**Министерство образования и науки Российской Федерации**

**Федеральное государственное автономное образовательное учреждение**

**высшего образования**

**«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Институт ИМИТ**
**Кафедра Инфокоммуникаций**

**Дисциплина Языки программирования**

# ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ

## Основы ветвления Git

**Выполнил:**

Боржонов Ростислав Александрович

**студент 2 курса, 1 группы**

**специальности ИТС-б-о-20-1**

**очной формы обучения**

**Дата защиты**

«____» _____ 20___ г.

**Оценка** _____

**Проверил:**

Кандидат технических наук,
Доцент кафедры
инфокоммуникаций.
Воронкин Роман Александрович

_____

*(Подпись)*

**Ставрополь, 2021г.**

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

Создал общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT.

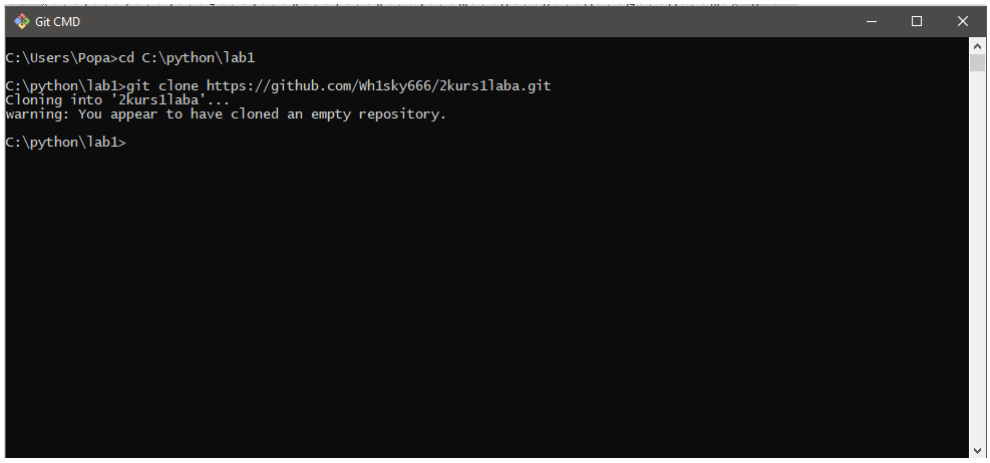


Рисунок 1. Клонирование репозитория.

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

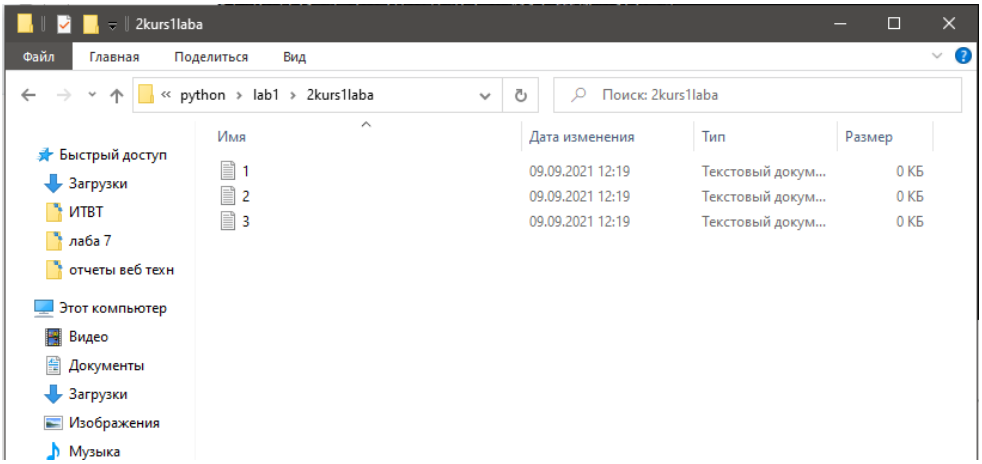


Рисунок 2. Папка с созданными тремя текстовыми документами.

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

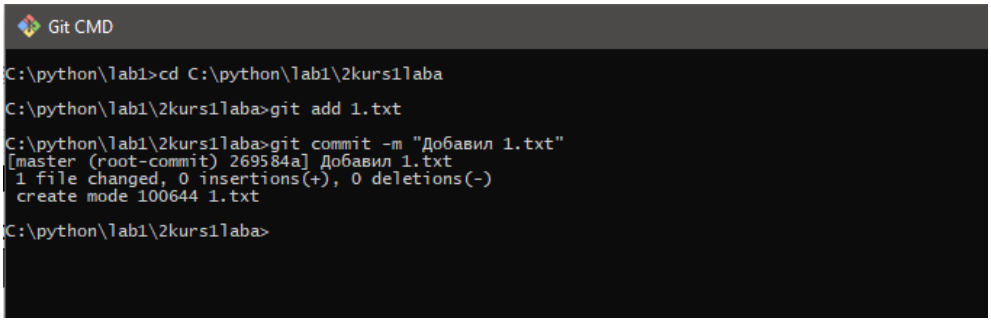


Рисунок 3. Индексация 1ого файла.

Проиндексировал второй и третий файлы.

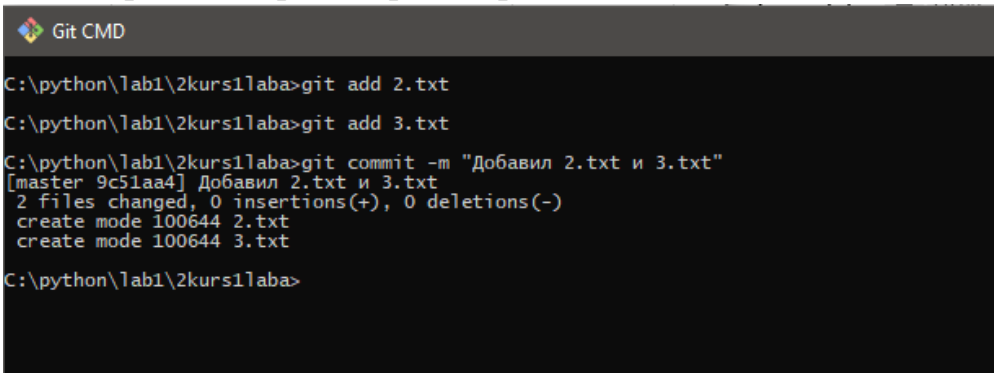


Рисунок 4. Индексация 2ого и 3его файла.

Создал новую ветку my_first_branch.

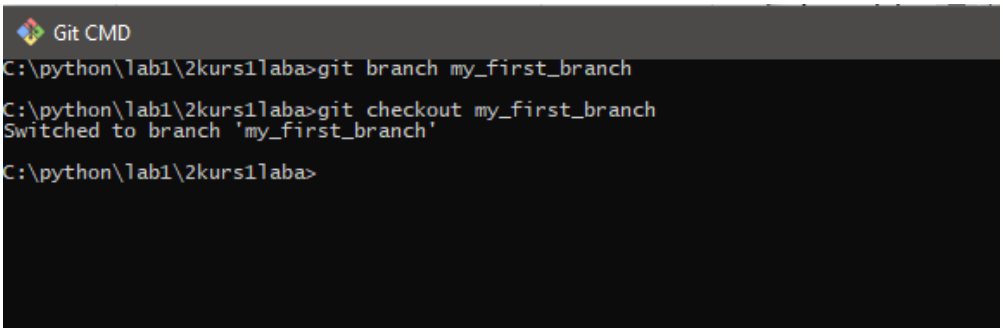


Рисунок 5. Создание новой ветки.

Перешел на ветку и создал новый файл in_branch.txt, закоммитил изменения.

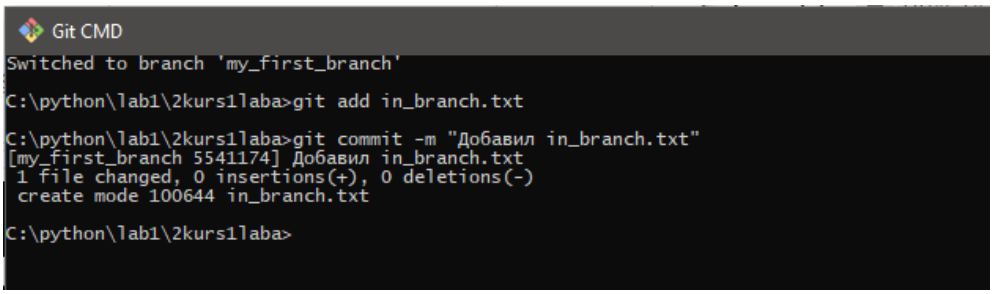


Рисунок 6. Коммит изменений.

Создал и сразу перешел на ветку new_branch.

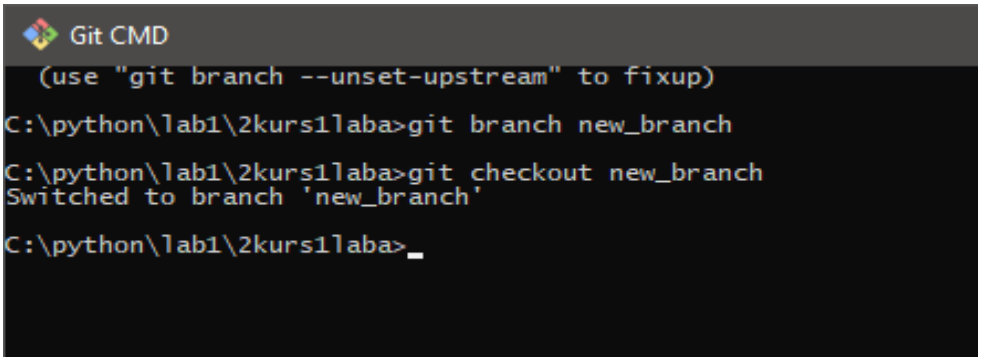


Рисунок 7. Создал новую ветку.

Сделал изменения в файле 1.txt, добавить строчку “new row in the 1.txt file”, закоммитил изменения.

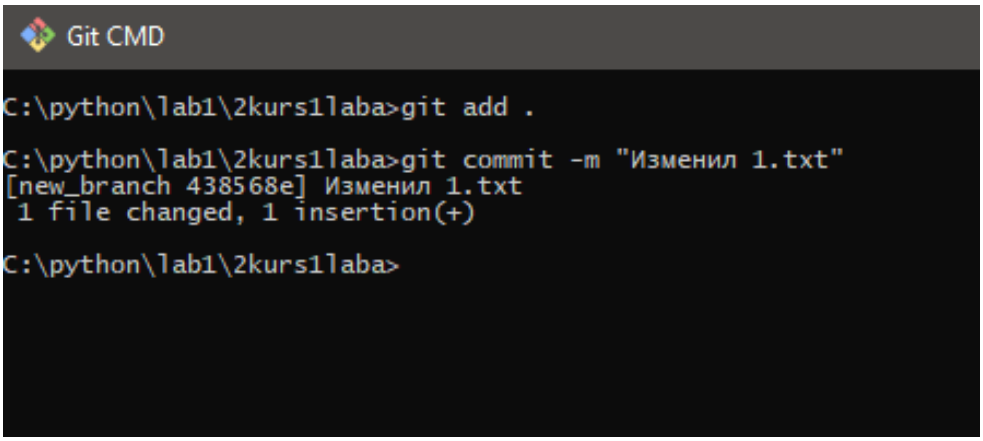


Рисунок 8. Коммит изменений.

Перешел на ветку master и слил ветки master и my_first_branch, после чего слил ветки master и new_branch.

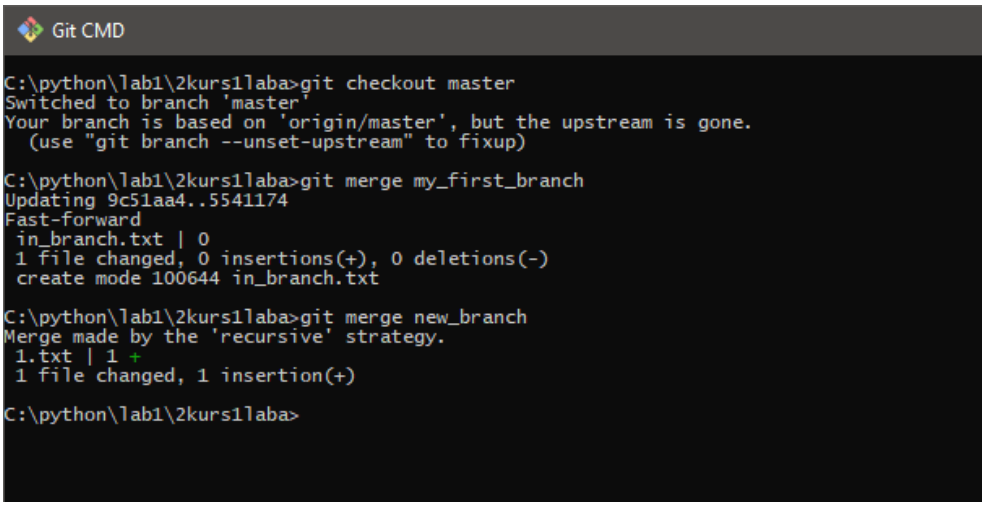


Рисунок 9. Слияние веток main, my_first_branch и new_branch.

Удалил ветки my_first_branch и new_branch. Создал ветки branch_1 и branch_2.

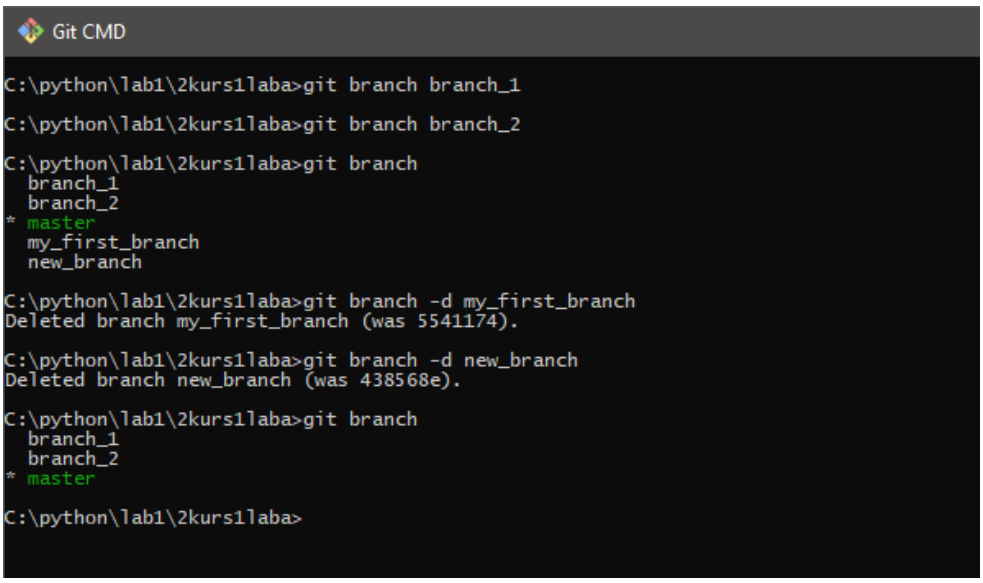


Рисунок 10. Удаление веток my_first_branch и new_branch, и создание my_first_branch и new_branch.

Перешел на ветку branch_1 и изменил файл 1.txt, удалил все содержимое и добавил текст “fix in the 1.txt”, изменл файл 3.txt, удалил все содержимое и добавил текст “fix in the 3.txt”, закоммитил изменения.

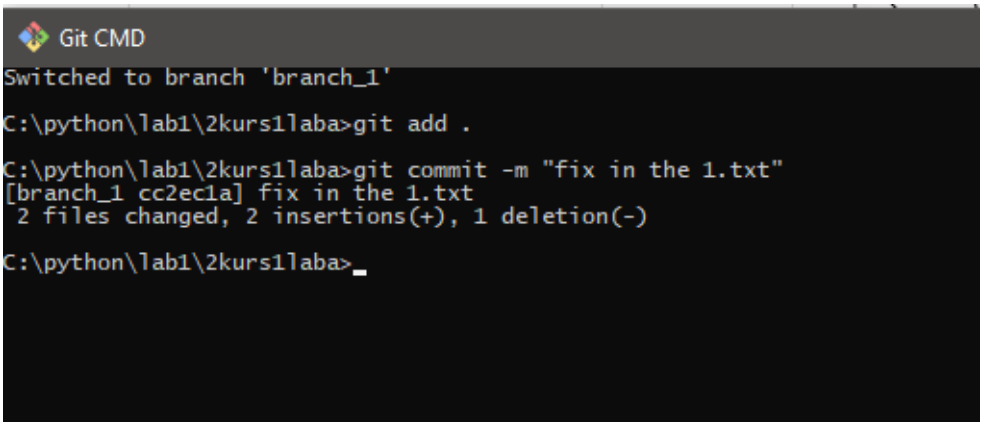


Рисунок 11. Коммит изменений.

Перейти на ветку branch_2 и также изменил файл 1.txt, удалил все содержимое и добавил текст “My fix in the 1.txt”, изменить файл 3.txt, удалить все содержимое и добавить текст “My fix in the 3.txt”, закоммитить изменения.

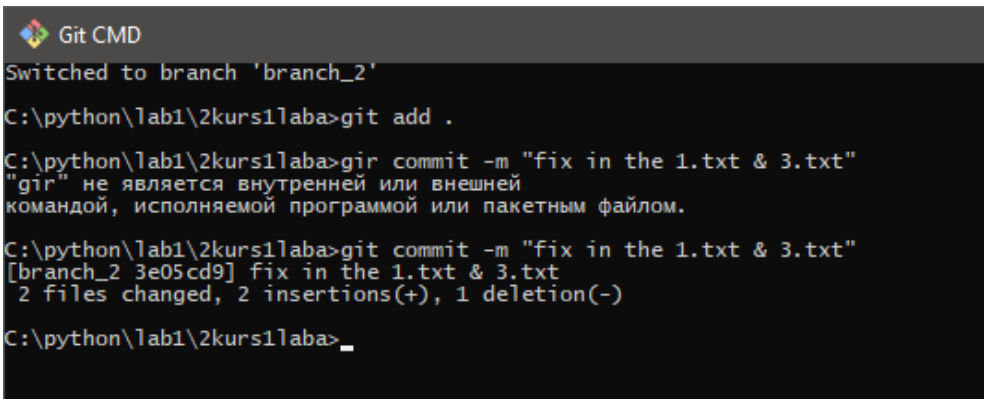


Рисунок 12. Коммит изменений.

Слил изменения ветки branch_2 в ветку branch_1.

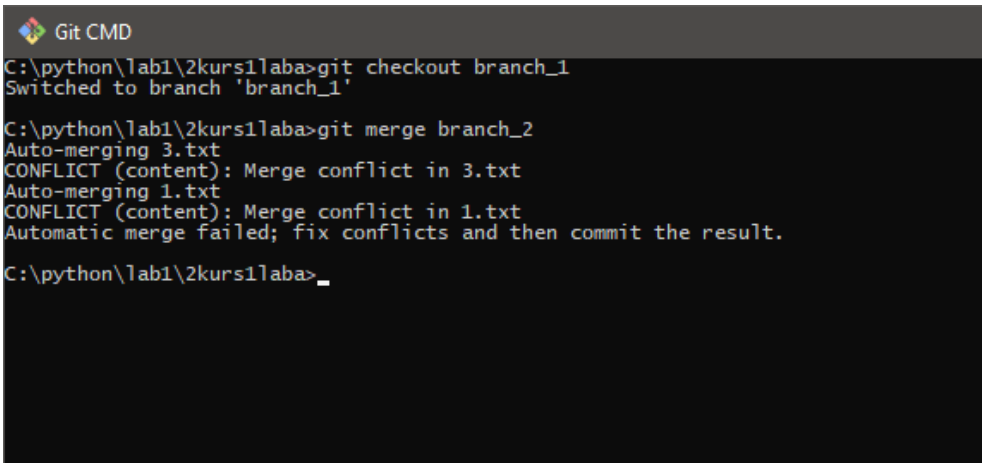


Рисунок 13. Слияние ветки branch_2 в ветку branch_1

Решил конфликт файла 1.txt в ручном режиме

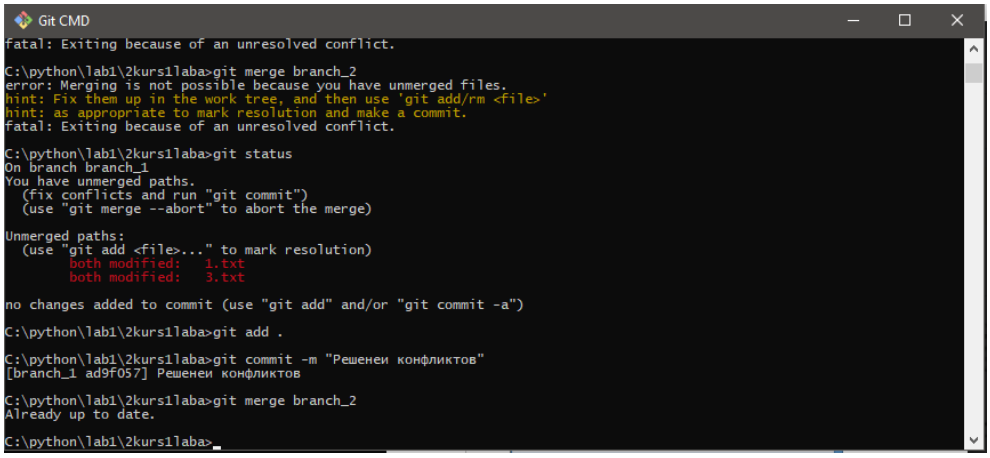


Рисунок 14. Решение конфликта файлов.

Отправить ветку branch_1 на GitHub.

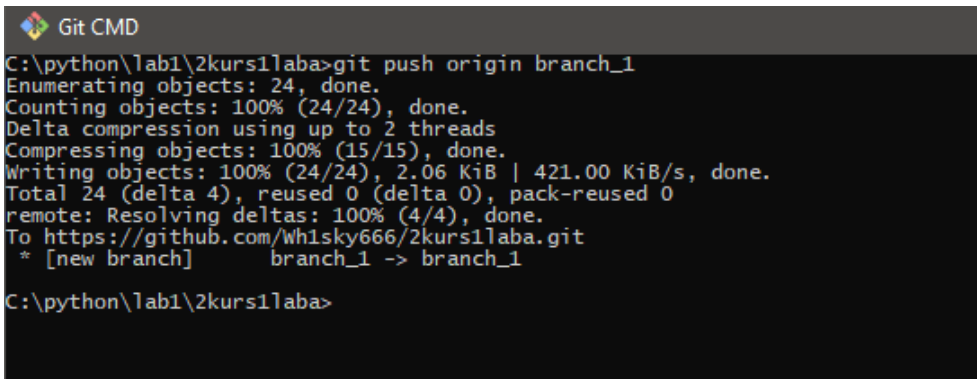


Рисунок 15. Отправление ветки branch_1 на GitHub.

Создал средствами GitHub удаленную ветку branch_3.

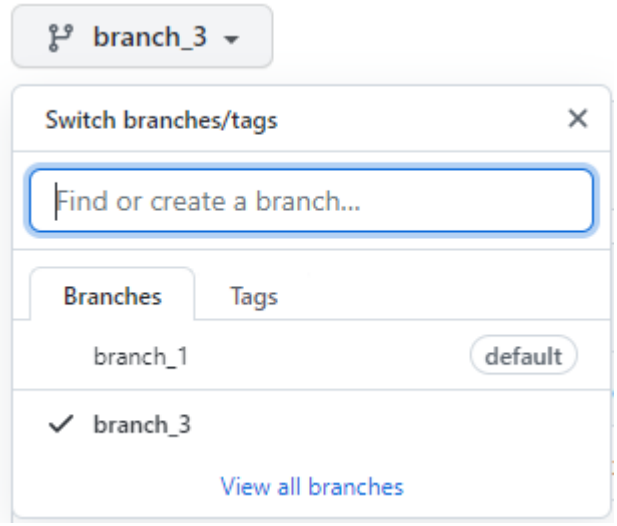


Рисунок 16. Создание удаленной ветки branch_3.

Создал в локальном репозитории ветку отслеживания удаленной ветки branch_3.

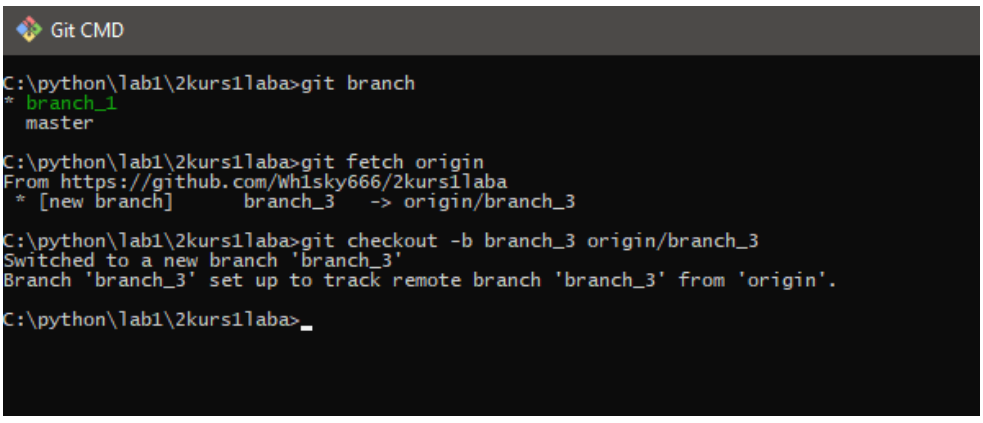


Рисунок 17. Сделал отслеживание branch_3 в локальном репозитории.

Перешел на ветку branch_3 и добавил файл файл 2.txt строку "the final fantasy in the 4.txt file". Выполнил перемещение ветки master на ветку branch_2. Отправил изменения веток master и branch_2 на GitHub.

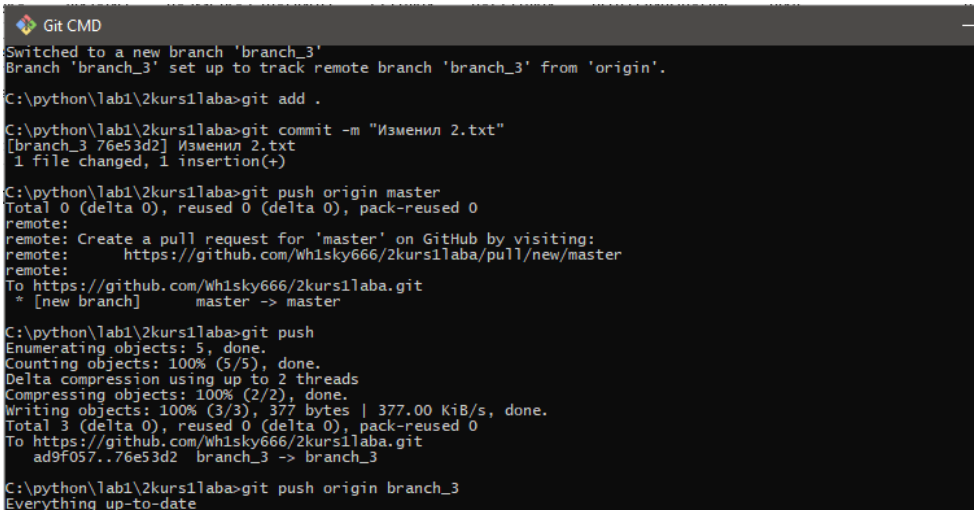


Рисунок 18. Коммит изменений во втором файле. Отправление веток master и branch_2 на GitHub.

Ответы на контрольные вопросы:

1. Что такое ветка?

Ветка в Git — это простой перемещаемый указатель на один из коммитов. По умолчанию, имя основной ветки в Git — master.

2. Что такое HEAD?

HEAD – это указатель, задача которого ссылаться на определенный коммит в репозитории.

3. Способы создания веток.

Ветки можно создать с помощью команды “git branch имя_ветки”, и чтобы перейти на неё необходимо использовать команду “git checkout имя_ветки”. Можно же создать ветку и сразу перейти на неё с помощью “git checkout –b имя_ветки”.

4. Как узнать текущую ветку?

С помощью команды git branch высветятся все ветки, текущая будет подкрашена и/или будет со знаком *.

5. Как переключаться между ветками?

Чтобы переходить по веткам необходимо использовать команду “git checkout имя_ветки”

6. Что такое удаленная ветка?

Это ветка, находящаяся на удалённом сервере GitHub.

7. Что такое ветка отслеживания?

Ветки слежения — это ссылки на определённое состояние удалённых веток. Это локальные ветки, которые напрямую связаны с удалённой веткой. Если, находясь на ветке слежения, выполнить git pull, то Git уже будет знать с какого сервера получать данные и какую ветку использовать для слияния. При клонировании репозитория, как правило, автоматически создаётся ветка master, которая следит за origin/master.

8. Как создать ветку отслеживания?

С помощью команды git checkout -b имя_ветки origin/имя_ветки.

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

С помощью команды git push имя_ветки.

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

Git fetch лишь показывает изменения веток на сервере, но не копирует их на локальный репозиторий, в отличие от команды Git pull.

11. Как удалить локальную и удаленную ветки?

Локальную ветку можно удалить с помощью команды git branch –d имя_ветки.

12. Изучить модель ветвления git-flow. Какие основные типы веток присутствуют в модели git-flow? Как организована работа с ветками в модели git-flow? В чем недостатки git-flow?

Git Flow описывает несколько веток для разработки, релизов и взаимодействия между ними. Репозиторий содержит 2 главные ветки:

– master (стандартная/основная ветка);

– develop (ответвляется от основной, в нее добавляются новшества, затем сливается с основной);

Как следствие master выступает релизной веткой в этой концепции. В Git Flow мы можем использовать следующие типы веток:

– Feature branches;

– Release branches;

– Hotfix branches;

Минусы git-flow:

– git flow может замедлять работу;

– релизы сложно делать часто;

– большие функции могут потратить дни на конфликты и форсировать несколько циклов тестирования;

**Вывод:** исследовали базовые возможности по работе с локальными и удаленными ветками Git.