Федеральное агентство связи

Ордена трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский технический университет связи и информатики»

Кафедра Математической кибернетики и информационных технологий

Отчет по лабораторной работе № 1

«Основы работы с системой контроля версий»

по дисциплине «Объектно-ориентированное программирование»

Выполнила студентка

Группы БФИ1901

Киселева Анна Сергеевна

Проверила: Мосева М.С

Москва, 2020

**Цель работы:** изучение базовых функций системы контроля версий на примере Git.

**Задание:** Определить проект (состоящий из нескольких файлов), который будет добавлен под версионный контроль, и произвести ряд операций в СКВ, включающих:

1. Создание репозитория или клонирование существующего

2. Внесение ряда изменений в файл(ы) и выполнение коммитов (кол-во не меньше 2)

3. Создание веток и выполнение ряда изменений в проекте в созданных ветках

4. Слияние ветки с веткой мастер без конфликтов

5. Слияние ветки с веткой мастер с решением возникающих конфликтов

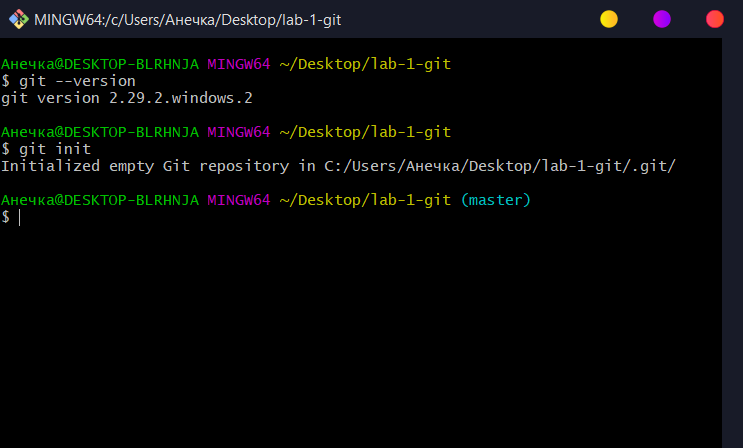
6. Откат изменений до одного из предыдущих коммитов

Ход работы

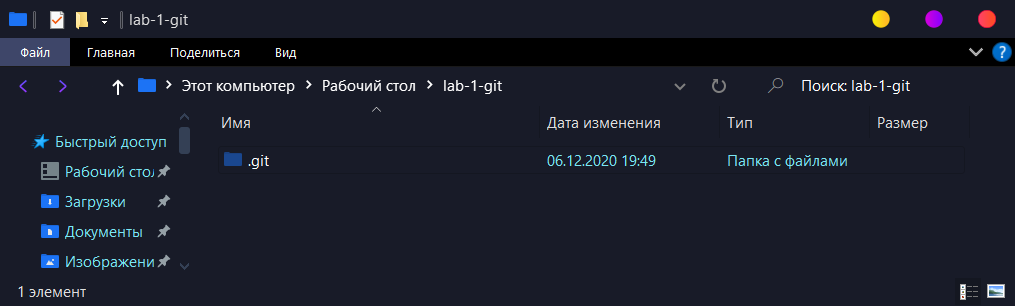
№1



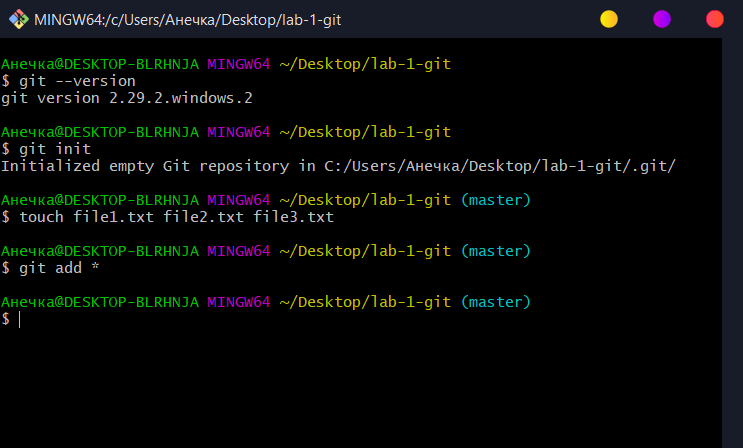
С помощью команды $git init мы выбираем папку, куда мы сохраним созданный репозиторий. Где будет образована ветка master.



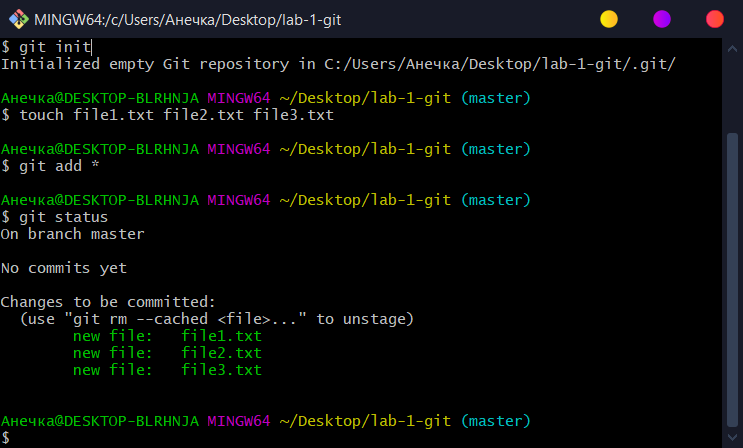
Демонстрируем работу команды, что папка создалась.



Дальше мы создаем в этой папке три файла, с которыми в дальнейшем будем работать. И командой $ git add \* охватываем эти файлы, чтобы закоммитить их. \* - для всех файлов, если бы ее не было, коммит образовался бы только на первом (верхнем).



Командой $ git status проверяем статус наших команд. Что файлы созданы и готовы к коммиту (коммитов еще нет).

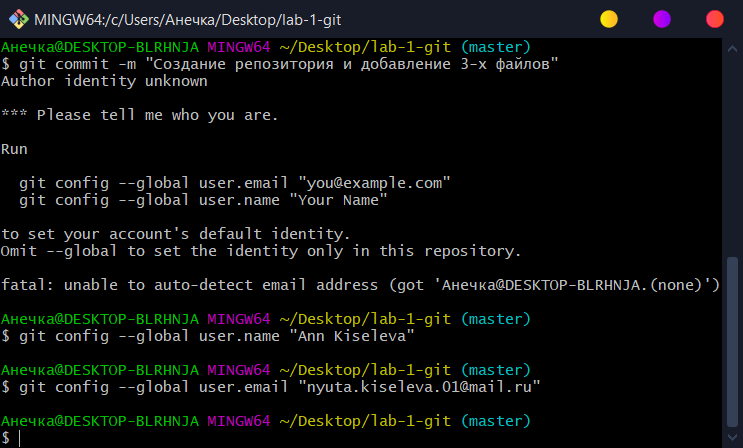


№2

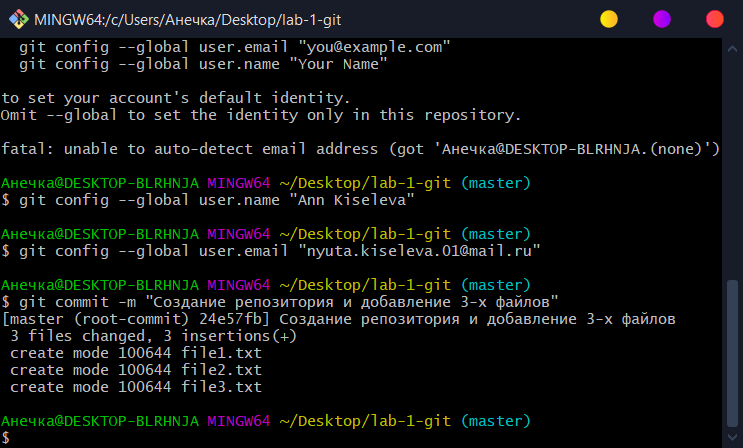
Добавление коммитов командой $ git commit -m. Сообщаем в коммите о том, что мы сделали.

Возникает ошибка, так как перед этим действием нужно было идентифицироваться.

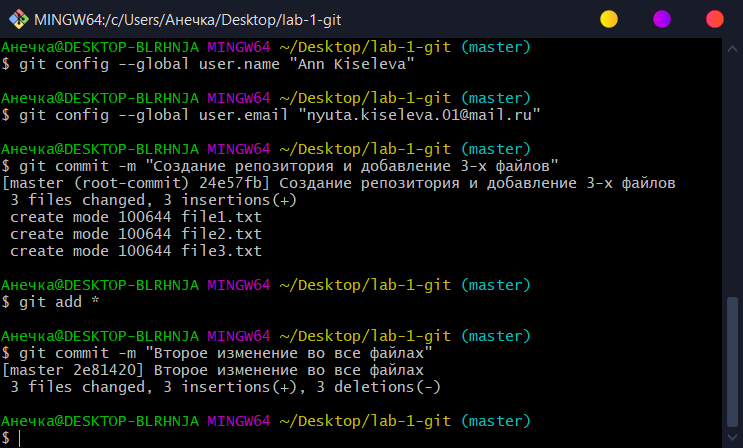
Идентифицируемся: вводим имя и почту с помощью команды $ git config --global.



Повторяем попытку добавить коммит. Все получилось удачно.

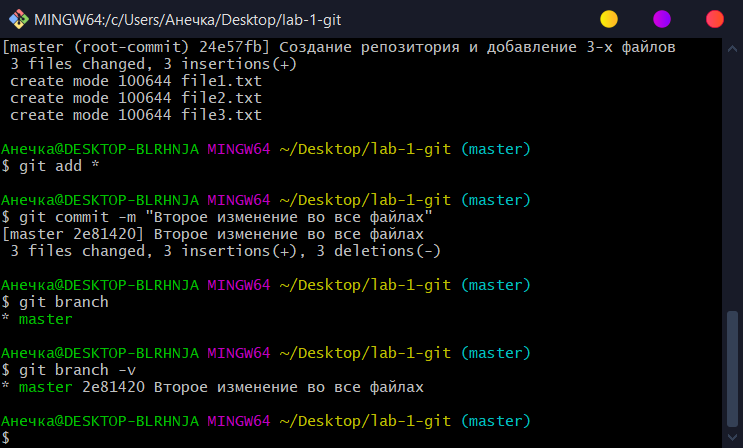


Снова с помощью команды $ git add\* охватываем все файлы для дальнейшего добавления второго коммита по заданию. И добавляет второй коммит с помощью команды $ git commit -m.

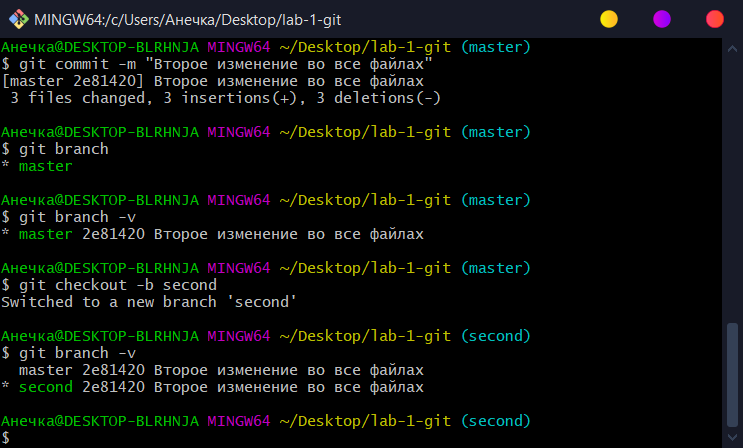


№3

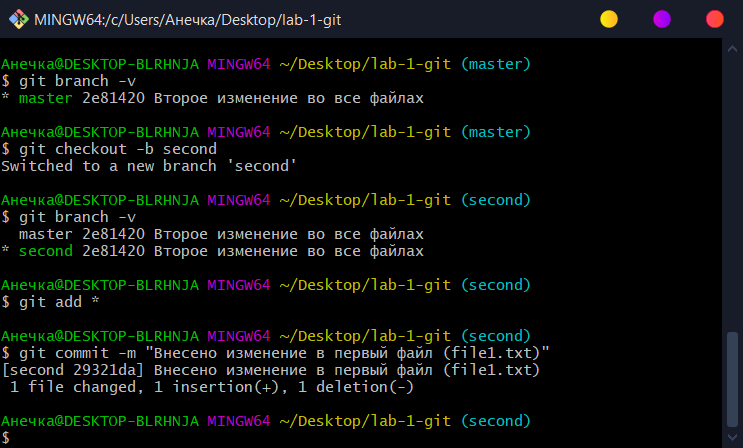
С помощью команды $ git branch проверяем список существующих веток. Команда $ git branch -v показывает последний коммит на ветке.



С помощью команды $ git checkout -b second создаем и перескакиваем на новую ветку «second».

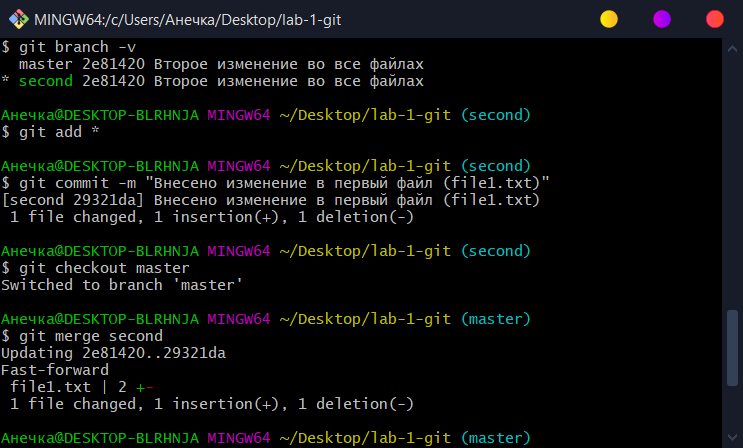


С помощью команды $ git add \* снова охватываем файлы, изменяем содержимое в одном из файлов (конкретно в первом) и с помощью $ git commit -m коммитим.



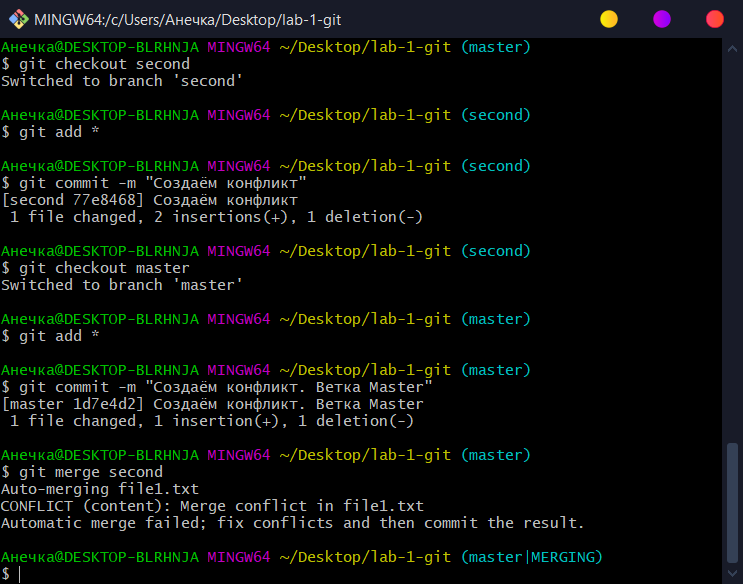
№4

С помощью $ git checkout master переключаемся на ветку «master» и командой $ git merge производим слияние ветки «second» с веткой «master». Без конфликта.

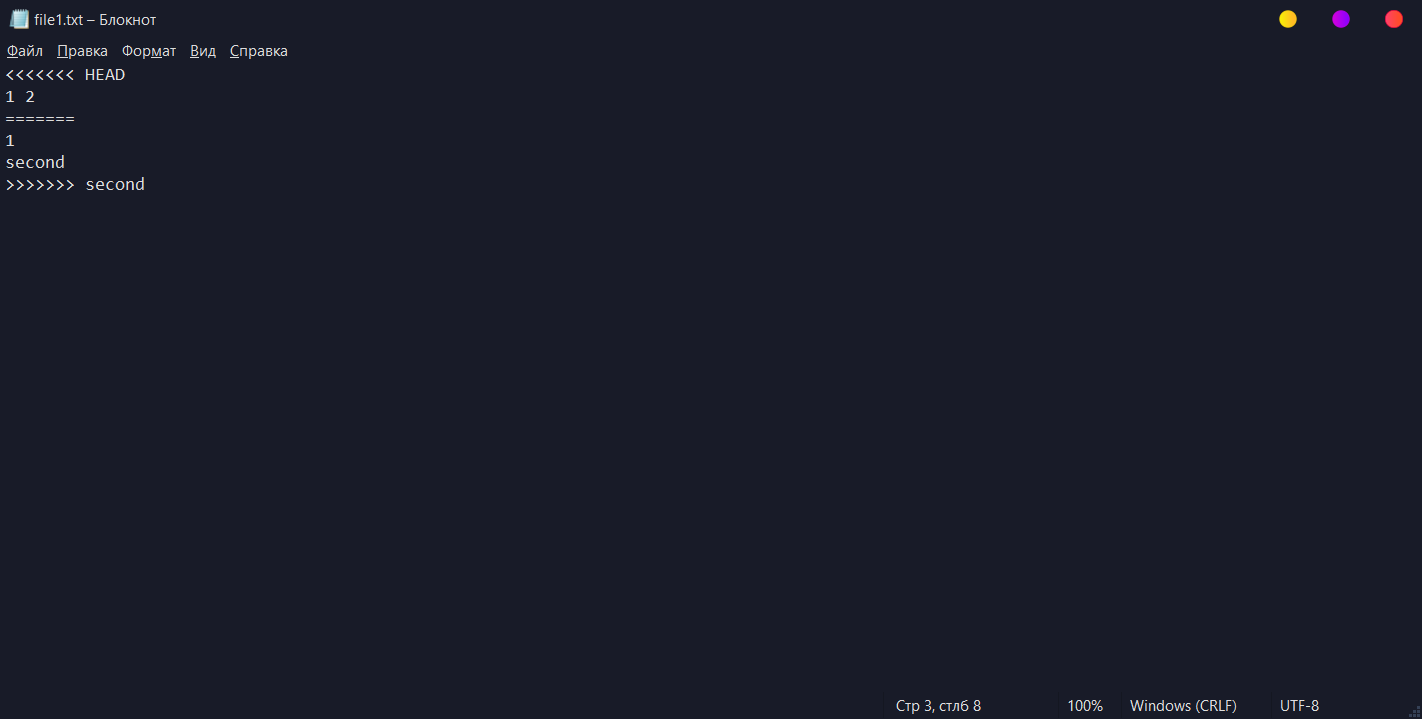


№5

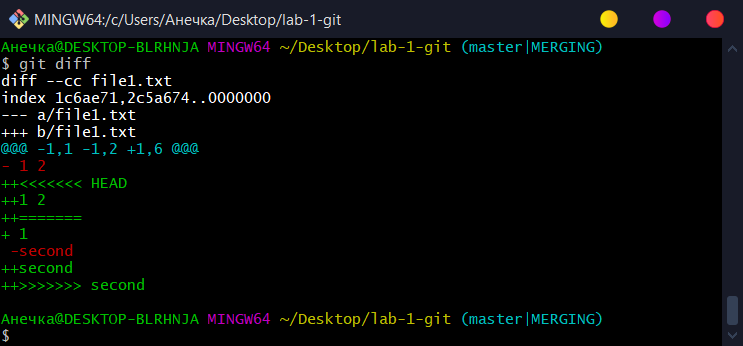
Переходим на ветку second, добавляем изменения в первый файл, коммитим, что создаем конфликт. Переходим на ветку master, так же делаем изменения в файле 1, но уже другое и так же коммитим. Пробуем слить ветки. У нас получается создать конфликт.



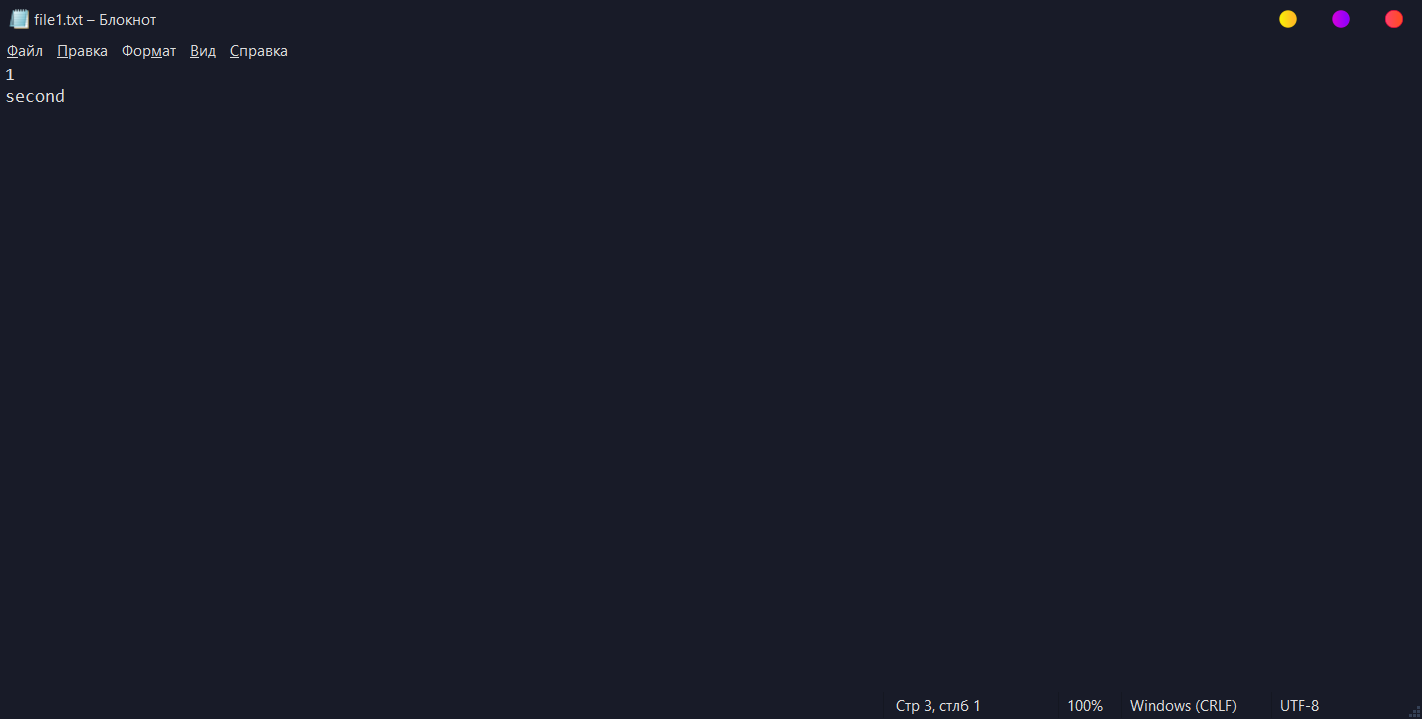
Демонстрируем это, открывая файл, где мы проводили изменения.



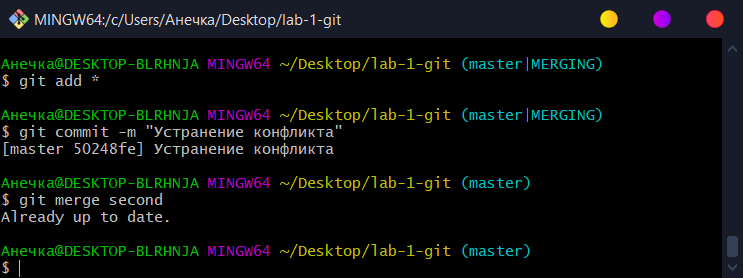
С помощью команды $ git diff проверяем конфликт в консоли.



Для того, чтобы устранить конфликт, на одной из веток подстраиваем изменения первого файла под другую ветку.

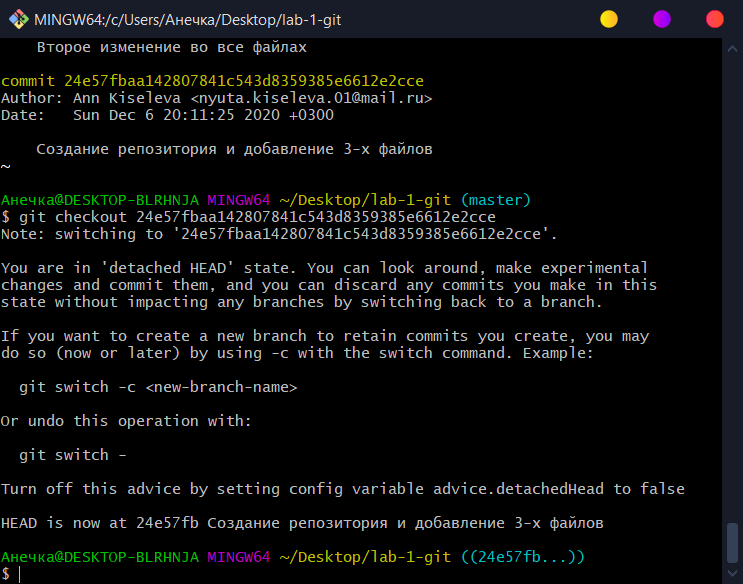


Пробуем слить ветки. Конфликт устранен.



№6

Для того, чтобы сделать откат до какого-либо изменения, нужно узнать, как это изменение назвал компьютер, узнаем это с помощью команды $ git log. Командой $ git checkout и номером изменения. Производим откат до этого изменения.



Вывод: изучила базовые функции системы контроля версий на примере Git.