



Министерство науки и высшего образования Российской Федерации
Калужский филиал
федерального государственного бюджетного
образовательного учреждения высшего образования
*«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)*

ФАКУЛЬТЕТ **ИУК «Информатика и управление»**

КАФЕДРА **ИУК4 «Программная инженерия»**

Лабораторная работа №3

«Работа с системой управления версиями»

ДИСЦИПЛИНА: «Современные проблемы программной инженерии»

Выполнил: студент гр. ИУК4-21М _____ (Сафронов Н.С.)
(подпись) (Ф.И.О.)

Проверил: _____ (Кручинин И.И.)
(подпись) (Ф.И.О.)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:

- Оценка:

Калуга, 2025

Цель работы: изучение базовых возможностей системы управления версиями. Рассмотреть базовые команды для работы с репозиторием.

Задание на лабораторную работу

1. Если система контроля версий Git не установлена, то установить ее (параметры оставить по умолчанию).
2. Запустить Git GUI или TortoiseGit (или консоль). Создать новый репозиторий (в папке по фамилии студента).
3. Добавить в папку репозитория файлы. Зафиксировать состояние репозитория (выполнить commit).
4. Внести изменения в файлы. Зафиксировать новое состояние репозитория.
5. Создать новую ветку 1. Внести в нее изменения (добавить новый файл и изменить существующий файл: добавить, удалить и изменить строки) и зафиксировать их.
6. Переключиться на ветку мастера. Внести в нее изменения (добавить новый файл; изменить существующие файлы: добавить, удалить и изменить строки первоначального файла) и зафиксировать их.
7. Продемонстрировать слияние веток. Разрешить возникший конфликт.
8. Просмотреть дерево изменений веток (историю).
9. Продемонстрировать откат изменений в ветке 1.
10. Создать удаленный репозиторий (на github.com или в сети).
11. Отправить данные на удаленный репозиторий (выполняется одним из студентов подгруппы). Добавить к удаленному репозиторию участников проекта.
12. Получить данные из удаленного репозитория (выполняется прочими студентами).
13. Изменить полученные данные.
14. Зафиксировать изменения и отправить их на удаленный репозиторий (выполняется всеми студентами подгруппы).

15. Получить данные из удаленного репозитория.
16. Просмотреть историю изменений.

Дополнительно:

1. Продемонстрировать работу revert и reset.
2. Продемонстрировать сохранение изменений в stash с последующим восстановлением.
3. Продемонстрировать создание и применение серии патчей.
4. Продемонстрировать создание и применение тегов.
5. Продемонстрировать rebase.

Результаты выполнения работы

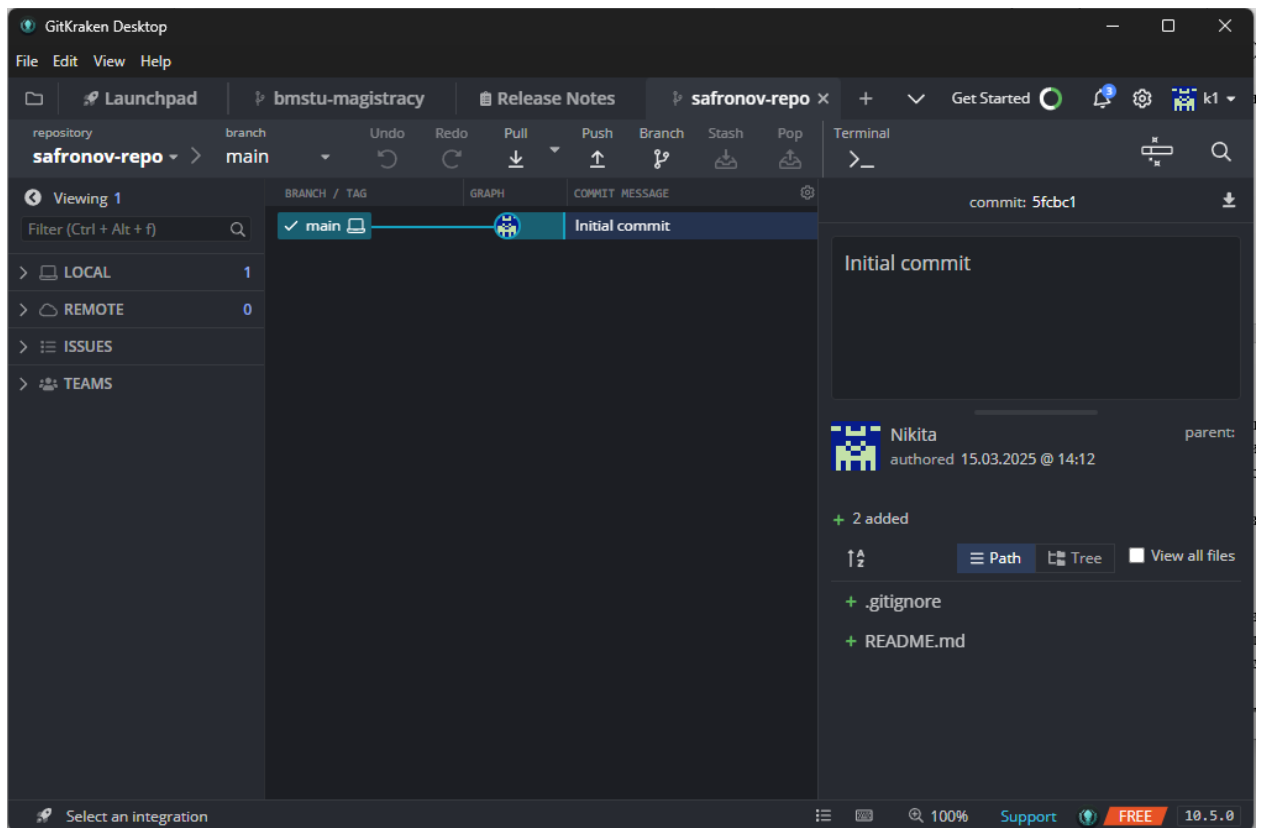


Рисунок 1 – Созданный репозиторий

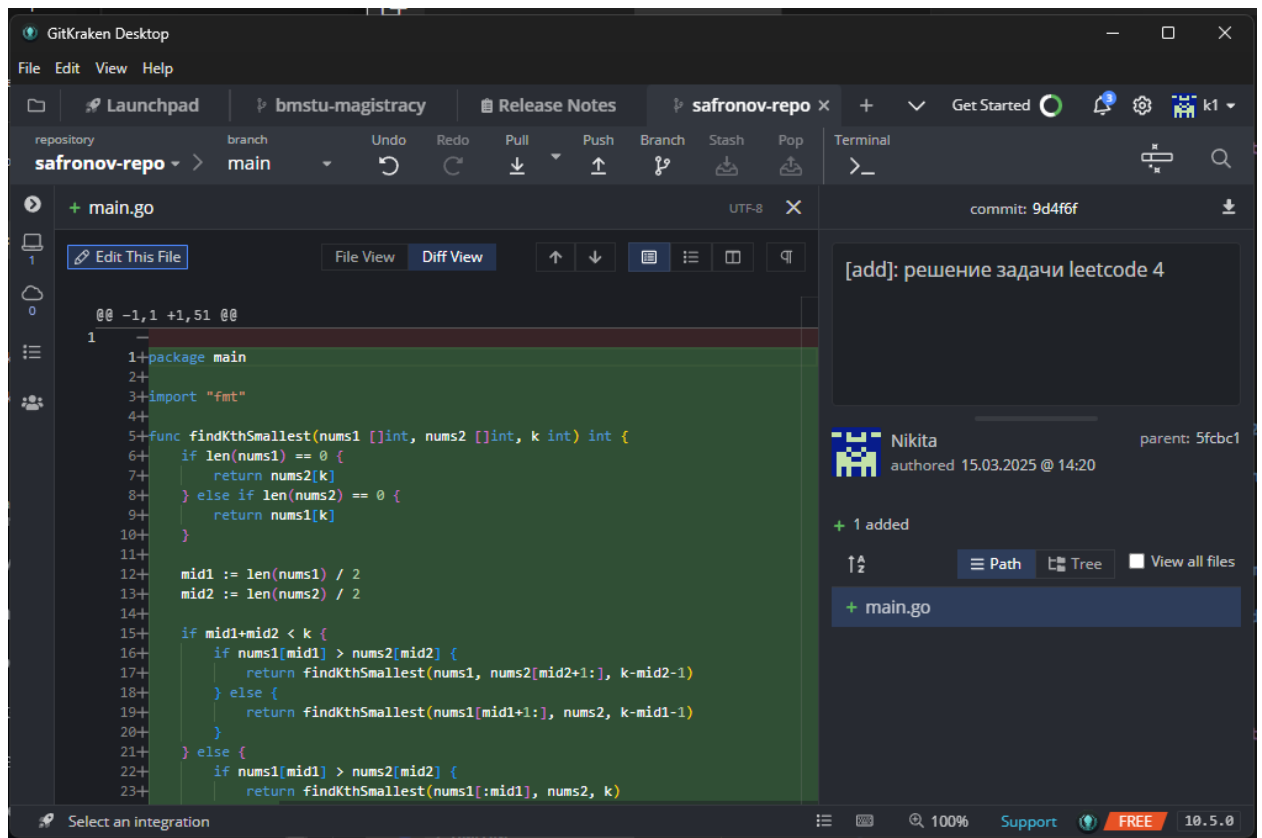


Рисунок 2 – Первый коммит в репозитории

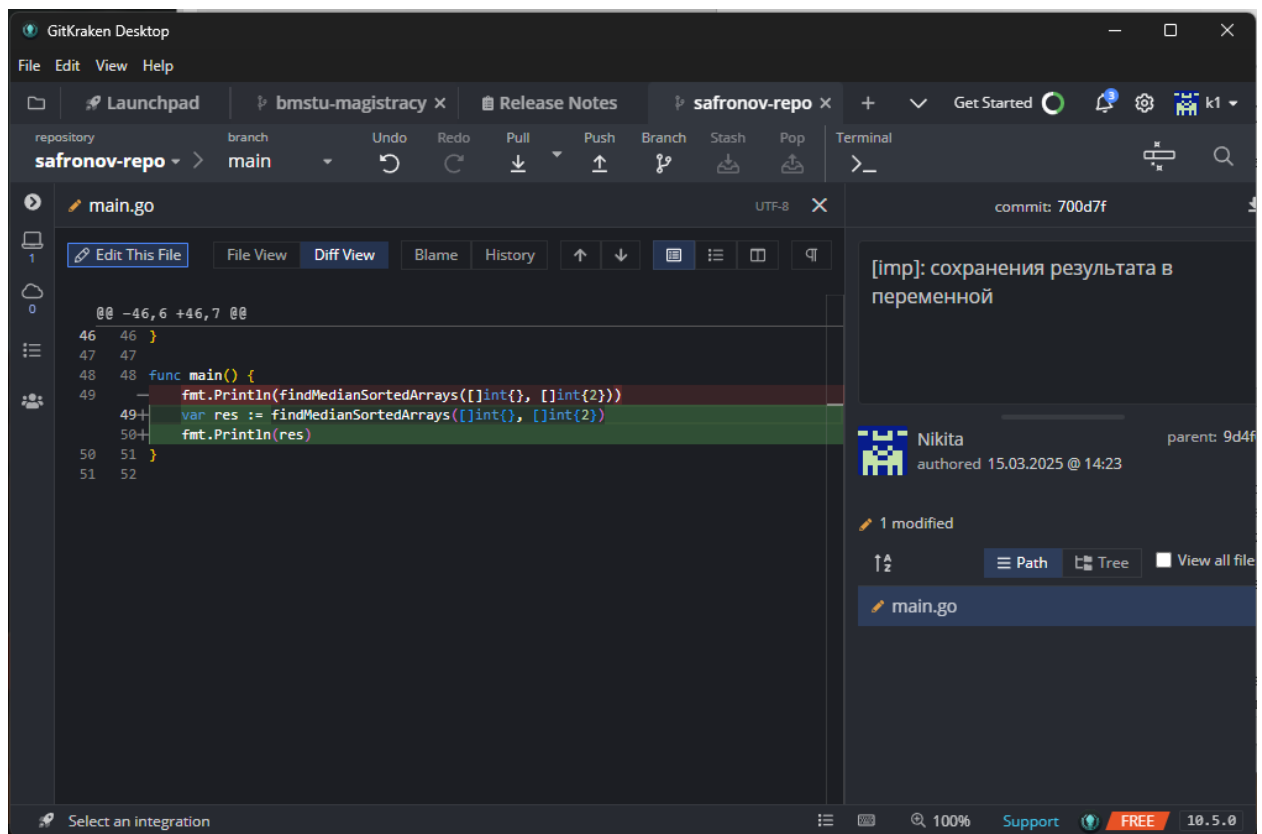


Рисунок 3 – Внесение изменений в файл

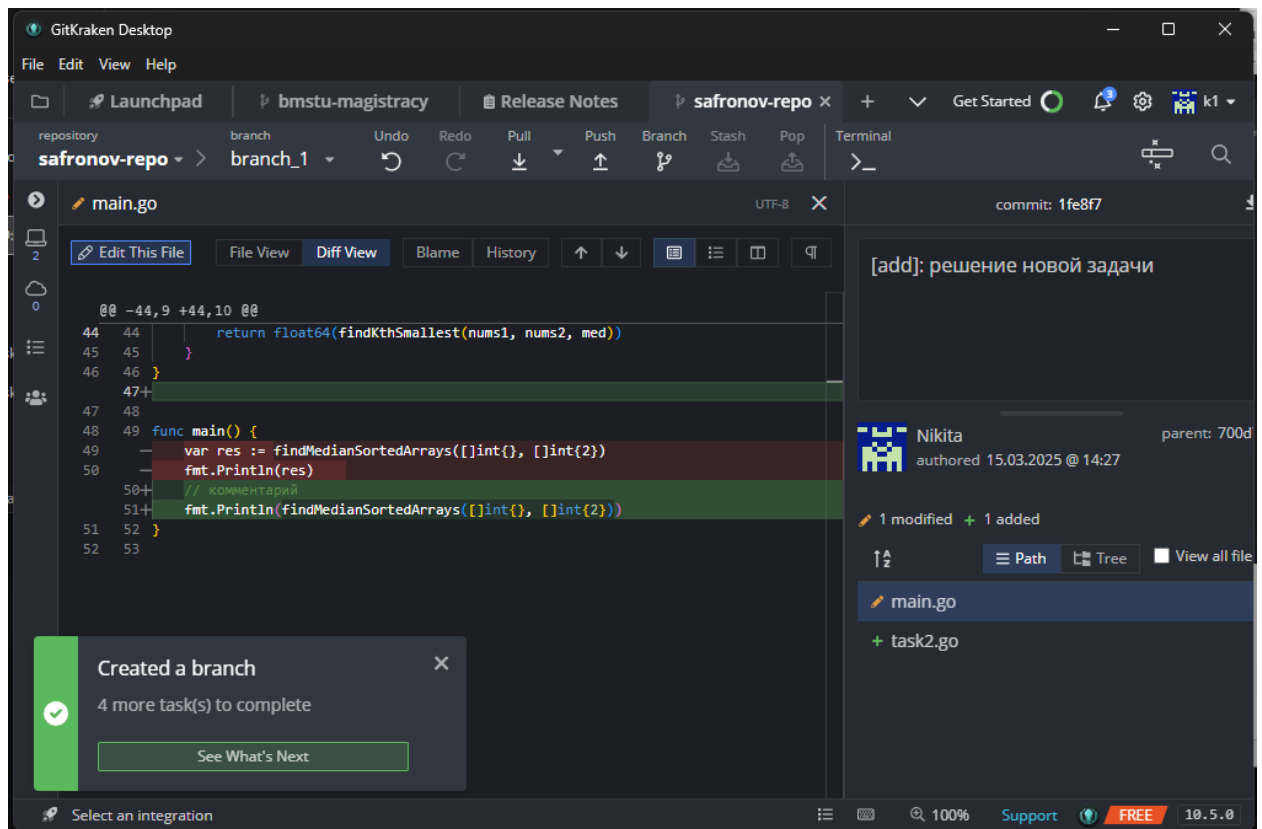


Рисунок 4 – Созданная новая ветка и коммит в ней

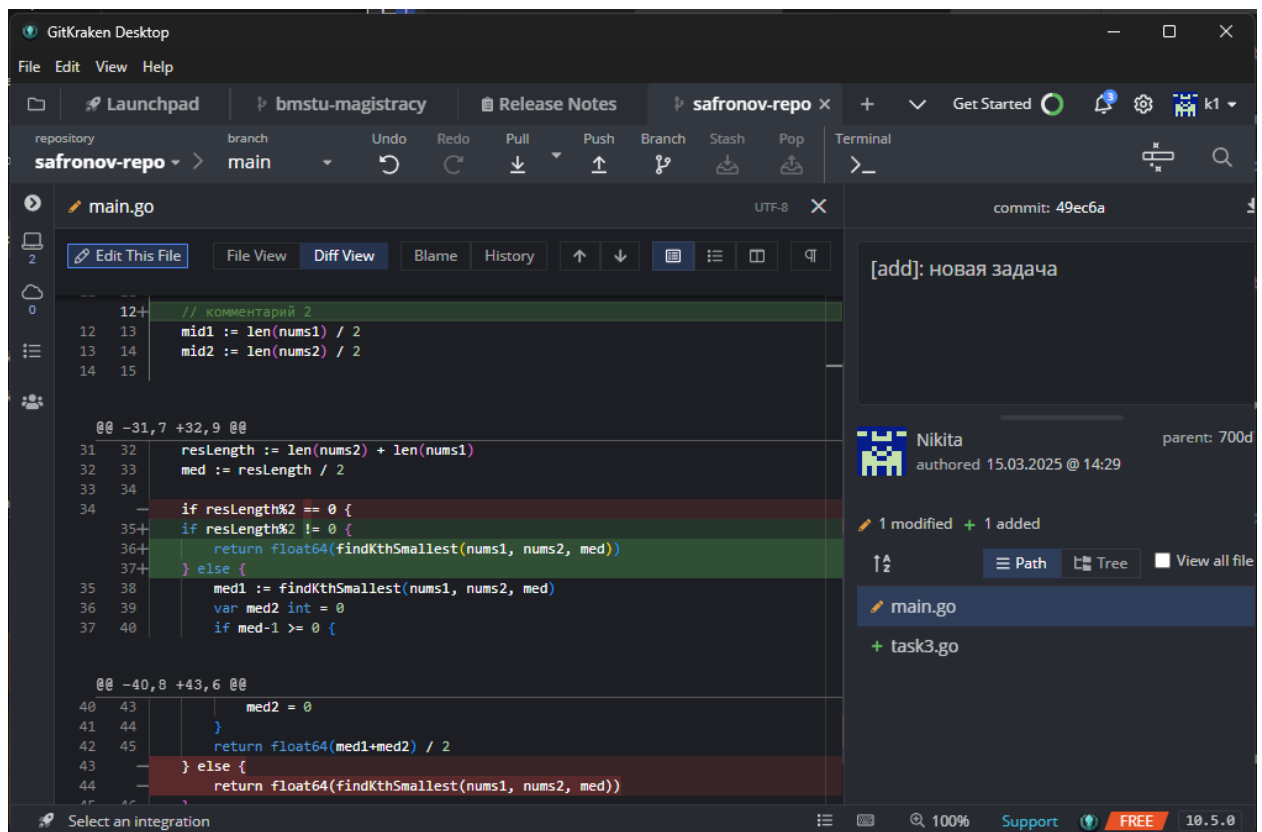


Рисунок 5 – Изменения в master-ветке

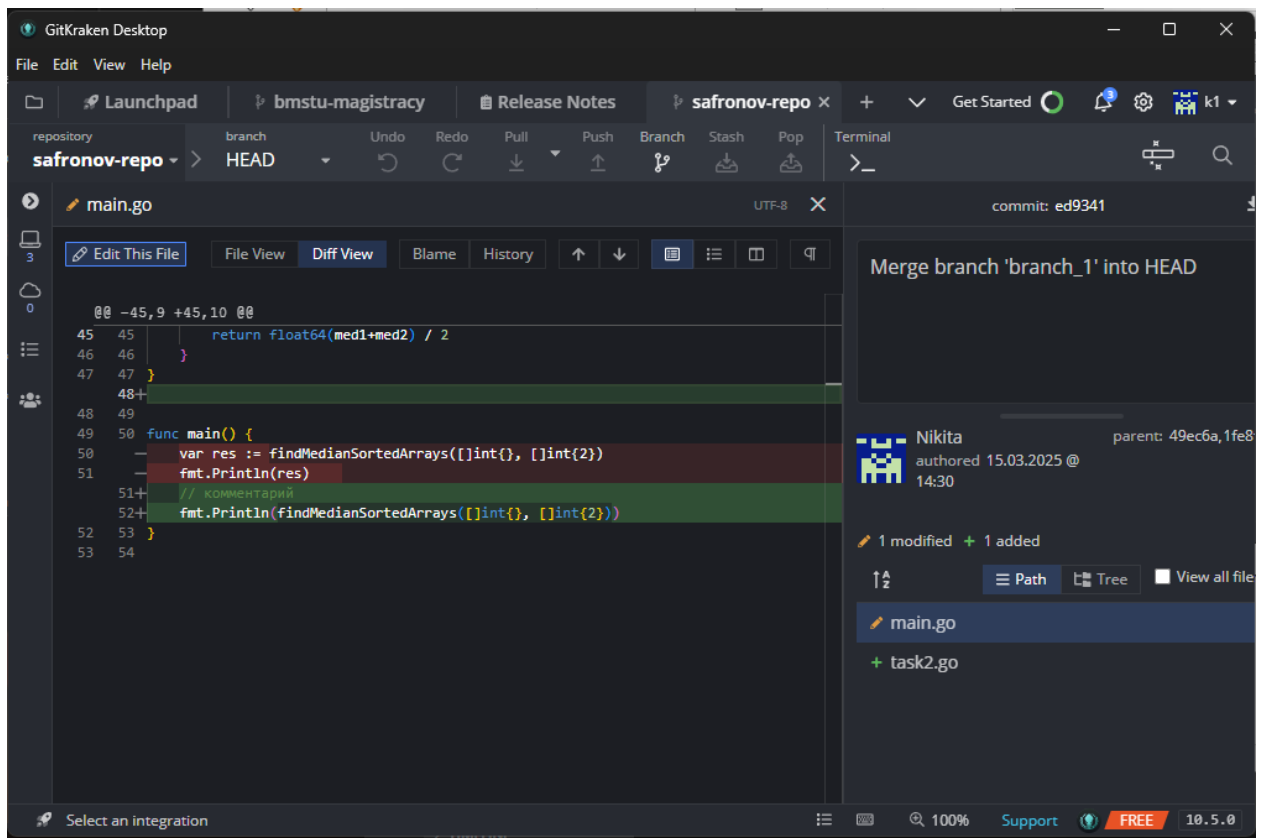


Рисунок 6 – Слияние ветки branch_1 в master-ветку

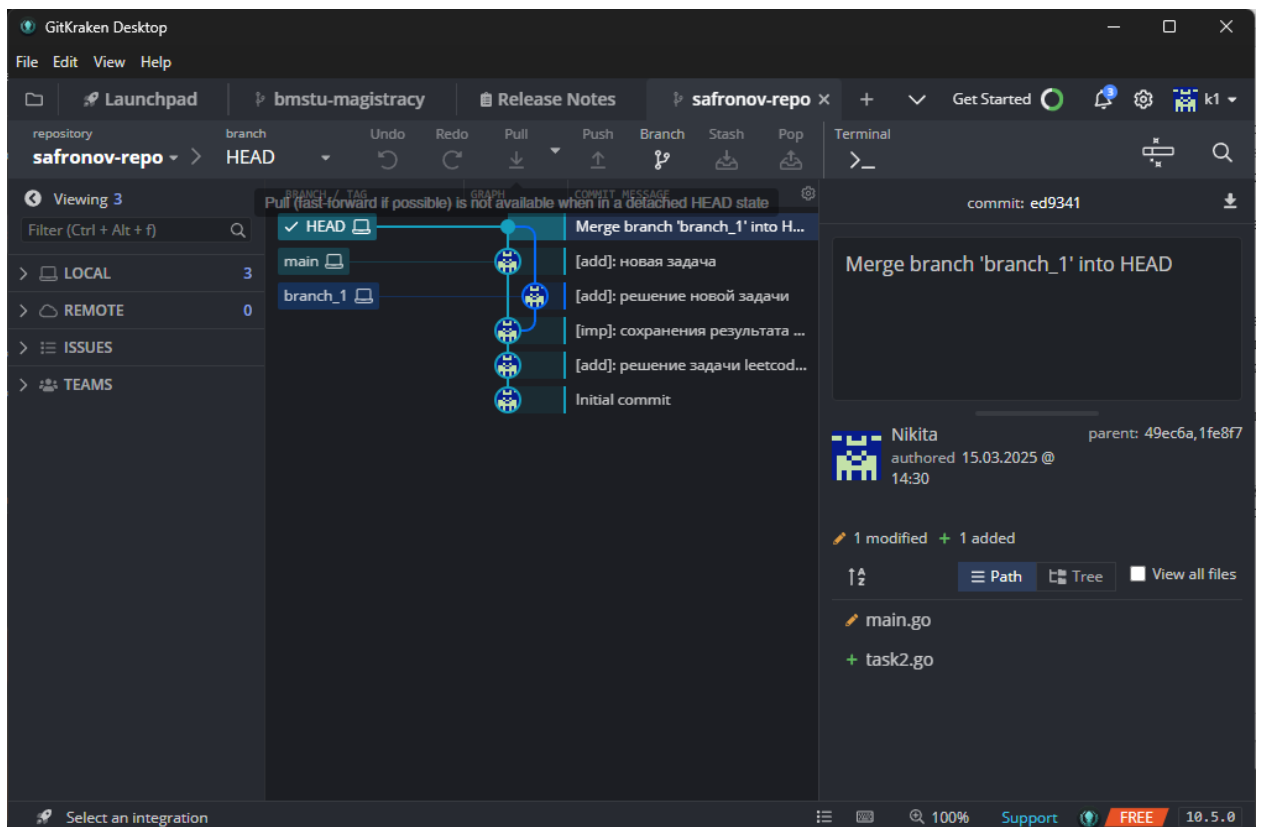


Рисунок 7 – Дерево изменений репозитория

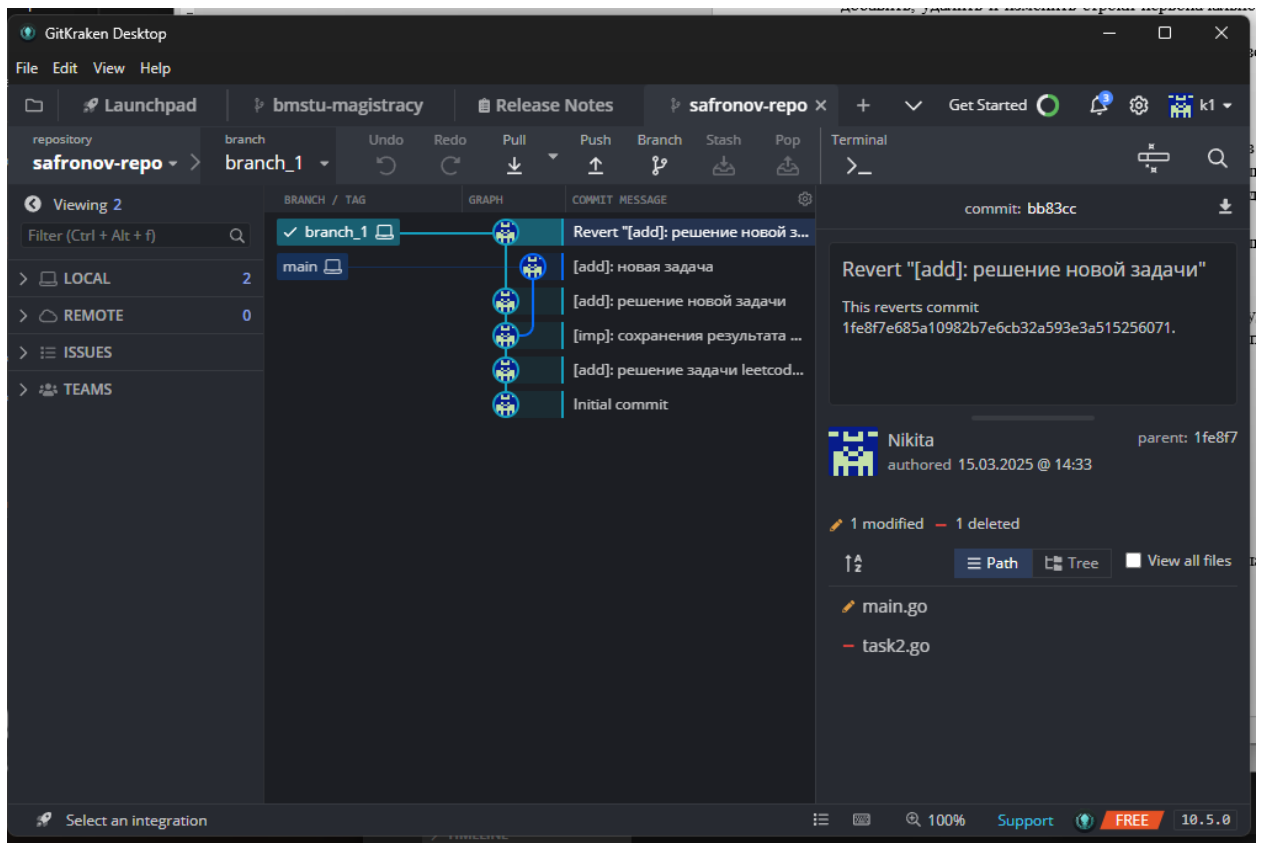


Рисунок 8 – Revert в ветке branch_1

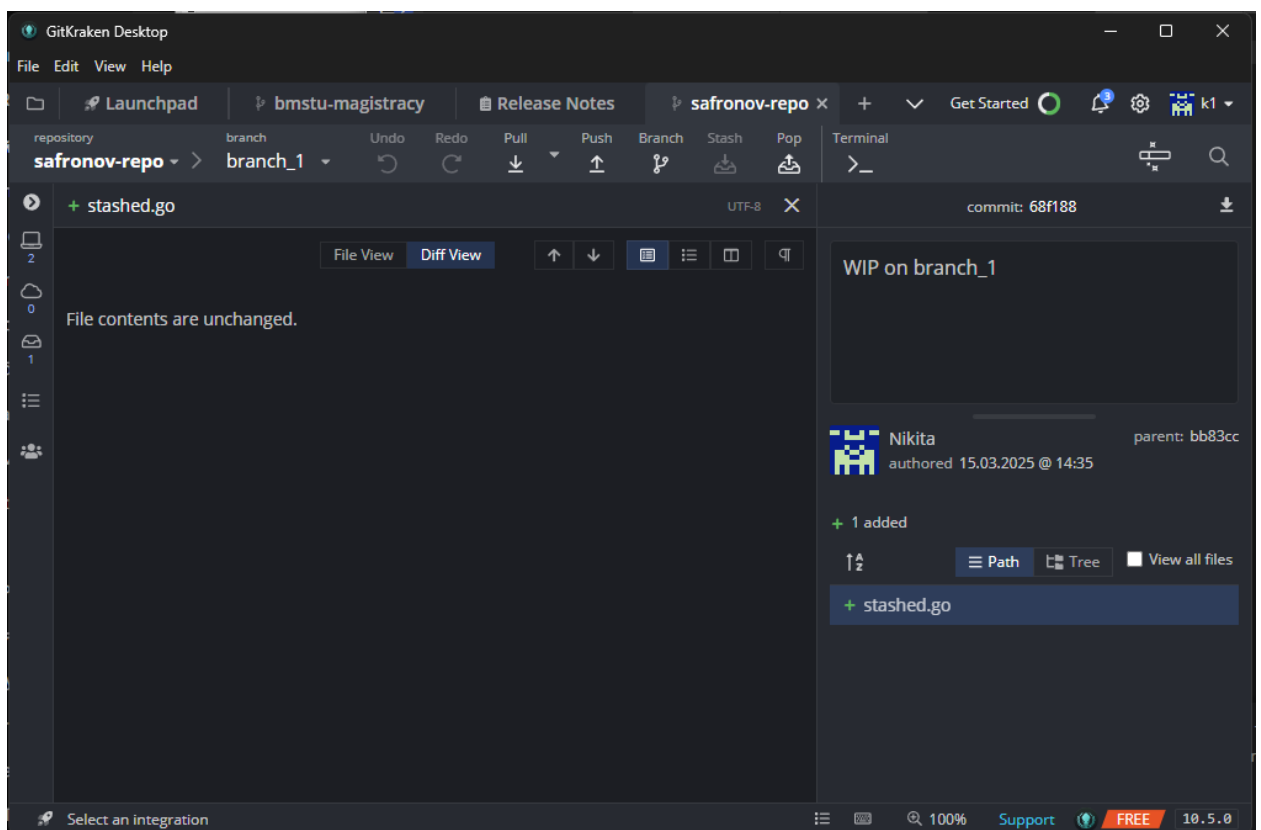


Рисунок 9 – Созданный stash

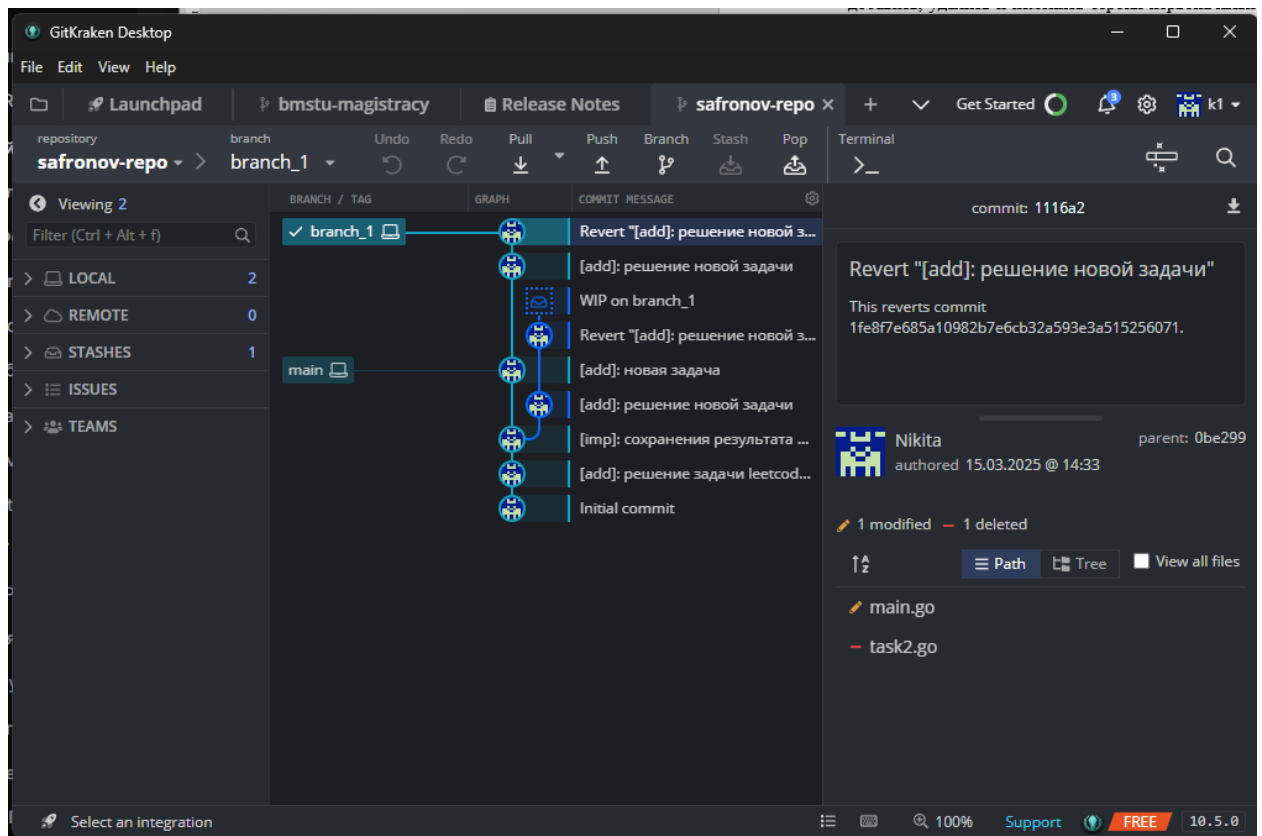


Рисунок 10 – Результат rebase ветки branch_1 на master

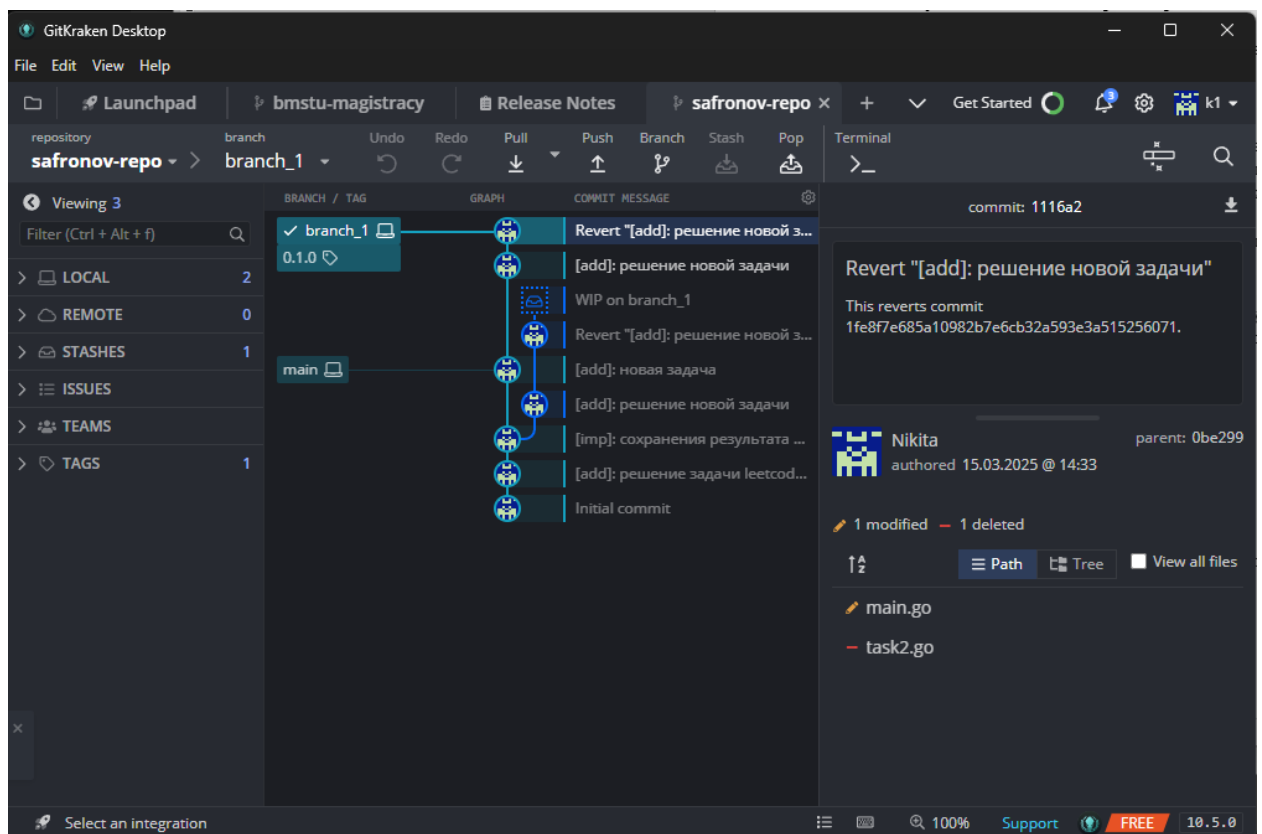


Рисунок 11 – Создание тэга 0.1.0 в ветке branch_1

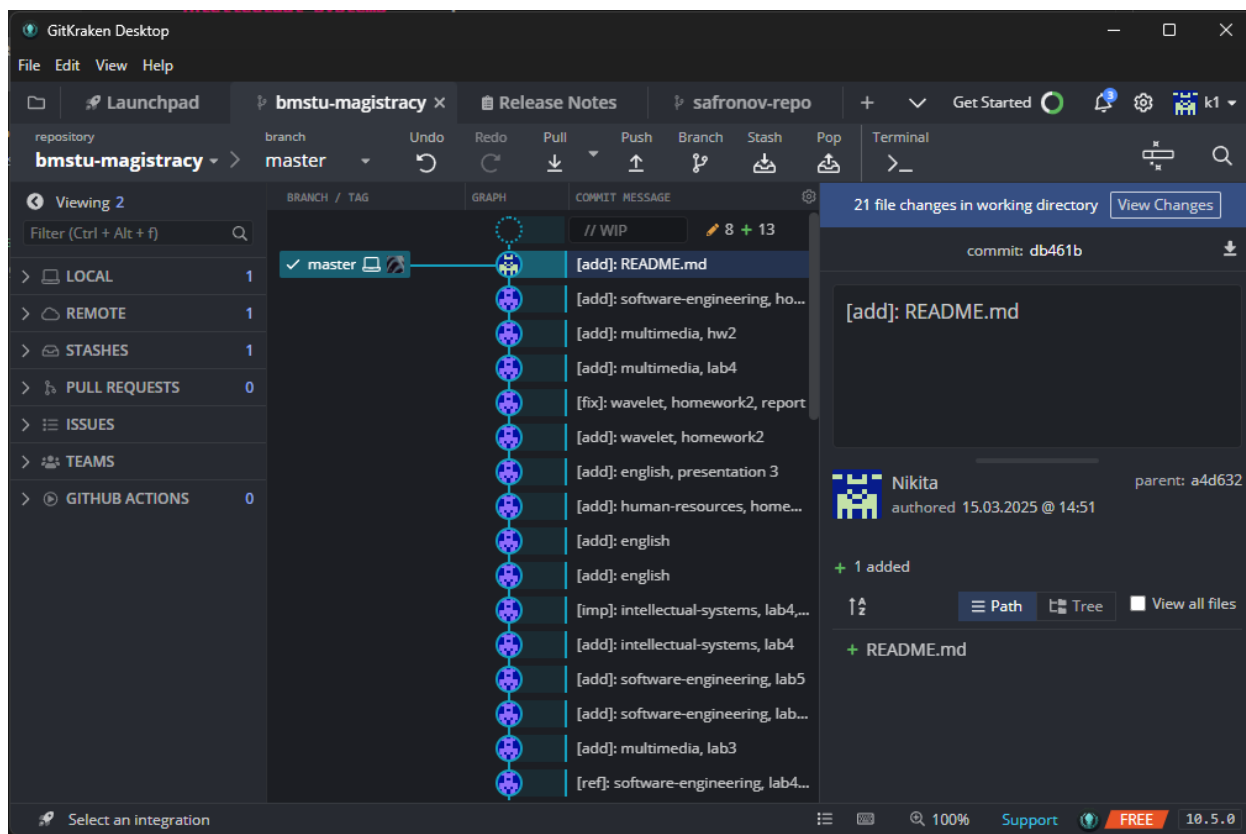


Рисунок 12 – Отправка изменений в удалённый репозиторий и история изменений в нём

Выводы: изучены базовые возможности системы управления версиями (на примере Git) и освоены ключевые команды для работы с репозиторием. В процессе выполнения лабораторной работы были рассмотрены основные операции, такие как создание репозитория, фиксация изменений (commit), ветвление (branch), слияние (merge), разрешение конфликтов, а также работа с удалёнными репозиториями (push, pull).