Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития

Кафедра инфокоммуникаций

**ОТЧЕТ**

**ПО ПРАКТИЧЕСКОЙ РАБОТЕ №3**

**дисциплины «Программирование на Python»**

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Выполнил:  Лейс Алексей Вячеславович  2 курс, группа ИВТ-б-о-22-1,  09.03.01 «Информатика и вычислительная техника», направленность (профиль) «Программное обеспечение средств вычислительной техники и автоматизированных систем»  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | |
|  | | Руководитель практики: кандидат тех. наук доцент кафедры инфокоммуникаций: Воронкин Р.А  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | |
|  | |  | |

Отчет защищен с оценкой \_\_\_\_\_\_\_\_\_\_\_ Дата защиты\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ставрополь, 2022 г.

**Тема:** Основы ветвления Git

**Цель работы:** исследование базовых возможностей по работе с локальными и удаленными ветками Git.

Порядок выполнения работы:

Создаю три файла: 1.txt, 2.txt, 3.txt.

Проиндексировал первый файл и сделать коммит с комментарием "add 1.txt file".

Проиндексировал второй и третий файлы.

Перезаписал уже сделанный коммит с новым комментарием "add 2.txt and 3.txt."

Результат:

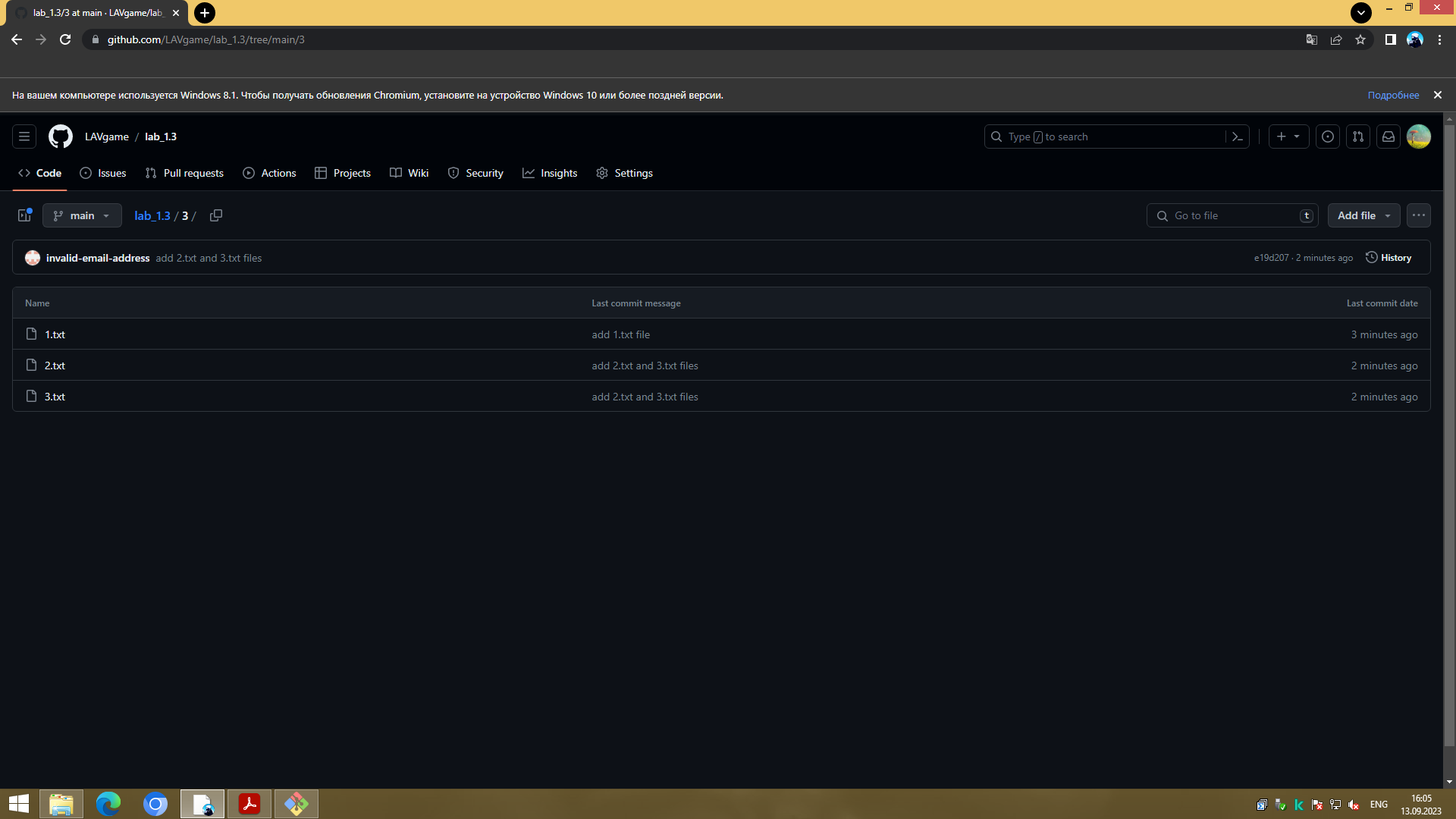


Рисунок 1 Созданные файлы

Создать новую ветку my\_first\_branch.

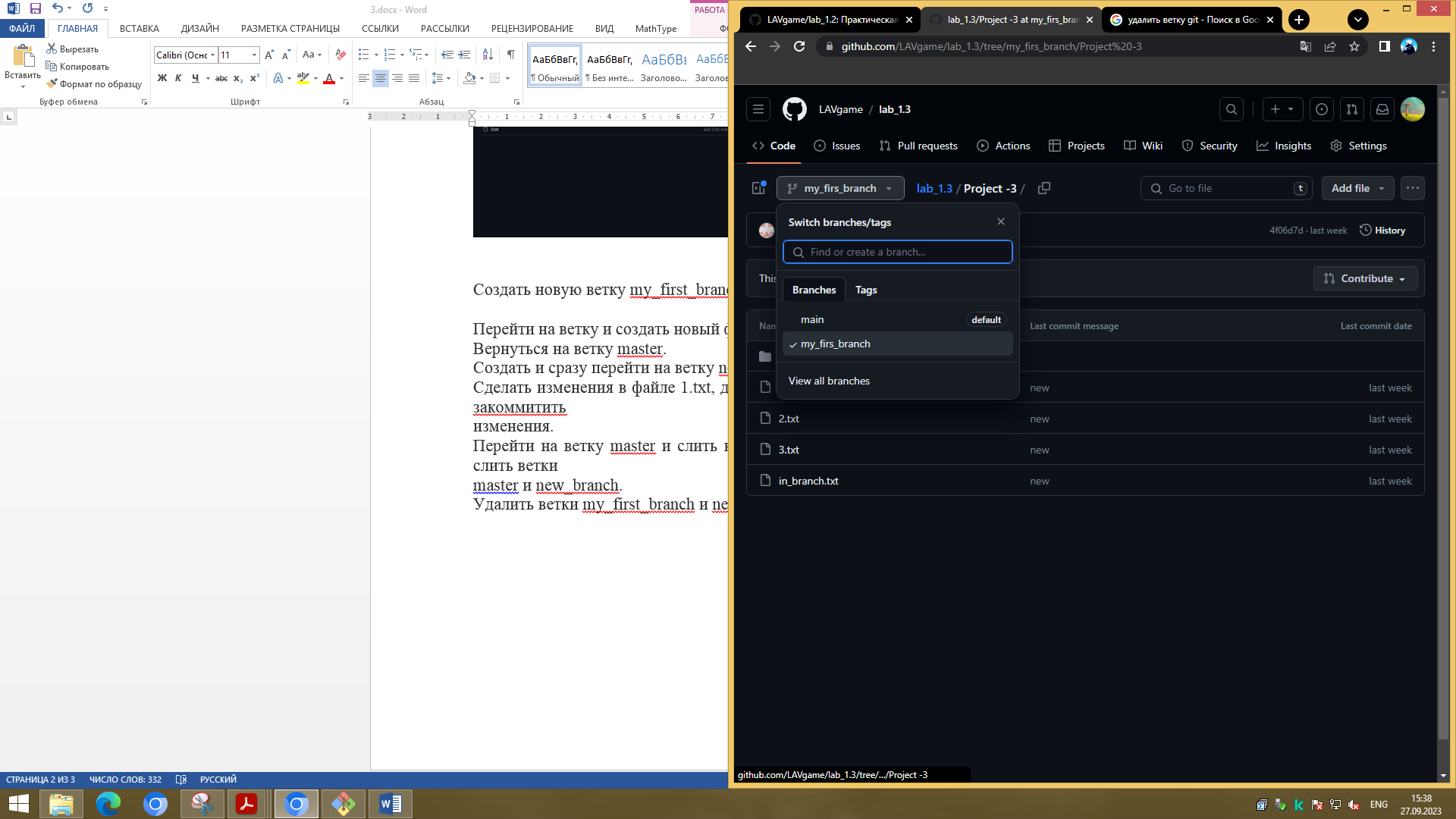


Рисунок 2 Созданная ветка

Создание на новой ветке файла.

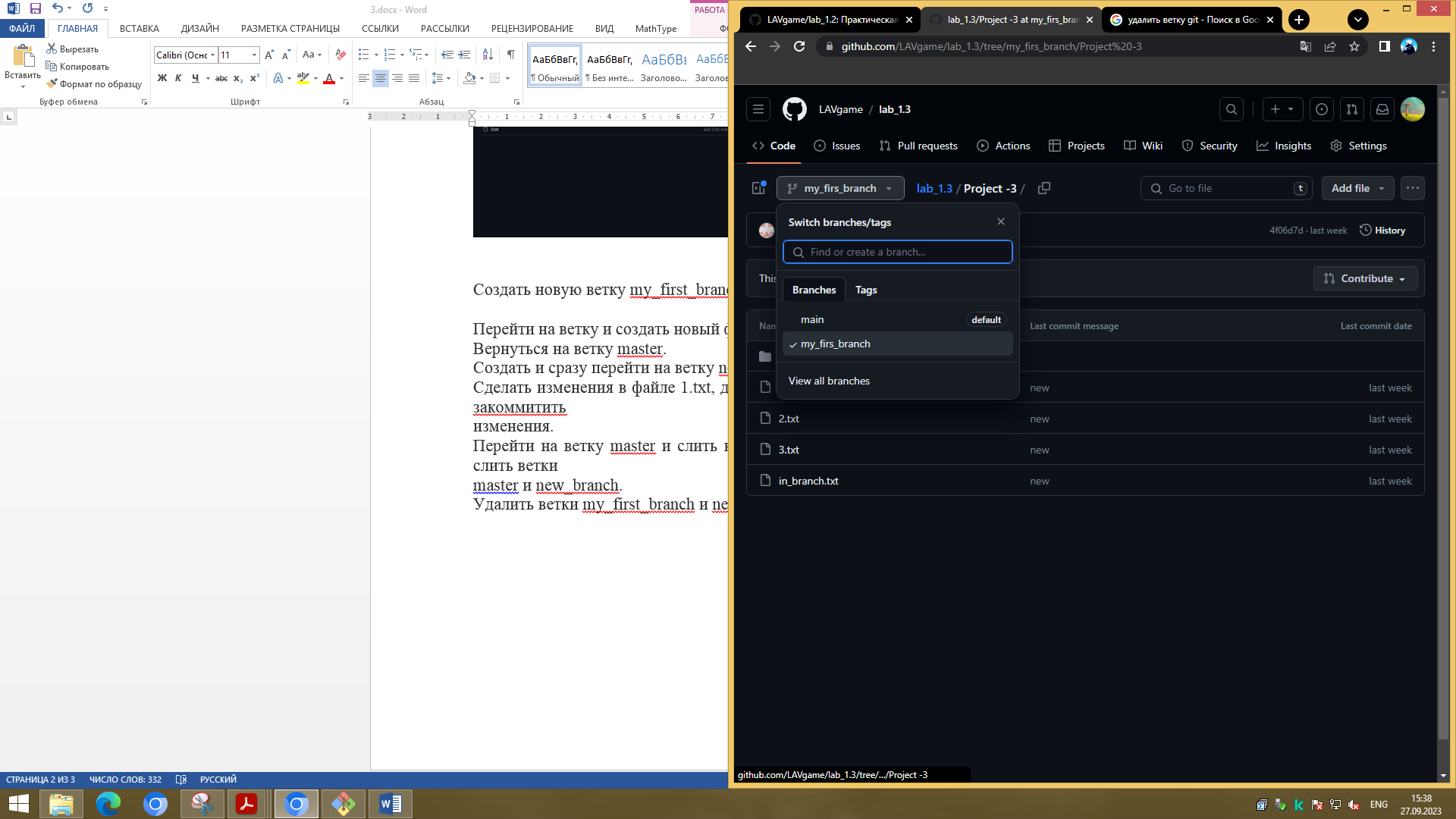


Рисунок 3 Файл на ветке

Удалить ветки my\_first\_branch и new\_branch.

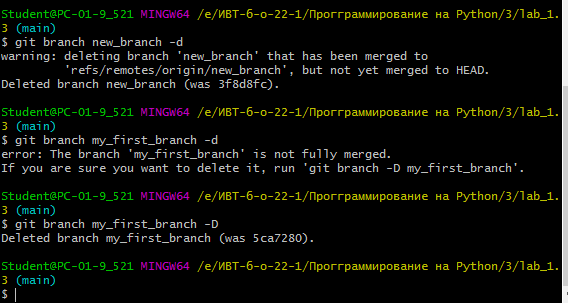


Рисунок 4 Удаление ветки

Создать ветки branch\_1 и branch\_2.

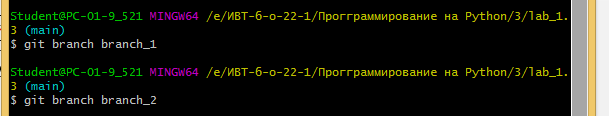


Рисунок 5 Создание веток

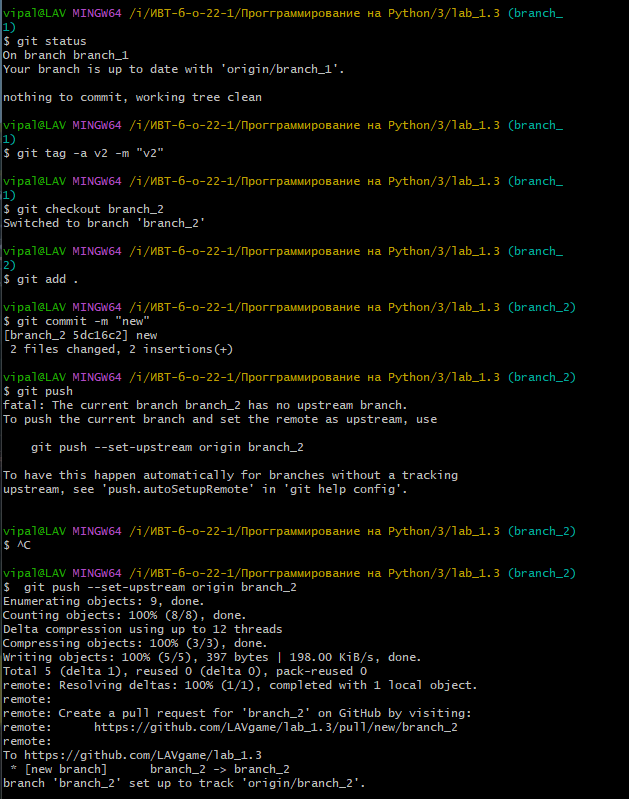


Рисунок 5 Переход из веток

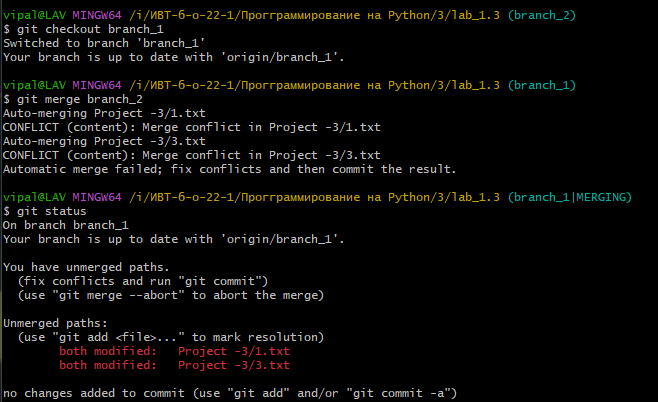


Рисунок 6 Слияние веток

Отправить ветку branch\_1 на GitHub.

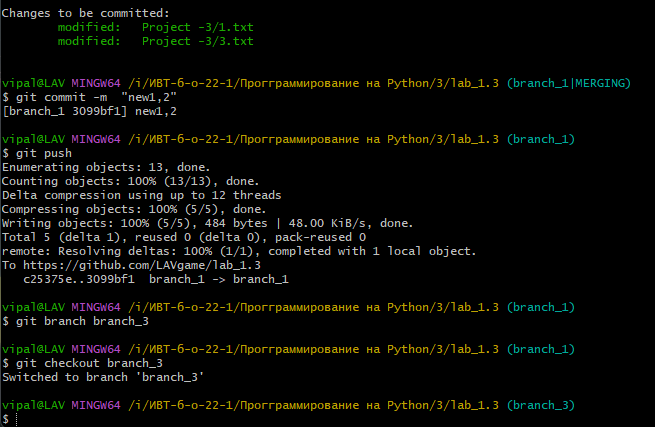


Рисунок 7 Готовые изменения и отправка

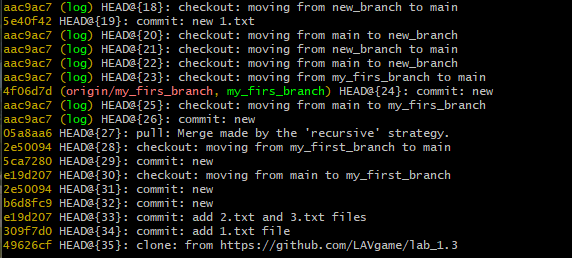


Рисунок 8 Создание 3 ветки и слияние Main с 2й веткой

**Ответы на вопросы:**

1. Что такое ветка?

Ветка в Git - это отдельная линия разработки, которая позволяет работать над изменениями независимо от других веток.

2. Что такое HEAD?

HEAD - это указатель на текущий коммит в вашем репозитории. Он указывает на последний коммит в текущей активной ветке.

3. Способы создания веток:

- `git branch <branch\_name>` - создать локальную ветку.

- `git checkout -b <branch\_name>` - создать и переключиться на новую локальную ветку.

- `git push origin <branch\_name>` - создать удаленную ветку.

4. Как узнать текущую ветку?

Используйте команду `git branch` или `git status` для определения текущей ветки.

5. Как переключаться между ветками?

Используйте команду `git checkout <branch\_name>` для переключения на другую ветку.

6. Что такое удаленная ветка?

Удаленная ветка - это ветка, которая существует на удаленном сервере Git, а не локально.

7. Что такое ветка отслеживания?

Ветка отслеживания (tracking branch) - это локальная ветка, связанная с удаленной веткой, что позволяет отслеживать изменения на удаленном сервере.

8. Как создать ветку отслеживания?

Используйте команду `git checkout --track origin/<branch\_name>` для создания локальной ветки, отслеживающей удаленную ветку.

9. Как отправить изменения из локальной ветки в удаленную ветку?

Используйте команду `git push origin <local\_branch>:<remote\_branch>`.

10. В чем отличие команд git fetch и git pull?

`git fetch` загружает изменения с удаленного репозитория, но не объединяет их с текущей веткой, в то время как `git pull` загружает и объединяет изменения.

11. Как удалить локальную и удаленную ветки?

- Для удаления локальной ветки: `git branch -d <branch\_name>` или `git branch -D <branch\_name>` (если ветка не слита).

- Для удаления удаленной ветки: `git push origin --delete <branch\_name>`.

12. Основные типы веток в модели git-flow:

Основные типы веток в модели git-flow включают "feature" (для новых функций), "release" (для подготовки релизов), "hotfix" (для быстрых исправлений), "develop" (для разработки) и "master" (для стабильных релизов).

Организация работы с ветками в git-flow предполагает создание и слияние конкретных типов веток в соответствии с целями разработки.

Недостатки git-flow включают сложность и избыточность структуры ветвления, что может быть излишним для небольших проектов.

13. Инструменты для работы с ветками Git в средствах с GUI:

- GitHub Desktop: Обеспечивает удобный интерфейс для создания, переключения и удаления веток.

- Sourcetree: Позволяет визуально управлять ветками, включая создание, слияние и удаление.

- GitKraken: Обладает функциональностью для работы с ветками и отслеживания изменений.

- GitExtensions: Предоставляет удобные инструменты для управления ветками и слияниями.

**Вывод**: В ходе исследования базовых возможностей по работе с локальными и удаленными ветками Git были рассмотрены ключевые аспекты управления ветками в системе контроля версий Git.