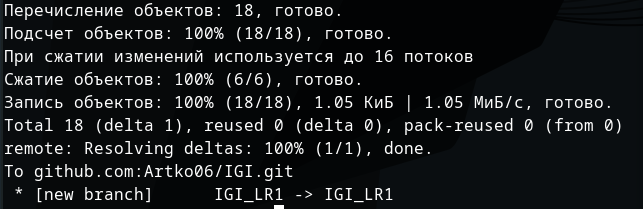
* 1. просмотрите информацию о нем, внесите изменения и опять просмотрите (remote, fetch).





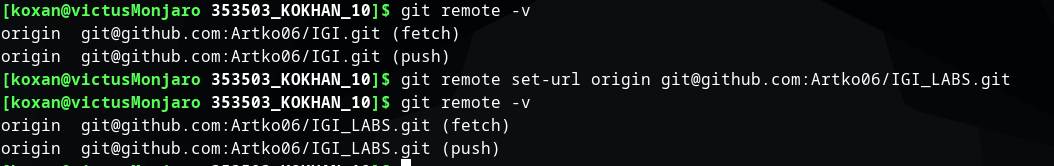
* 1. измените несколько файлов в вашем проекте и перенесите их в удаленный репозиторий,



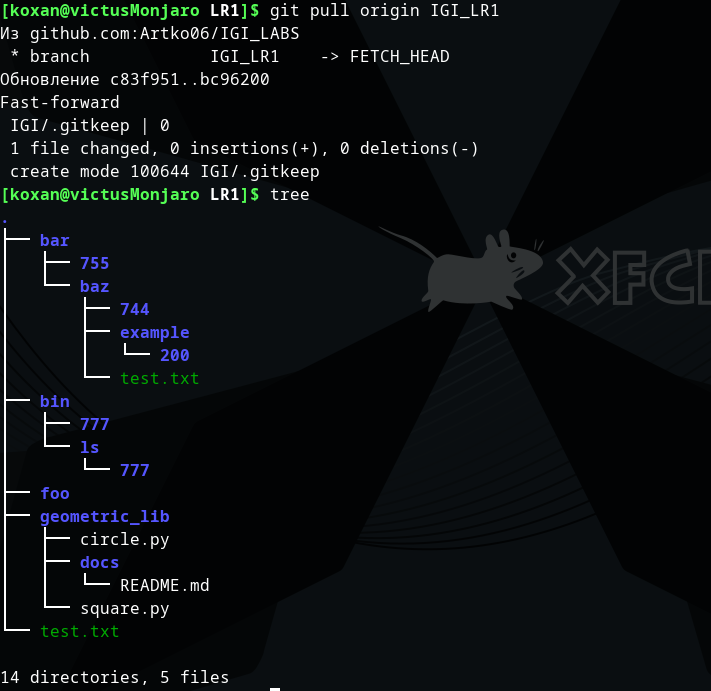


* 1. переименуйте свой удаленный репозиторий





* 1. получите данные из удаленного репозитория

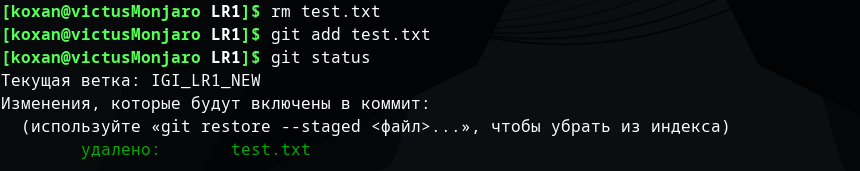


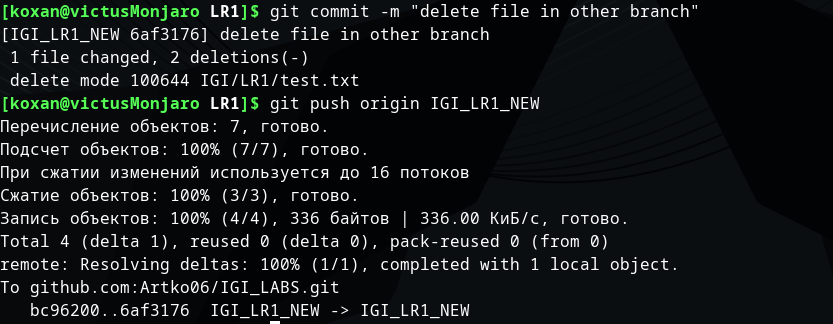
* 1. просмотреть историю изменений



* 1. создайте дополнительную ветку 1 в удаленном реппозитории, перенесите в нее два файла из другой ветки, удалите один из файлов







* 1. создайте дополнительную ветку 2 в удаленном реппозитории, перенесите в нее два файла из вашего проекта, слить изменения в основную ветку, удалить дополнительную ветку 2

