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 №1

на тему:

**«Система контроля версий GIT»**

БГУИР 6-05-0612-02 03

|  |
| --- |
| Выполнил студент группы 353504  ВАЛИУЛЛИН Константин Алексеевич |
|  |
| (дата, подпись студента) |
| Проверил доцент каф. Информатики  ЖВАКИНА Анна Васильевна |
|  |
| (дата, подпись преподавателя) |

Минск 2025

# 1 Индивидуальное задание

**Вариант 3.**1. Подготовьте рабочее окружение в соответствии с типом вашей операционной системы (https://smartiqa.ru/courses/git/lesson-1 ):

a. Установите Git

b. Выполните базовую настройку

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

(Наименование для базовой папки: 253501\_IVANOV\_12, где 12 – порядковый номер в журнале)

GROUP\_NUMBER\_YOUR\_LAST\_NAME\_ NUMBER\_IN\_LIST

IGI

LR1

Папки и файлы в соответствии с индивидуальным заданием

Отчет по ЛР1 со скринами действий и результатов

LR2

LR3

LR4

LR5

**STRWEB**

LR1

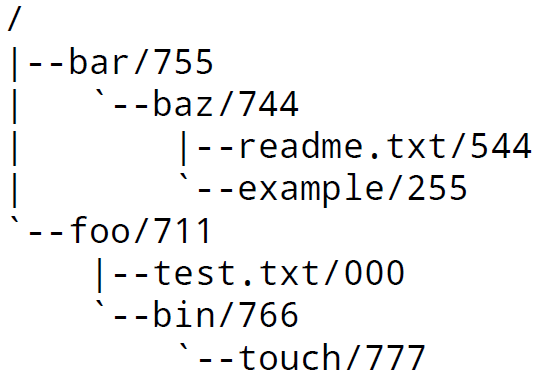
LR2

LR3

LR4

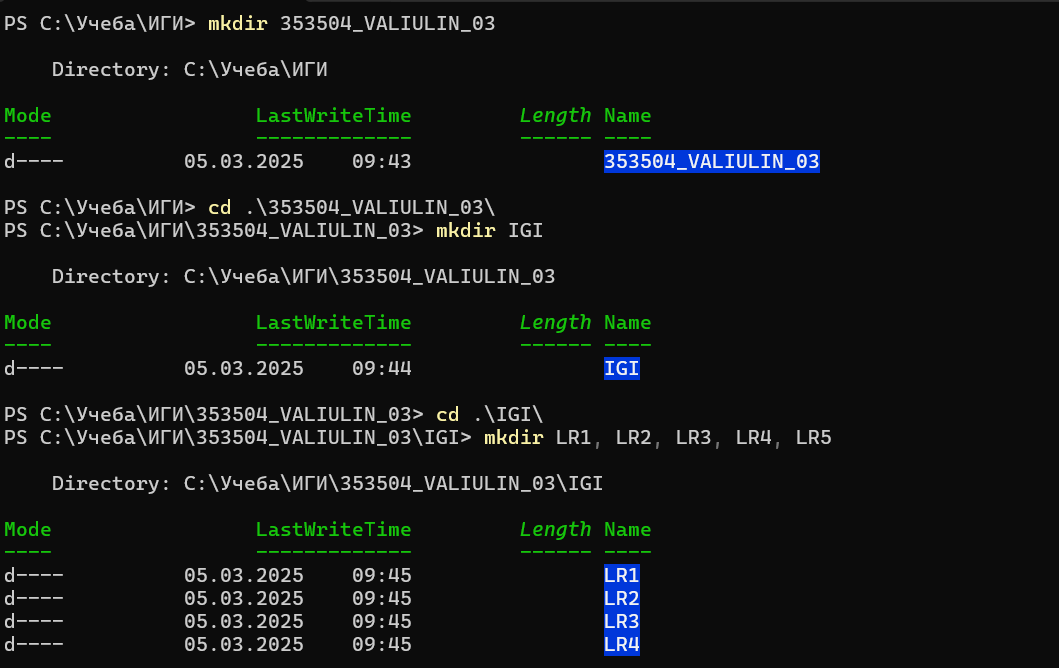
1. *Создайте репозиторий внутри папки* LR1 *с помощью клонирования проекта с сайта github.com (*https://github.com/smartiqaorg/geometric\_lib*)*
2. *Выполните следующие действия, каждый раз проверяя состояние файлов в локальном репозитории до и после выполнения операции:*
   1. сделайте индексацию нескольких файлов в вашем Git-репозитории (git add …),
   2. внесите изменения в файлы,
   3. выполните коммиты к проиндексированным файлам,
   4. добавьте и удалите файлы (один и группу файлов с определенным расширением)
   5. переименуйте файлы
   6. создайте для IGI и STRWEB, а также для каждой ЛР, вложенной в них, отдельные ветки, переместите папки и файлы на соответствующие ветки
   7. 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) <https://code.mu/ru/tool/git/basis/limit-output-commit-history-flags/>
   8. исследуйте флаги для git log: p , stat ,shortstat, name-only, name-status, relative-date, pretty, graph,
   9. найдите разницу в изменениях для каждого коммита
   10. создайте новый файл test.txt и вставьте в него строчку 'номер варианта'. затем сделайте коммит и отмените его
   11. проиндексируйте файл в вашем Git-репозитории, затем отмените его индексацию
   12. внесите изменения в файл в вашем Git-репозитории и затем верните его в исходное состояние.
3. *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
   7. создайте дополнительную ветку 1 в удаленном реппозитории, перенесите в нее два файла из другой ветки, удалите один из файлов
   8. создайте дополнительную ветку 2 в удаленном реппозитории, перенесите в нее два файла из вашего проекта, слить изменения в основную ветку, удалить дополнительную ветку 2

Структура папок:



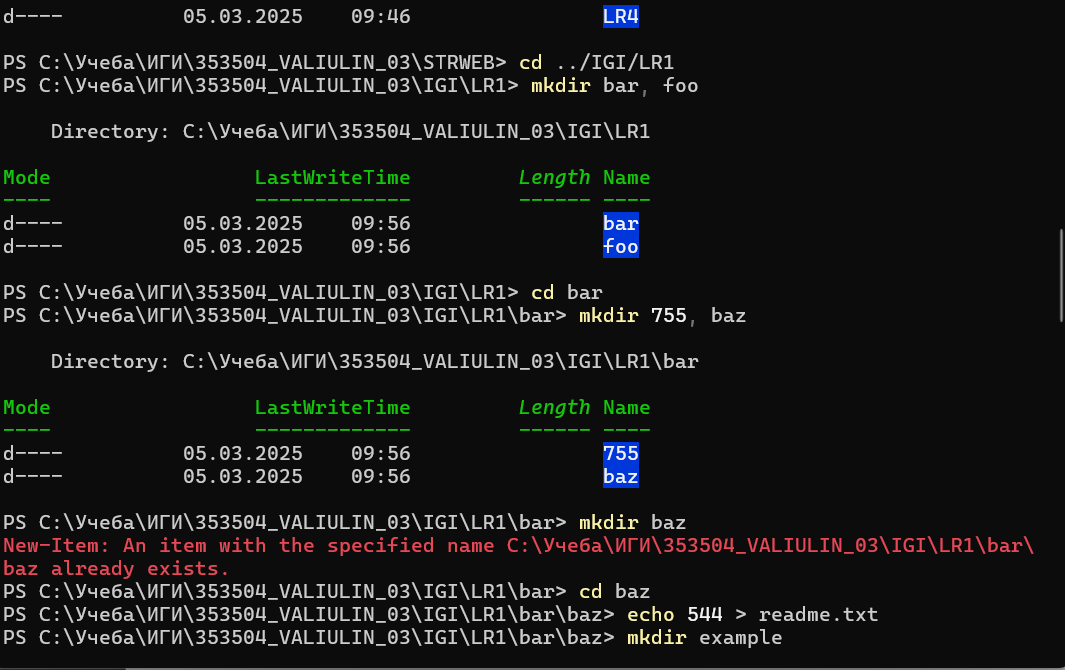
# 2 Выполнение работы

Была создана структура папок:



Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.



Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

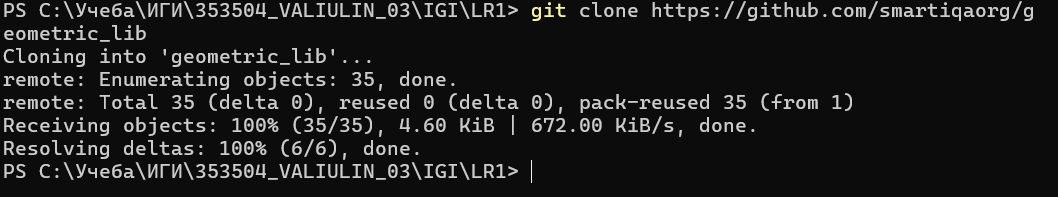
Изображение выглядит как текст, снимок экрана, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

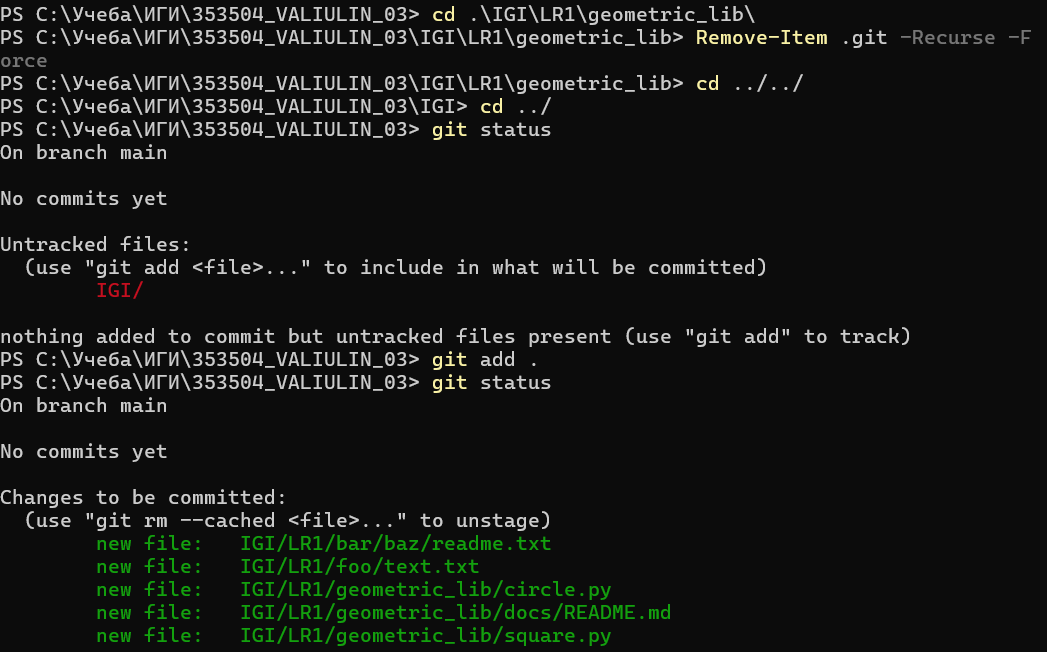
Итоговый результат создания директорий:  
  
Изображение выглядит как текст, снимок экрана, диаграмма

Контент, сгенерированный ИИ, может содержать ошибки.

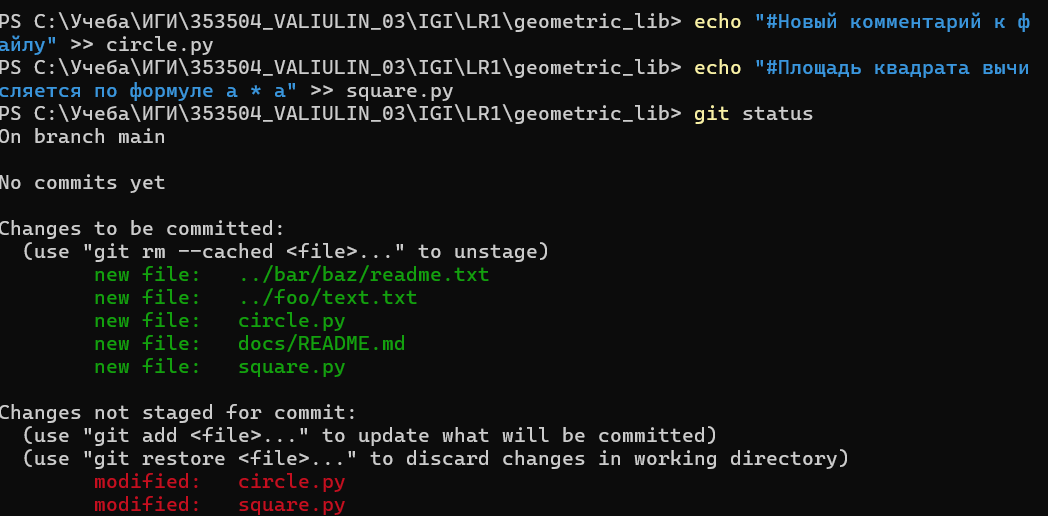
Был склонирован git репозиторий



Начинаем выполнение пункта 4:

А) Инициализируем репозиторий, удаляем папку .git в geometric\_lib, для того чтобы объединить все файлы в рамках одного репозитория 353504\_VALIULIN\_03  
  


В файлы были внесены изменения, после этого был осуществлен коммит.



Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Добавление и удаление файла:

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Добавление и удаление группы файлов:

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Переименование файлов  
Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.  
И коммит

Изображение выглядит как текст, снимок экрана, Шрифт

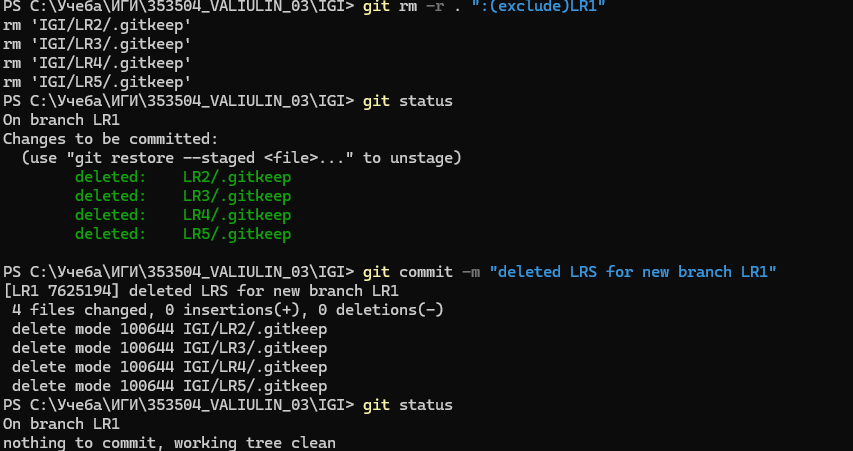
Контент, сгенерированный ИИ, может содержать ошибки.

Были созданы необходимые ветки:

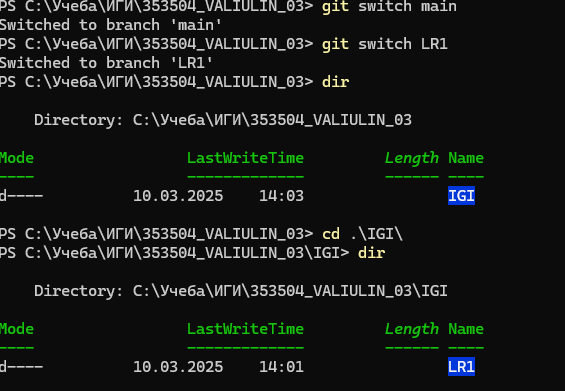
Изображение выглядит как текст, снимок экрана, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

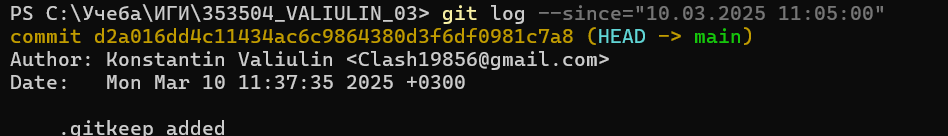
Каждой ветки были выделены свои папки:



Итоговые ветки:  
Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.  
И состояние в них, для примера LR1  


Начинаю разбираться с git log, применяются флаги –since, before, author, committer, grep



Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

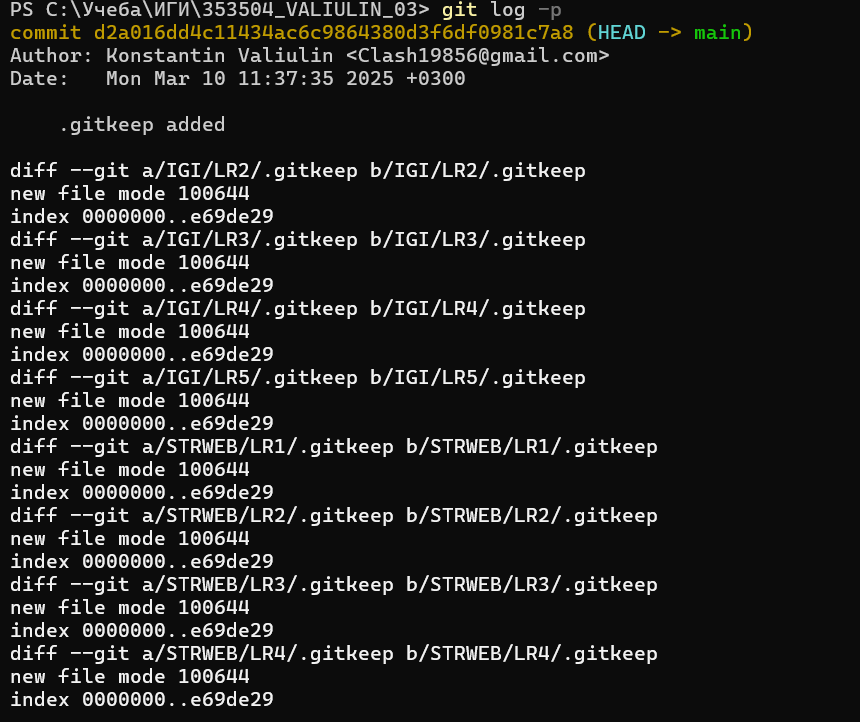
Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

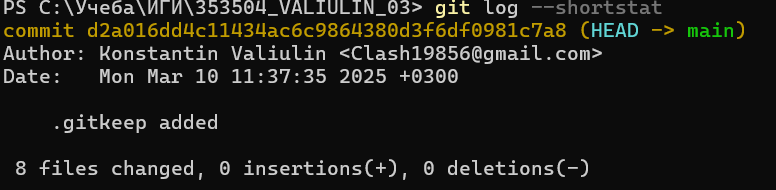
Переходим к пункту H)  
Применяются флаги:

-p (для того, чтобы увидеть diff для каждого коммита)  
--stat (статистика изменений в каждом коммите)  


Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

--shortstat для короткой статистики



--name-only для вывода только имен измененных файлов  
Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

--relative-date для вывода коммитов относительно текущей даты  
Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

--pretty для кастомного вывода коммитов  
  
Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

--graph для графического вывода (сейчас нет)  
Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Разница между конкретными коммитами

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Используем git reset –soft HEAD^1 для удаления коммита с оставлением изменений в индексе  
Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Убрали индексацию с файла:  
  
Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Изменяем файл, а затем восстанавливаем его из последнего коммита.  
  
Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

На гитхабе был создан удаленный репозиторий, который затем был связан с локальным с помощью команды git remote add origin <url>. После этого локальная ветка main была связана с веткой origin/main  
  


С помощью команды git remote show origin просматриваем основную информацию о репозитории.

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

С помощью git fetch скачиваем все изменения из удаленного репозитория.

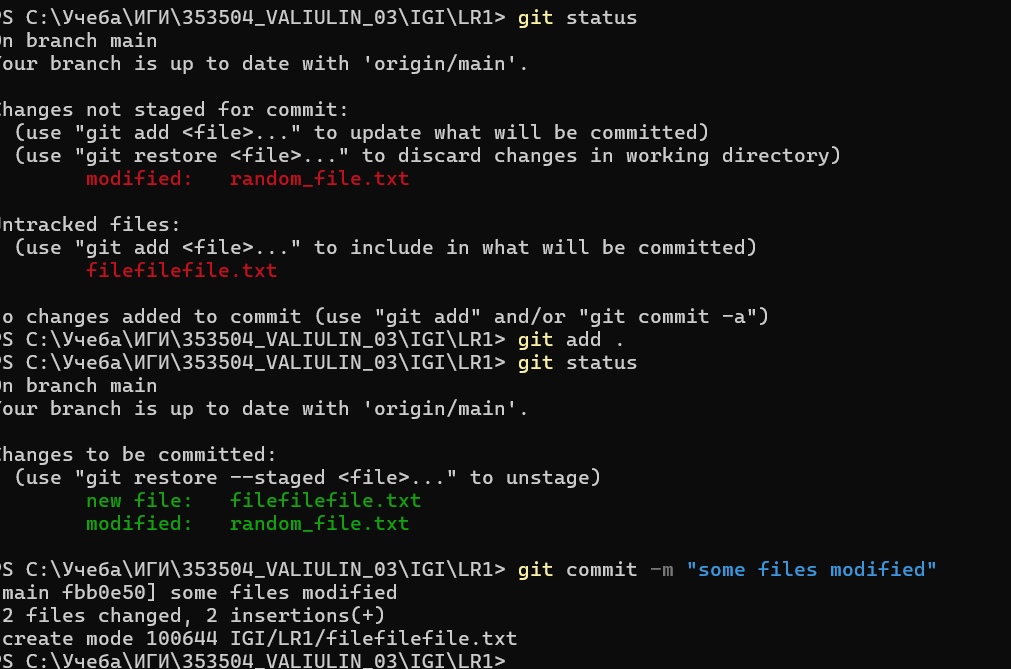
Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

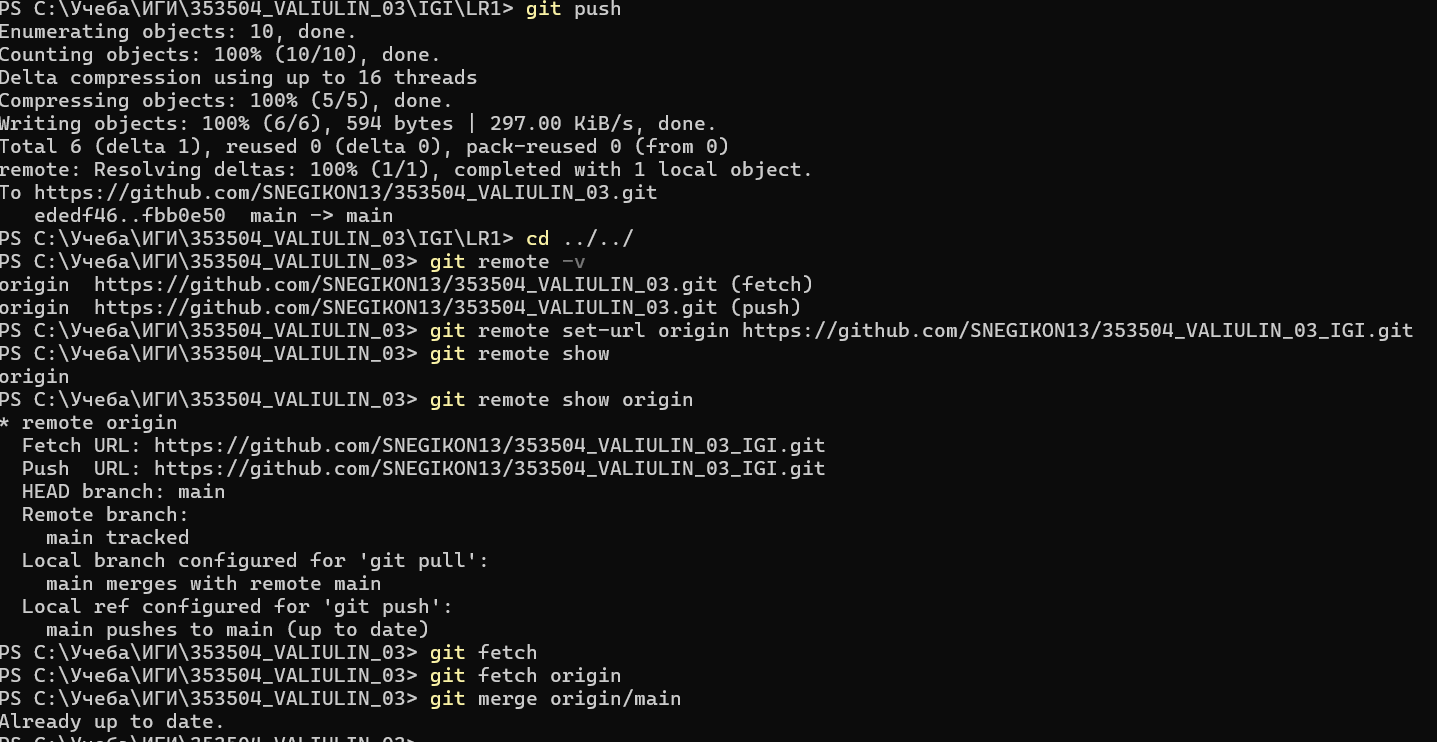
И затем мержим их с локальной веткой main  
  
Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

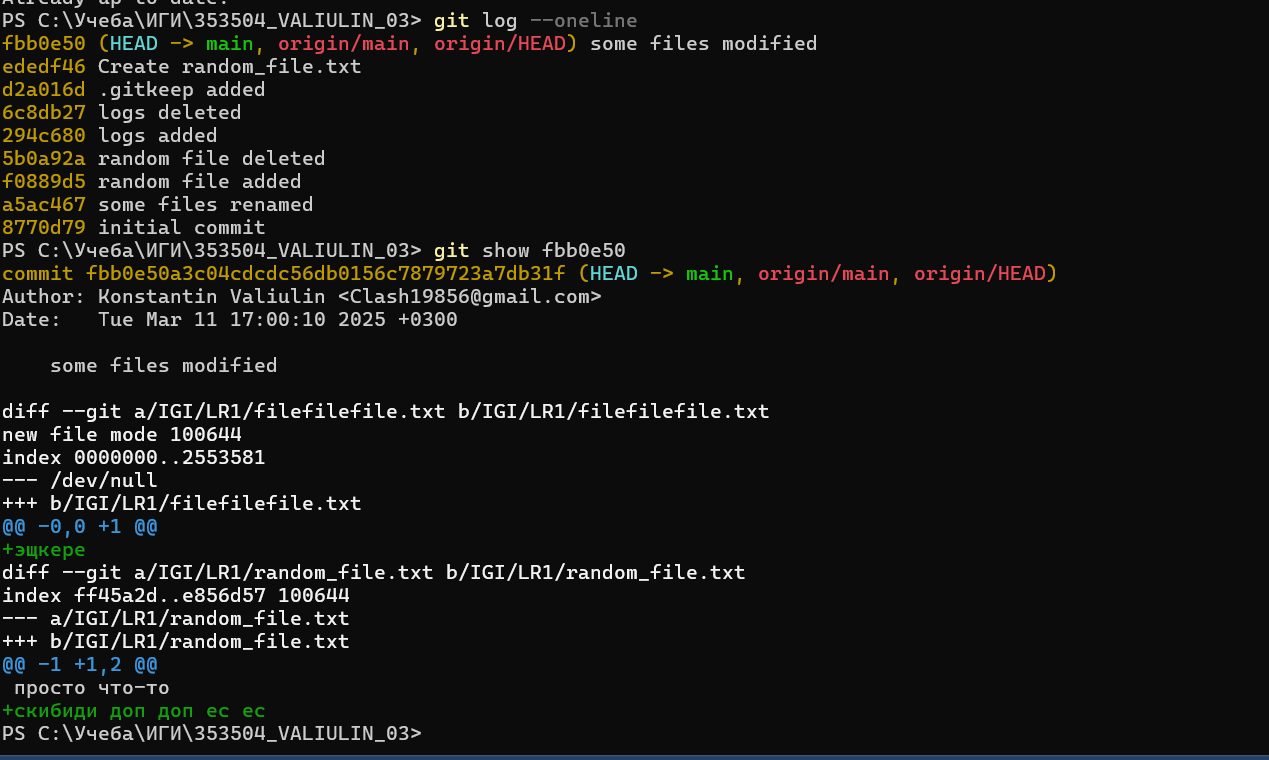
Был изменен существующий файл, создан новый и сделан коммит



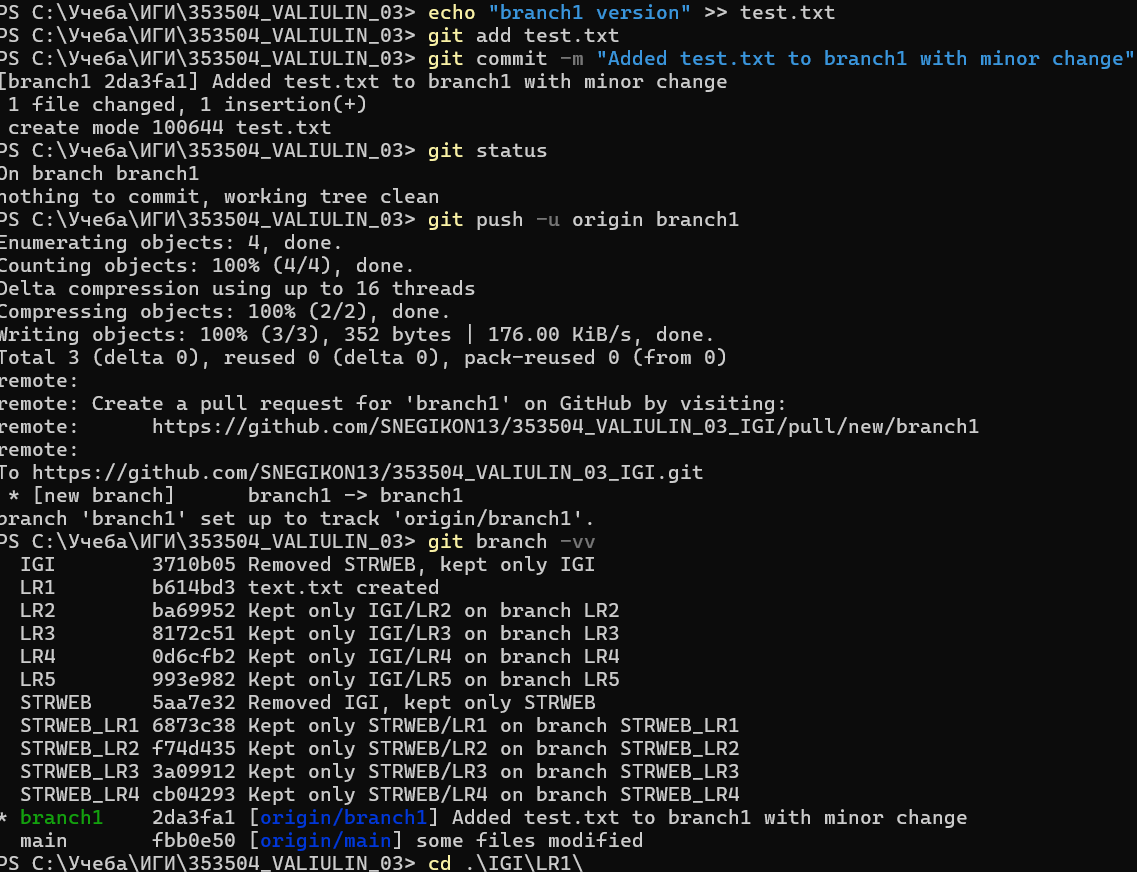
Мы запушили локальные изменения, затем изменили на гитхабе название репозитория и сменили ссылку на него в гите



Посмотрели историю коммитов и один отдельный коммит



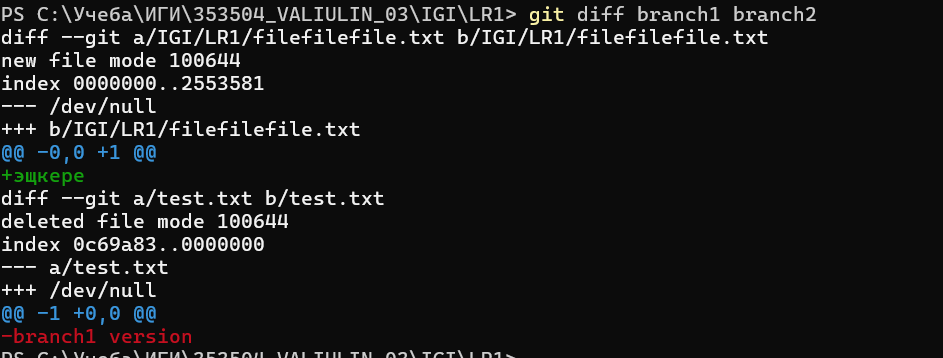
Создаем ветку branch1, создаем текстовый файл, делаем коммит, загружаем ветку в удаленный репозиторий, удаляем файл.



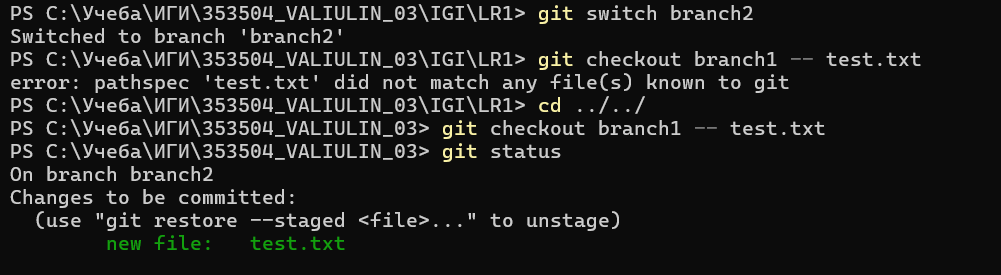
Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

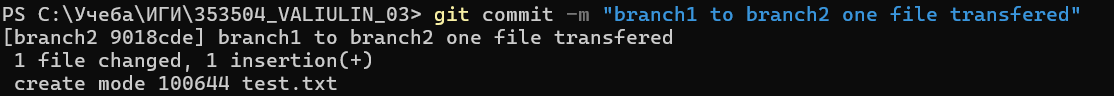
Создаем ветку branch2 из main, смотрим разницу между branch1 и branch2.



Перекидываем файл из branch1 в branch2



Делаем коммит

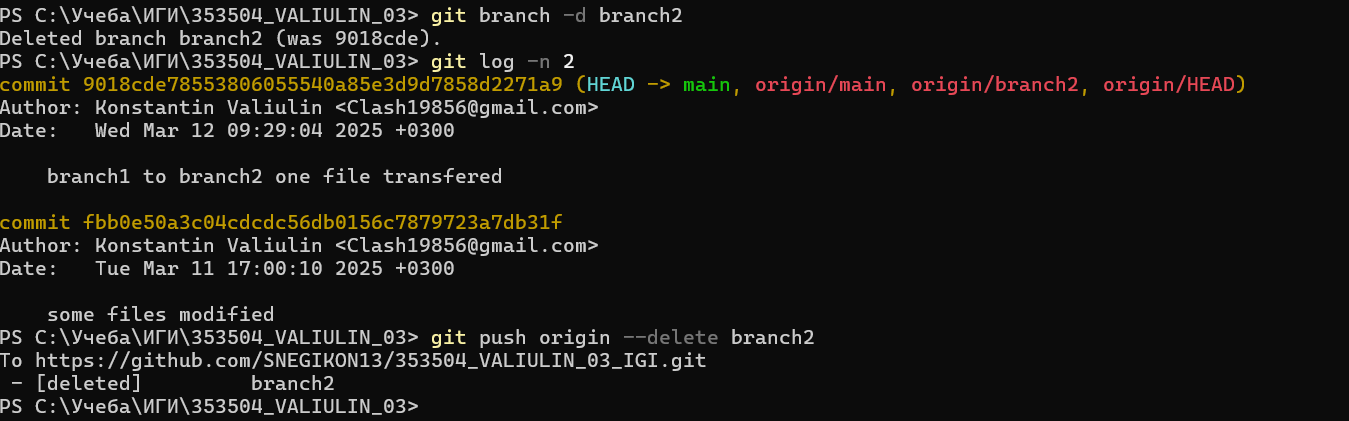


Мержим

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Удаляем ветку локально и на гитхабе



# Вывод

В рамках данной лабораторной работы был выполнен полный цикл работы с системой контроля версий Git, включая управление локальным и удалённым репозиториями. Работа проводилась в директории C:\Учеба\ИГИ\353504\_VALIULIN\_03 с использованием GitHub.

Сначала были пересозданы ветки IGI (с подветками LR1–LR5) и STRWEB (с подветками STRWEB\_LR1–STRWEB\_LR4) из ветки main с корректной структурой файлов, используя команды git checkout -b, git rm -r с исключающими выражениями и git commit. Затем изучены команды просмотра истории коммитов (git log) с фильтрами (-n, --since, --author, --grep) и флагами форматирования (--stat, -p, --graph). Создан файл test.txt, закоммичен и отменён (git reset --soft), проиндексирован и удалён из индекса (git restore --staged), изменён и восстановлен (git restore).

Далее создан удалённый репозиторий на GitHub (https://github.com/SNEGIKON13/353504\_VALIULIN\_03\_IGI.git), связан с локальным (git remote add origin), и выполнен первый push (git push -u origin main). Репозиторий переименован, данные получены (git fetch, git merge), история просмотрена (git log --oneline, git show). Создана ветка branch1, перенесены файлы test.txt и remote.txt из main, удалён remote.txt, изменения отправлены на GitHub (git push -u origin branch1). Затем создана ветка branch2 из main, перенесены test.txt и IGI/LR1/random\_file.txt из branch1 с помощью git checkout branch1 --, слиты в main (git merge) и удалены локально и удалённо (git branch -d, git push origin --delete).

В процессе работы освоены ключевые команды Git для управления ветками, файлами и удалёнными репозиториями, а также решены конфликты и ошибки (например, с путями и индексацией). Лабораторная работа успешно завершена, демонстрируя практические навыки работы с Git и GitHub.