Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Ульяновский государственный технический университет»

Кафедра «Вычислительная техника»

Разработка профессиональных приложений

(название дисциплины)

**Отчёт по лабораторной работе №5**

«GIT»

(название (тема) работы)

**Вариант №4**

Выполнил:

студент группы ИВТАПбд-21

Галацков И.А.

Проверил(а):

преподаватель кафедры «ВТ»

Исхаков И. И.

Ульяновск

2023

**Постановка задачи**

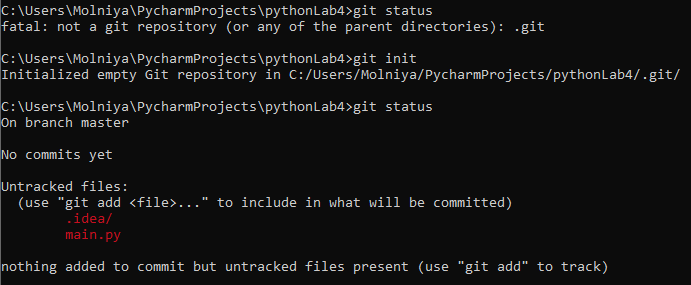
1. Установить Git (<https://git-scm.com/>)

2. Командами в КОНСОЛИ:

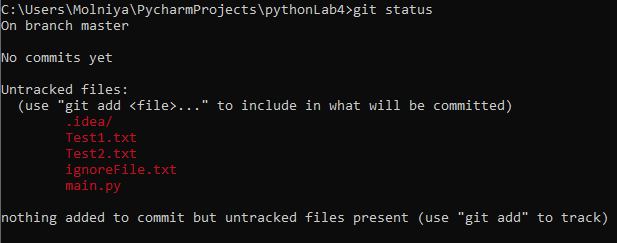
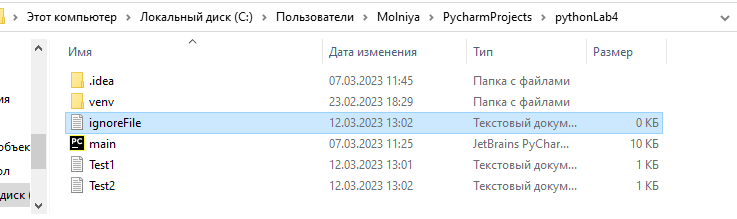
* 2.1. Создать локальный репозиторий
* 2.2. Добавить несколько файлов
* 2.3. Сделать один из файлов игнорируемым
* 2.4. Зафиксировать изменения
* 2.5. Внести изменения в файлы и зафиксировать изменения
* 2.6. Создать несколько бранчей, редактировать файлы в разных бранчах и фиксировать изменения. Осуществить слияние двух веток в одну.
* 2.7. Создать аккаунт github (если его нет, а если уже имеется, то использовать его), отправить исходные коды в удаленный репозиторий.
* 2.8. Склонировать удаленный репозиторий в папку, отличную от оригинальной.
* 2.9. Добиться конфликта файлов при применении изменений в оригинальном репозитории.
* 2.10. Разрешить конфликт.
* 2.11 Подготовить отчет о проделанной работе и прикрепить его в качестве ответа на это задание в learn. Отчет должен содержать список выполняемых в процессе выполнения лабораторной работы команд с их описанием, а также результат выполнения команд, скриншоты страниц с удаленного репозитория, чтоб были видны ветки и история операций (адрес репозитория также необходимо предоставить).

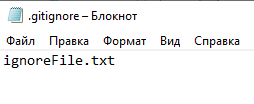
**Описание реализации**

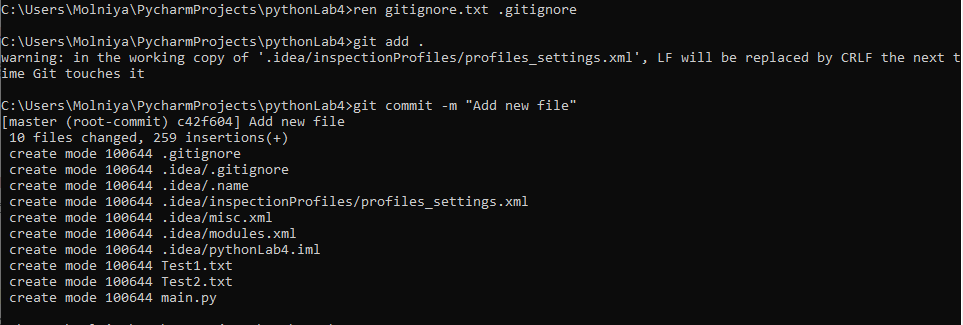
Сначала с помощью команды git init был создан локальный репозиторий



Далее для удобства выполнения работы в репозитории были созданы следующие файлы: ignoreFile, Test1, Test2

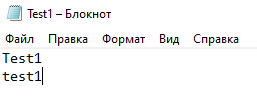
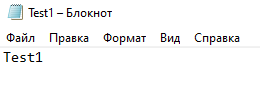
Далее создаем файл gitignore.txt и переименовываем его в «.gitignore», теперь если записать в .gitignore название файла, то он станет игнорируемым, эти файлы не будут попадать в коммиты, добавляем туда раннее созданный файл ignoreFile.txt и фиксируем изменения. С помощью команды git add . добавляем все содержимые файлы в индекс для последующего коммита, с помощью команды git commit создаем коммит фиксируя все изменения

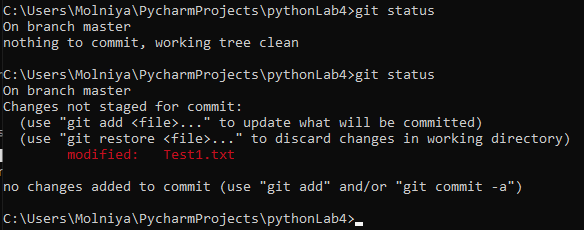


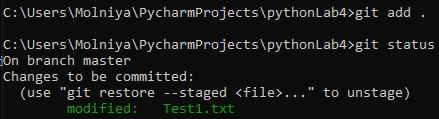


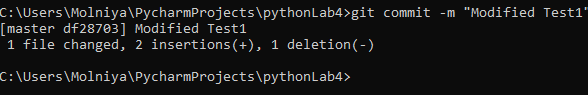
На скриншоте можем увидеть, что файл ignoreFile.txt не попал в коммит.

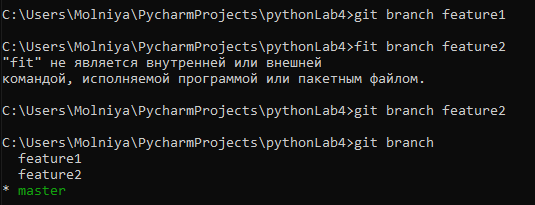
Следующим шагом добавим изменения в файл Test1.txt и зафиксируем изменения



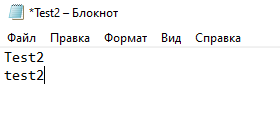


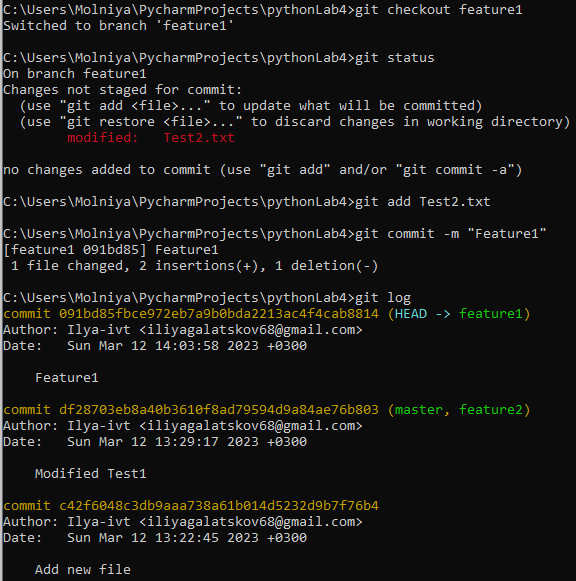


  
Далее создадим несколько бранчей с помощью команды git branch «название бранча» и с помощью команды git branch мы можем посмотреть список бранчей и тот бранч, на котором мы находимся в данный момент. Таким образом, создадим бранчи feature1 и feature2

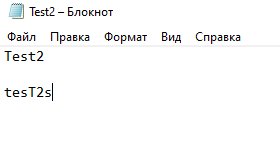


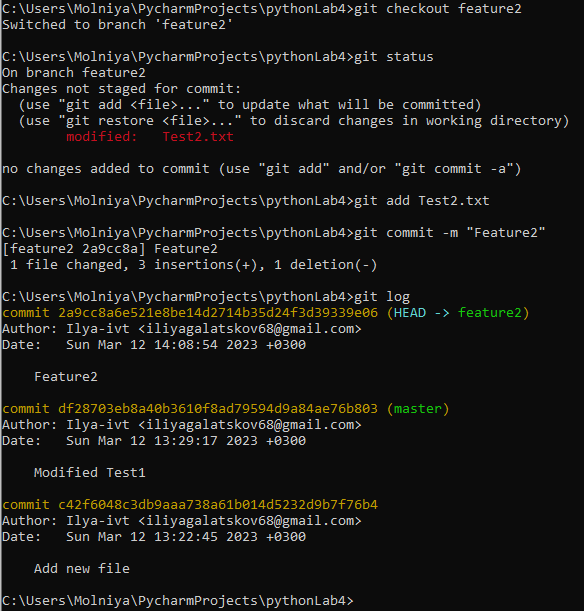
Далее с помощью команды git checkout переходим на нужный нам бранч, сначала переходим на бранч feature1 и редактируем файл Test2.txt и фиксируем изменения, с помощью команды git log мы можем посмотреть список коммитов



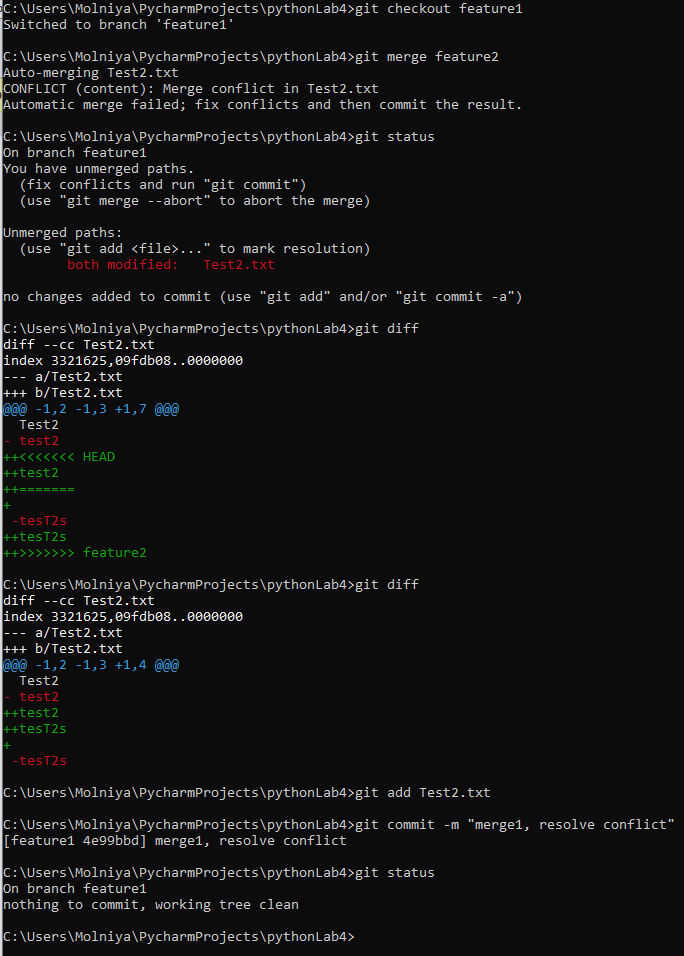


После переходим на бранч feature2 и снова вносим изменения в файл Test2.txt и фиксируем изменения

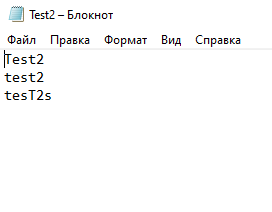




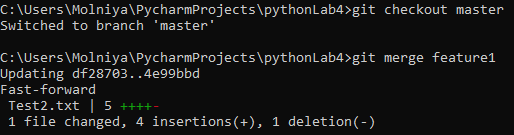
Далее осуществим слияние веток, перейдем снова на ветку feature1 и сольем ее с feature2 с помощью команды git merge, но так как мы изменяли один и тот же файл, то добились конфликта файлов, используя команду got diff мы можем посмотреть разницу между двумя деревьями и посмотреть как прошло слияние, таким образом мы увидим какие строки были удалены, а какие добавлены.



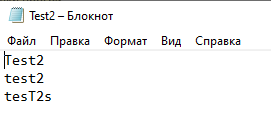
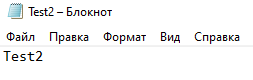
Разрешим конфликт вручную



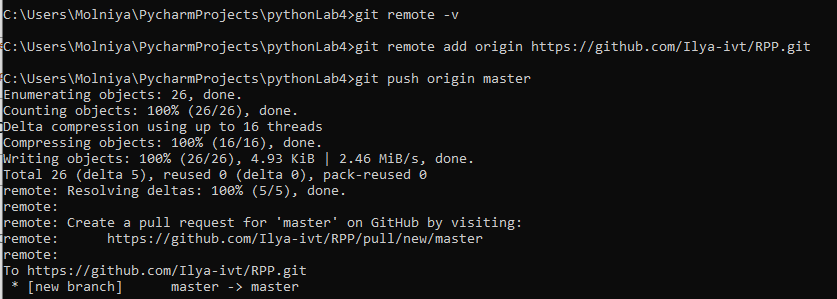
Далее перейдем на ветку master и осуществим слияние с веткой feature1

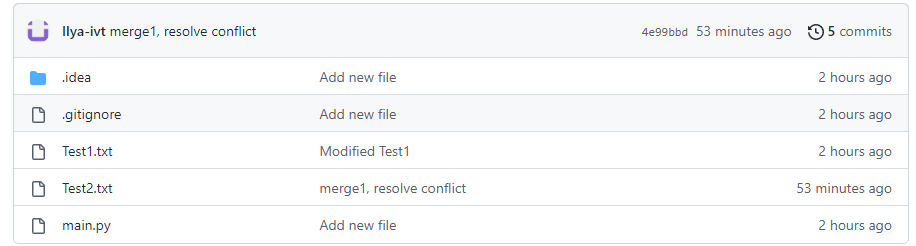


Слияние прошло без конфликтов

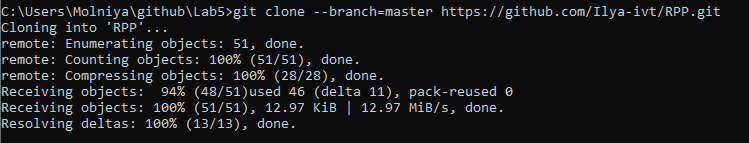


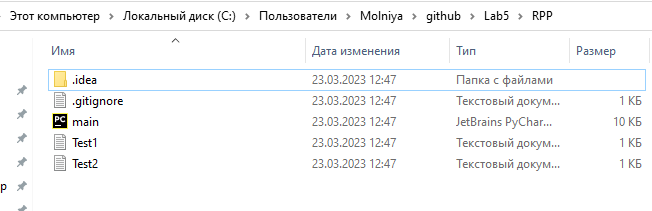
Следующем шагом отправим наш исходный каталог в удаленный репозиторий. С помощью команды git remote привязываем удаленный репозиторий по адресу к ссылке в нашем локальном репозитории. Командой git remote add добавляем ссылку на наш удаленный репозиторий, и с помощью команды git push отправляем наш каталог в удаленный репозиторий





В заключении склонируем наш удаленный репозиторий в новую папку с помощью команды git clone, так как нужные нам данные лежат в другой ветке не отмеченной по умолчанию, то при клонировании указываем нужную ветку





**Вывод**

Данную лабораторную работу можно считать завершенной, так как все поставленные задачи были выполнены, в ходе ее выполнения ,ьыли получены практические навыки работы с GIT