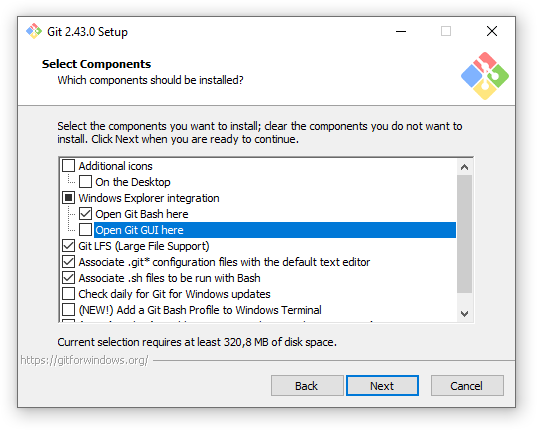
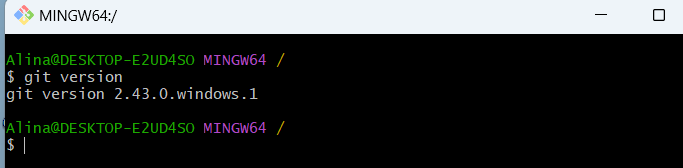
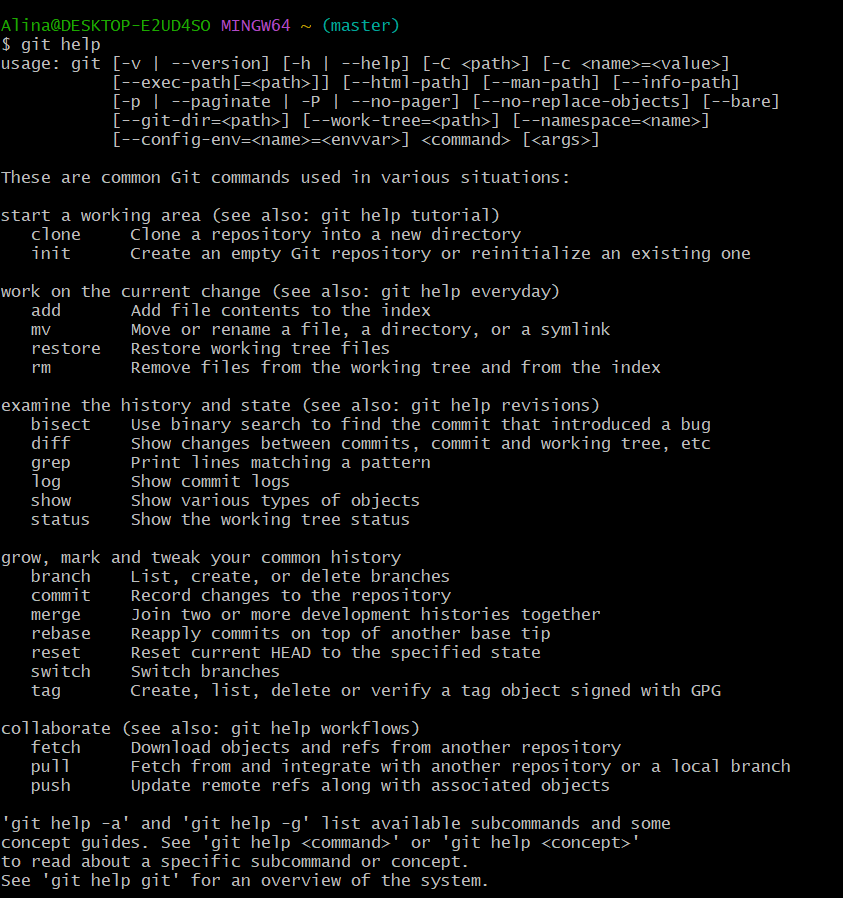
**Лабораторная работа №6**

Установка интеграции с Проводником, выбрав чекбокс: Git Bash Here

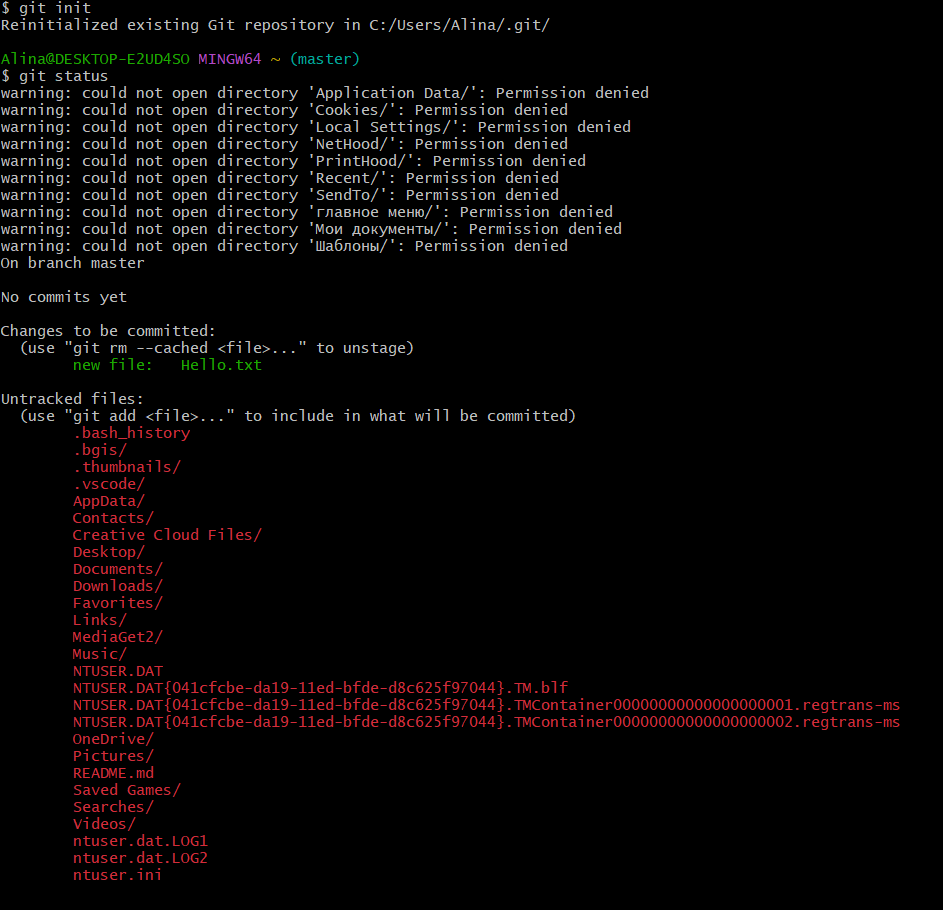


Проверка установленной версии

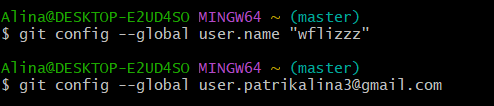


Для получения справки по Git используем команду **git help** и получим:

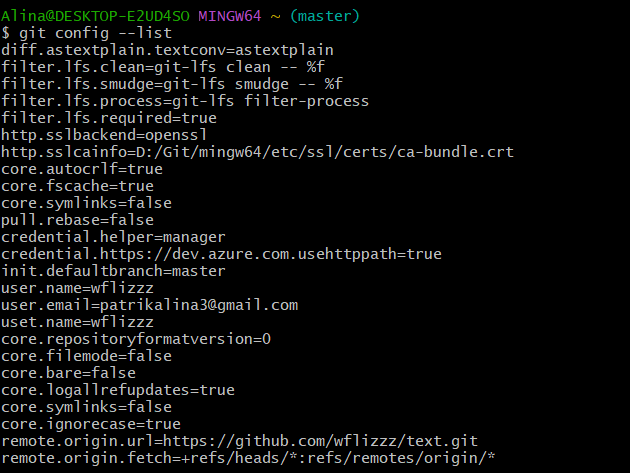
Создание локального репозитория с помощью команды **git init**



Настройка конфигурационного файла:



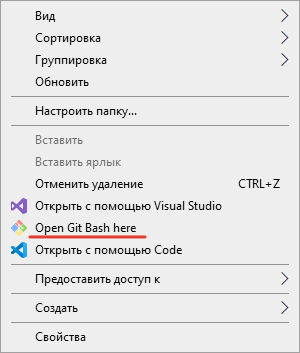
Проверка настройки конфигурационного файла командой **git config –list**



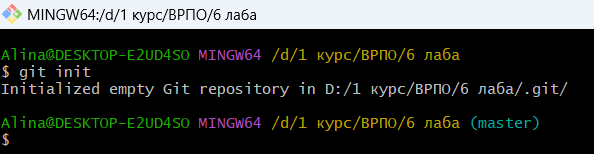
Инициализация локального репозитория в рабочей папке, где он будет размещён. Исседование изменения, которые произошли в папке:

1) Переходим в рабочую папку (в моём случаем 1 курс->ВРПО -> 6 лаба)

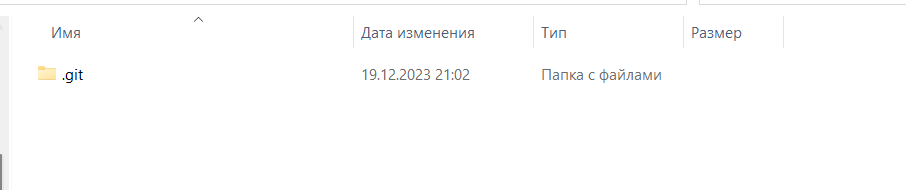
2) ПКМ открываем **Git Bash here**

****

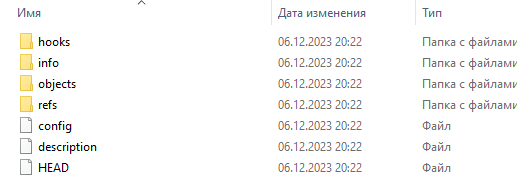
3) С помощью команды **git init** создаём локальный репозиторий



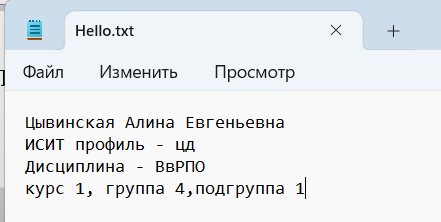
4) В папке появилась новая скрытая папка: .git – локальный репозиторий



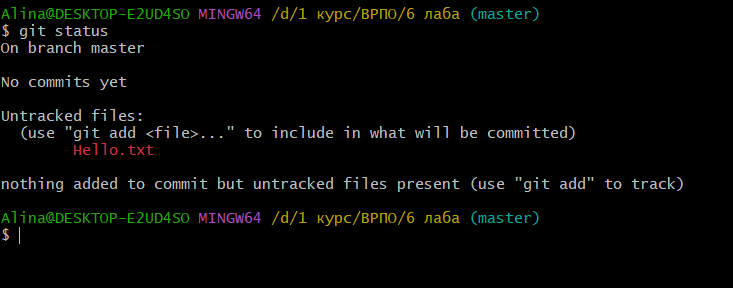
Где находятся:



Создание текстового файла Hello.txt:



Получение информации о статусе репозитория с помощью **git status**

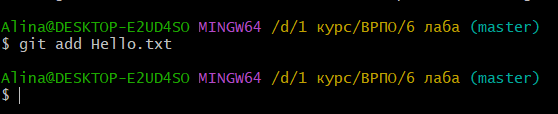
****

Можно заметить, что создана ветка **master**

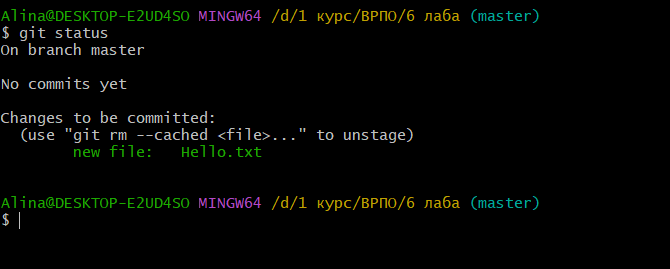
Красным цветом отмечены новые и модифицированные файлы и папки

Индексация файла Hello.txt и изменения в репозитории:

1) Добавим файл Hello.txt в репозиторий с помощью команды git add Hello.txt

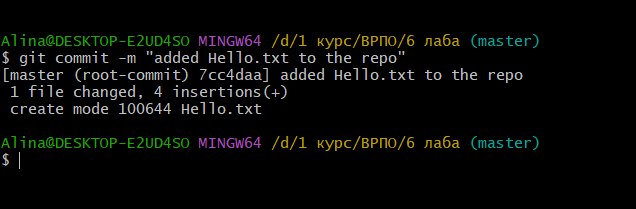


2) Проверим обновлённое состояние репозитория с помощью команды **git status**:



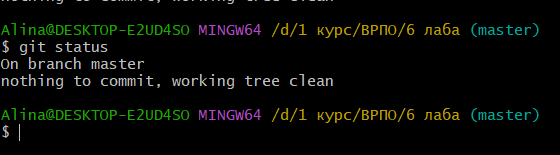
Теперь наш файл Hello.txt подсвечивается зелёным цветом, это значит, что он проиндексирован

Зафиксируем изменения индексированных файлов в репозитории с помощью команды **git commit –m “added Hello.txt to the repo”**:



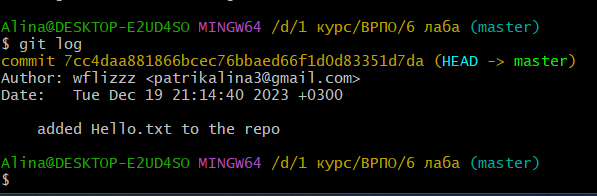
Ключ –m позволяет добавить комментарий, описывающий, что именно было изменено в коммите ("added Hello.txt to the repo")

Проверим изменения в репозитории:



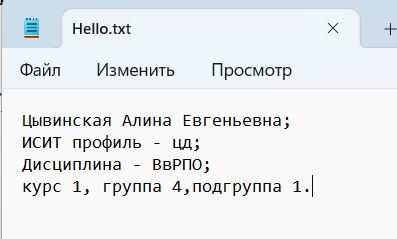
Git информирует об успешном создании нового коммита, в ветку **master** добавлен 1 файл

Просмотр журнала с помощью команды **git log**:

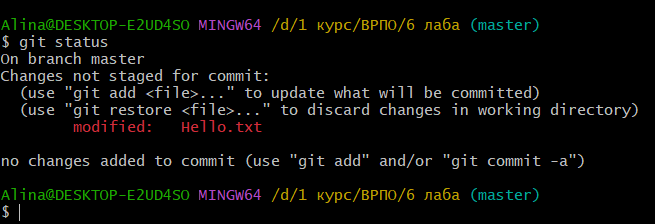


Получение информации о проиндексированных и неиндексированных файлах.

1) Изменим уже существующий файл Hello.txt и сохраним изменения

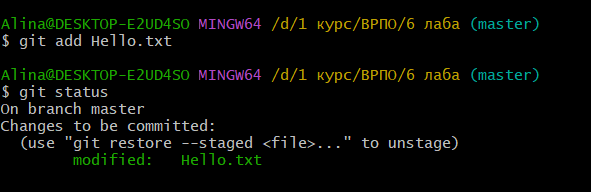


2) Просмотрим статус нашего репозитория



Можем заметить, что наш файл больше не подсвечивается зелёным цветом, и вместо надписи new file: Hello.txt теперь modified: Hello.txt, что указывает на изменения файла

3) При добавлении этого изменённого файла получим

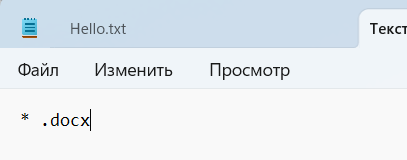


Теперь оно подсвечивается зелёным цветом

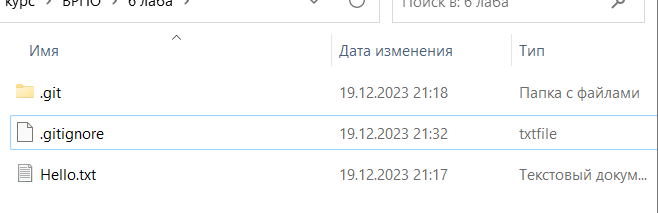
Настройка файла .gitignore

1) Создаём текстовый файл и прописываем внутри данного файла \*[расширения файлов, которые не хотим, чтобы они отображались в репозитории]

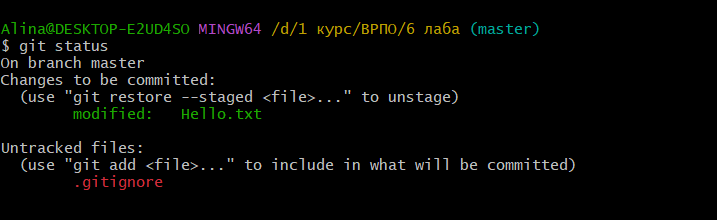
В моём случае – это файлы с расширением **.docx**



2) Далее переименовываем наш файл в файл с названием **.gitignore** убирая расширение **.txt** и сохраняем



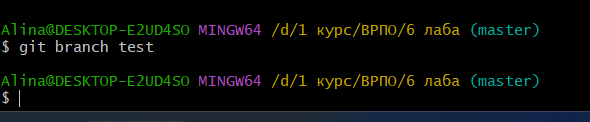
3) Далее просматриваем статус нашего репозитория и видим, что файл с расширением **.docx** больше не отображаются



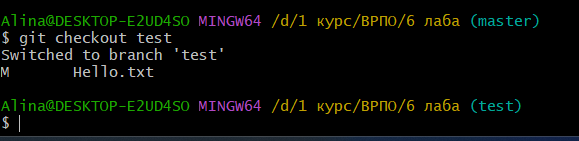
Создание новой ветки

Создание (branch) новой ветки, например, test и переключитесь (checkout) в нее

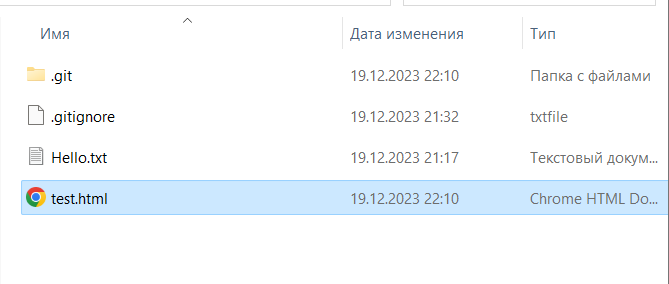
1)С помощью команды **git branch test** создаём новую ветку с именем **test**

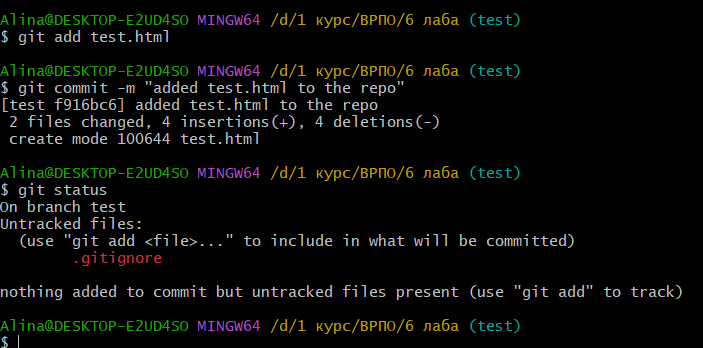
****

2) Переключим ветки с помощью команды **git checkout test**

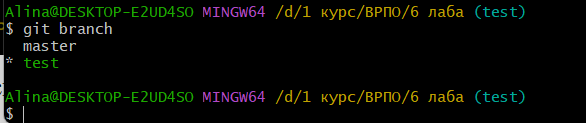
****

Создаём новый файл **test.html** и фиксируем его





C помощью команды **git branch** просмотрим список всех веток

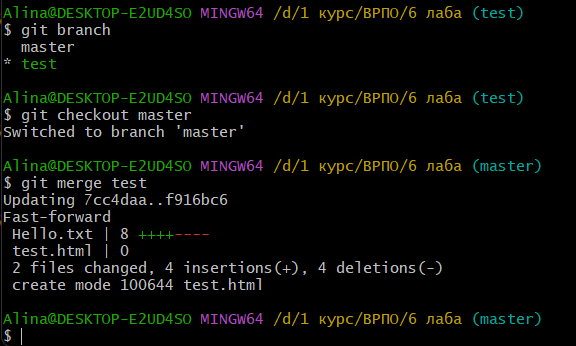


Текущей веткой являет **test**

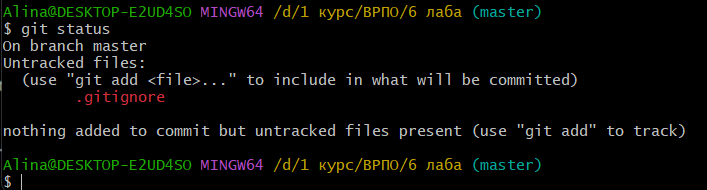
Выполним слияние веток с помощью команды **git merge**

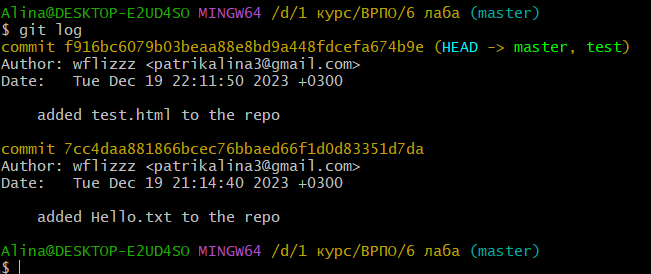
1) Переходим в ветку в которой будем выполнять слияние, а именно **master**

1.1) Для этого используем команду **git checkout master**

****

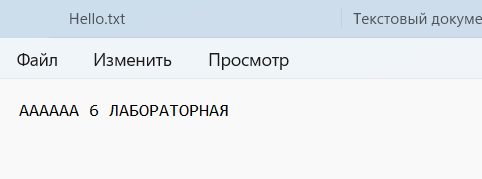
2) Просмотрим статус репозитории

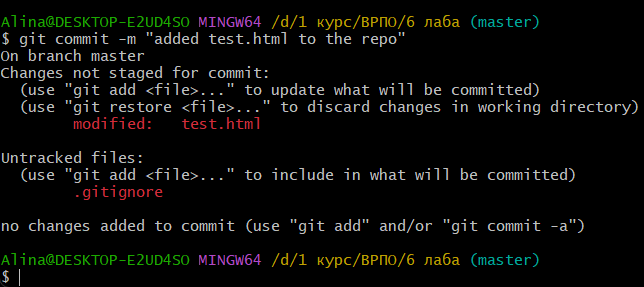




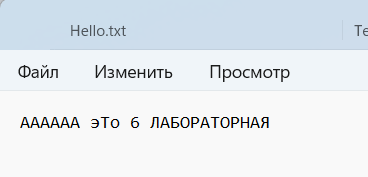
1) По очереди отредактируем файл test.html в ветках master и test. Выполним commit.

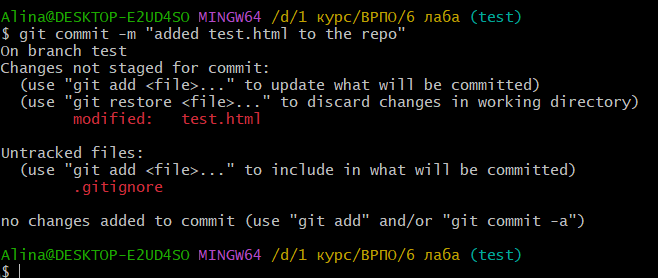
**Master**

****

****

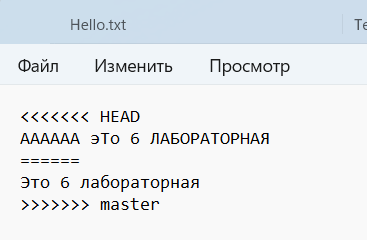
**Test**

****

****

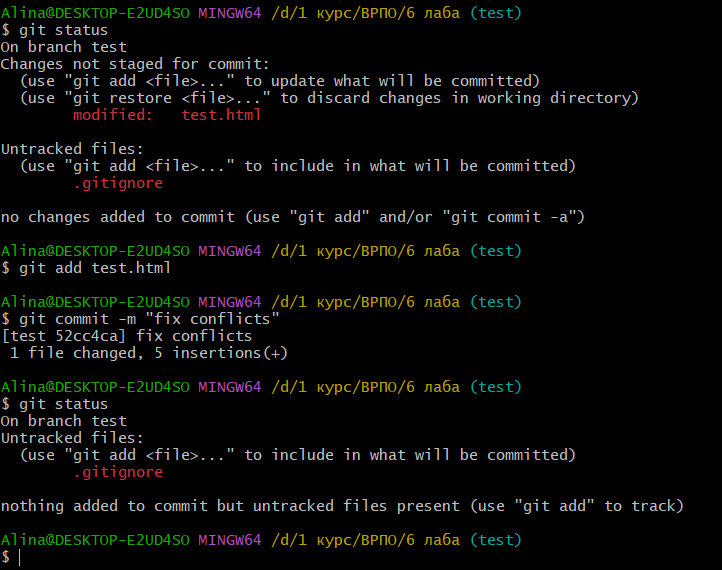
Выполним слияние в одной и второй ветках.

**Test**

****

Разрешим ситуацию конфликта слияний.

Изменим наш текстовый документ и оставим тот вариант, который нас устраивает, сохраним



1) git version (проверка установленной версии)

2) git help (справка по Git)

3) git init (создание локального репозитория)

настройка конфигурационного файла

4) git config --global user.name "имя" (имя)

5) git config --global user.email адресс электронной почты (электронная почта)

6) git config --list (проверка настройки конфигурационного файла)

7) git status (информация о статусе репозитория)

8) git add имя файла (добавление файла в репозиторий)

9) git commit -m "комментарий" (фиксация индексированных файлов в репозитории)

10) git log (просмотр журнала)

11) .gitignore (настройка файла НЕ ФУНКЦИЯ)

работа с ветками

12) git branch имя ветки (создание новой ветки)

13) git checkout имя ветки на которую переключаемся (переключение между ветками)

14) git branch(список всех веток)

15) git merge имя ветки с которой мы делаем слияние (слияние веток)