Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования «Ульяновский государственный технический университет»

Кафедра «Вычислительная техника»

**Отчет по лабораторной работе №5**

Дисциплина: «Разработка профессиональных приложений»

GIT

Выполнила:

студентка группы ИВТАПбд-22

Ренжина А.С.

Проверил:

преподаватель кафедры

«Вычислительная техника»

Исхаков И.И.

Ульяновск, 2023

**Задание:**

1. Установить Git

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.

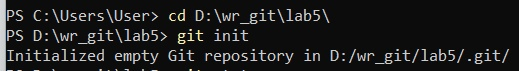
2.8. Склонировать удаленный репозиторий в папку, отличную от оригинальной.

2.9. Добиться конфликта файлов при применении изменений в оригинальном репозитории.

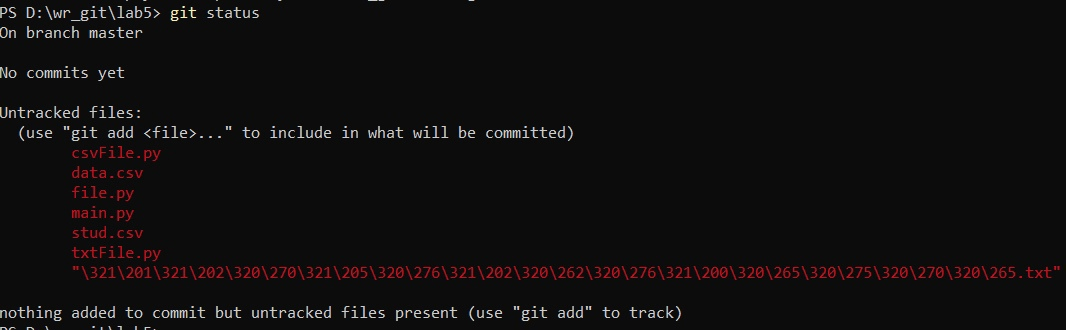
2.10. Разрешить конфликт.

**Описание реализации:**

Вся работа проходит в Windows PowerShell. Первым делом переносим файлы из лабораторной номер 4 в новую директорию на локальном компьютере c помощью команды консоли **cd <путь>**

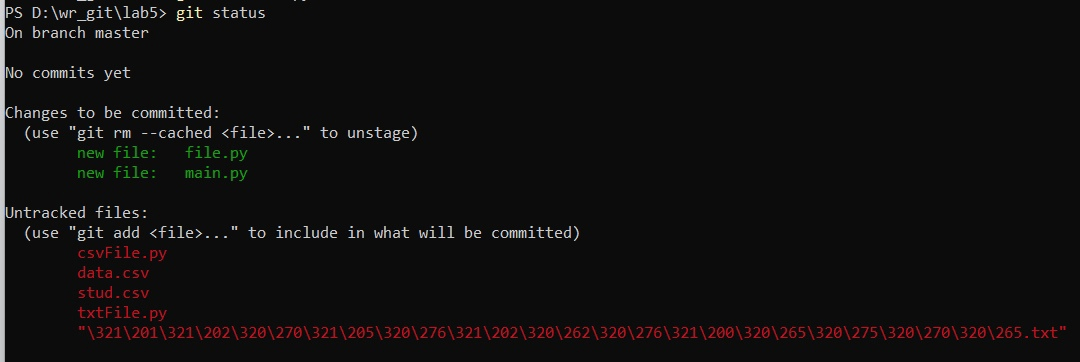
Создаем репозиторий гит с помощью команды **git init**.

Определяем статус гита, что файлы были не добавлены - **git status**.

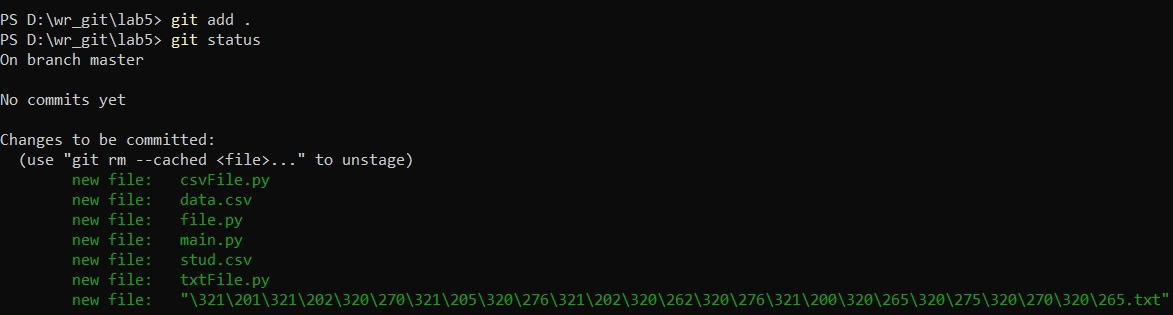


Добавляем файлы в наш репозиторий - **git add <имя\_файла>**

Проверяем статус и видно, что те файлы которые добавили, обновились, а те которые не добавили также выделены красным цветом.

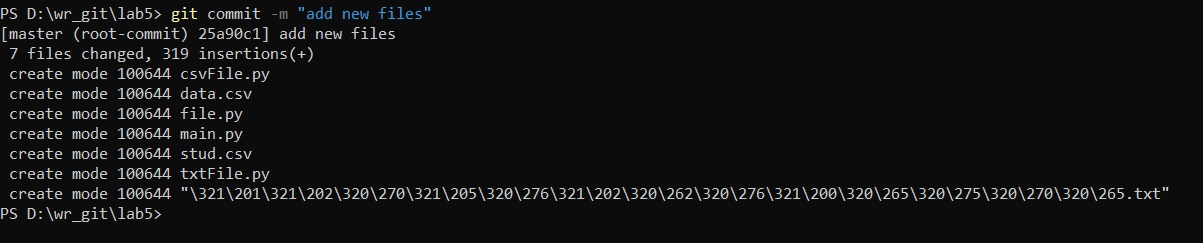


Добавим все файлы с помощью **git add .** или **git add -A** и проверим сразу статус, чтобы убедиться.

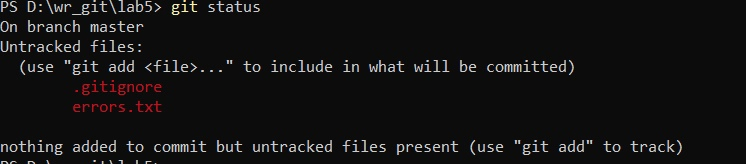


Так как мы добавили новые файлы, добавим коммит - команда

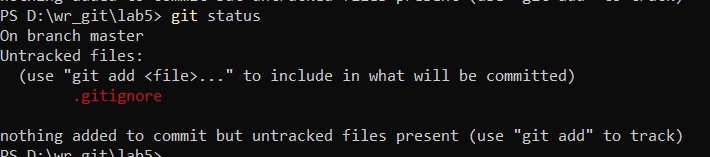
**git commit -m “текст”**

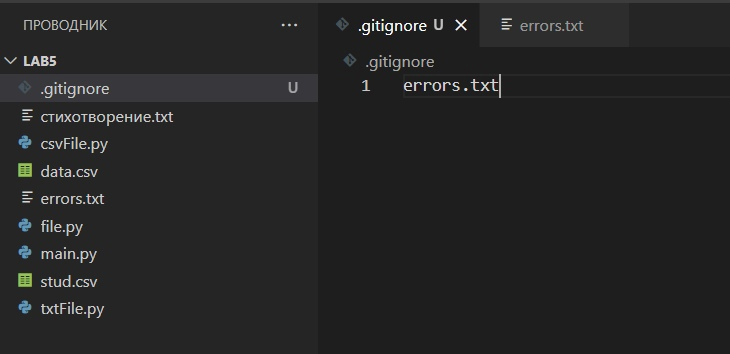
****

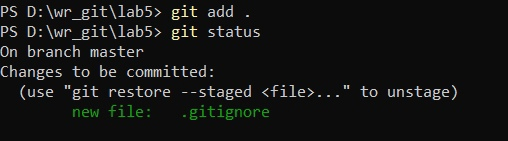
Далее, работа с gitignore. Он нужен для того, что если есть файл, который нужен только для локального компьютера и не надо добавлять в репозиторий. Создали файл errors.txt и .gitignore проверим статус.

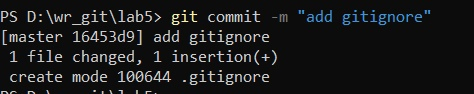


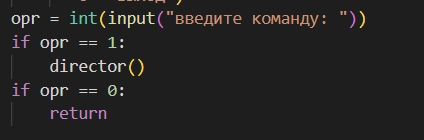
Для того, чтобы текстовый файл не отображался при проверки статуса, нужно записать название файл в .gitignore и также проверим статус.



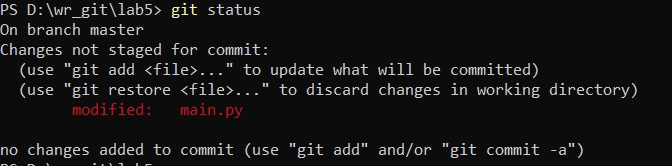


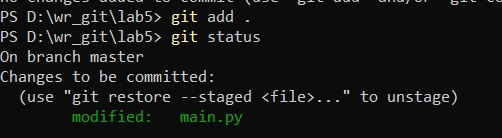
Видно, что теперь тот файл не отображается, добавим .gitignore в репозиторий.

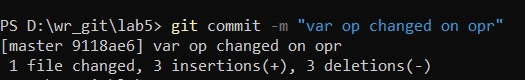
Добавим коммит.

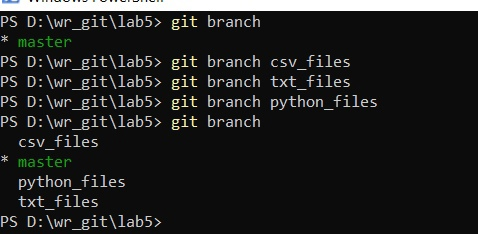
Далее работа с изменениями в файлах. Поменяем в одном из файлов название переменной. Вместо переменной op стало opr.

Проверим статус репозитория. Он выдал, что изменения не зафиксированы.



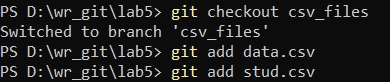
Добавим этот файл и добавим коммит к тому что изменили и проверим статус.



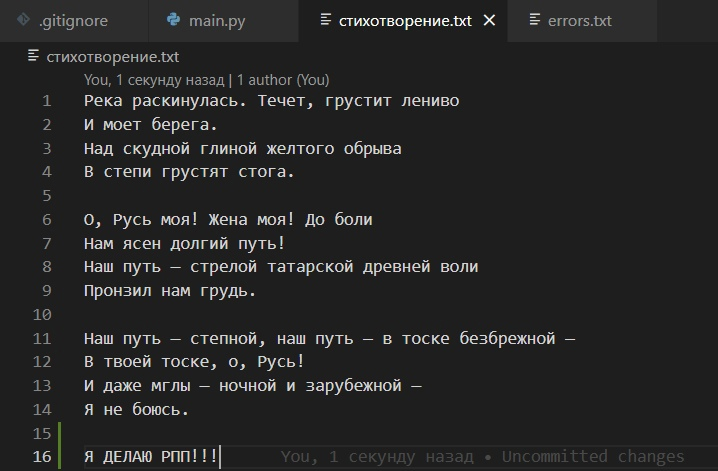
Далее работа с ветками. Так как для реальных проектов создают ветки, которые относятся к проекту по типу test и т.д, для лабораторной создаем ветки по типам файлов. Для этого нужна команда **git branch <название ветки>**, а для просмотра всех существующих веток **git branch.**

Самая главная ветвь - это ветвь master и звездочка показывает что на данный момент мы находимся на ней.

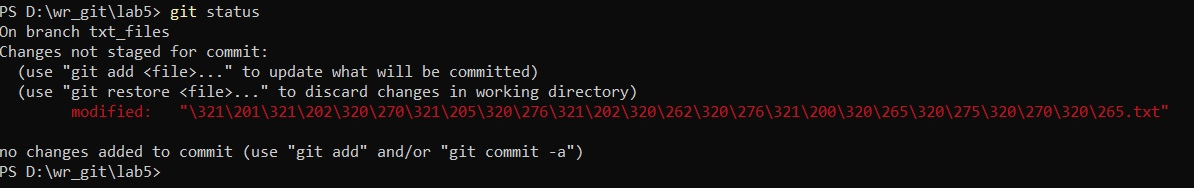
Для того чтобы переключиться на другую ветку используют **git checkout <название ветки>**.

Для добавления файлов в ветку используется команда **git add <название файла>**.

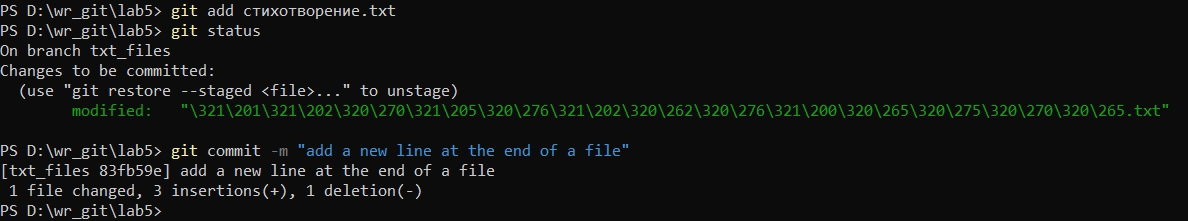
Теперь изменим файл в одной из веток.



И посмотрим на ту ветку где этот файл лежит и проверим статус, он показывает что он был изменен.

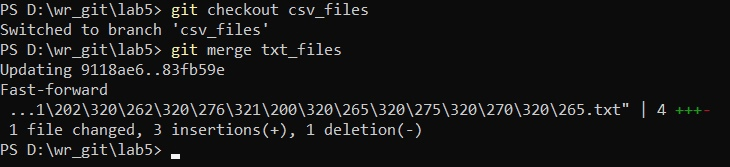


Добавим этот файл и добавим коммит о том, что изменили.



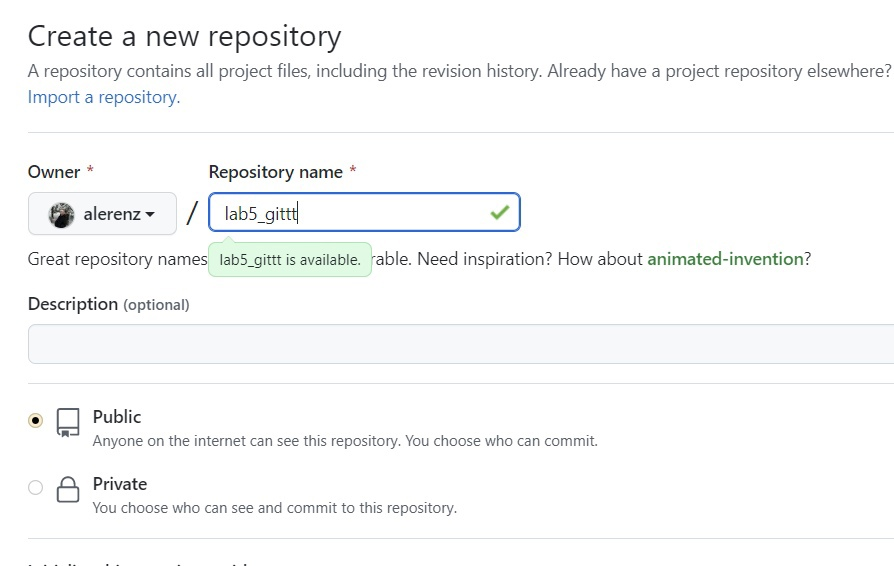
Для объединения веток используется команда **git merge <название ветки>**.

Для того чтобы к какой-то ветки присоединить другую, нужно обязательно перейти в ту ветку, в которую вы хотите объединить другую.

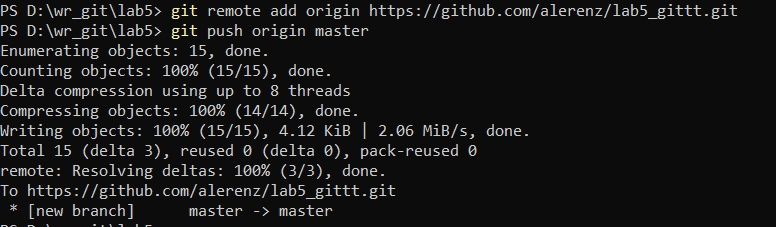


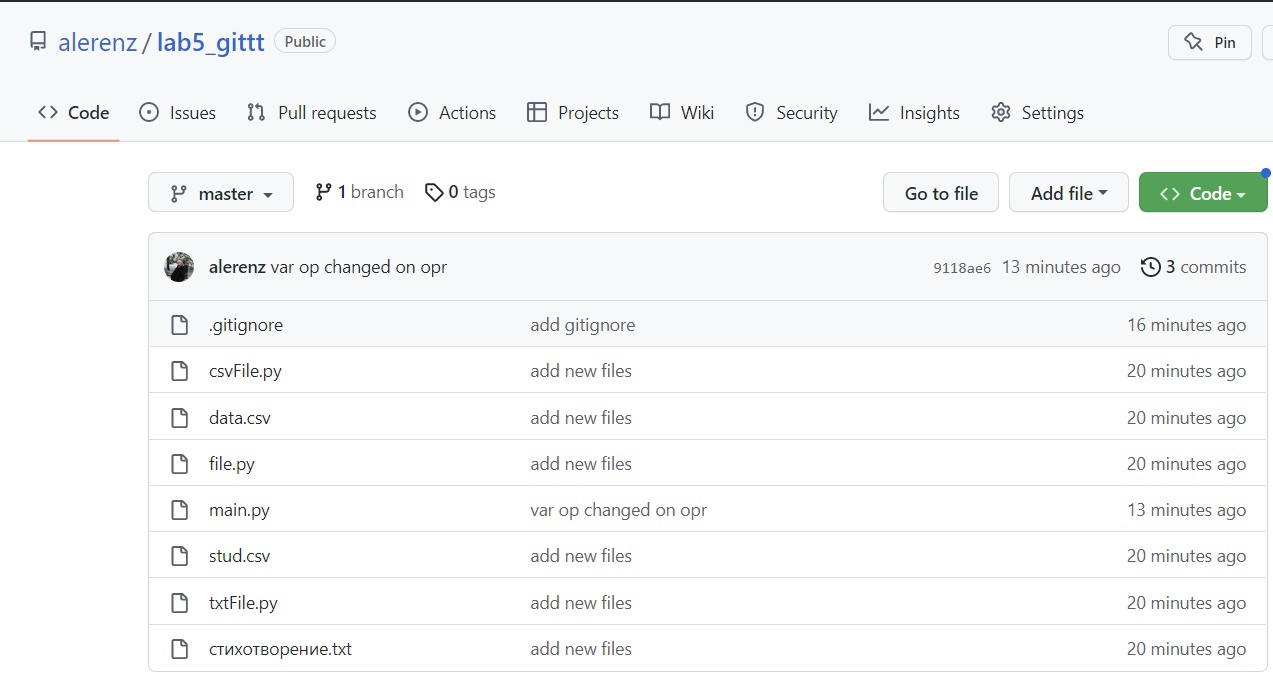
Далее создание репозитория на гитхаб.

Создаем новый репозиторий.

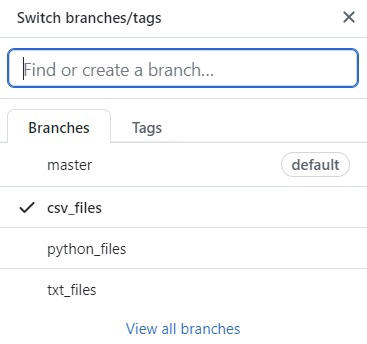
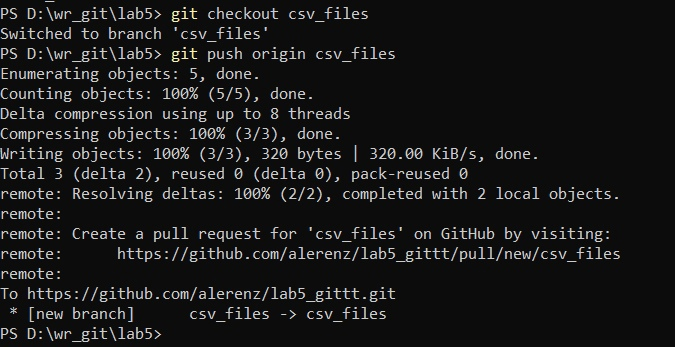


Через консоль подключаемся к репозиторию(**git remote add origin <ссылка на репозиторий>**), ввели в браузере логин и пароль(лично у меня открылся так) и добавляем наши файлы в главную ветку - git push origin master.



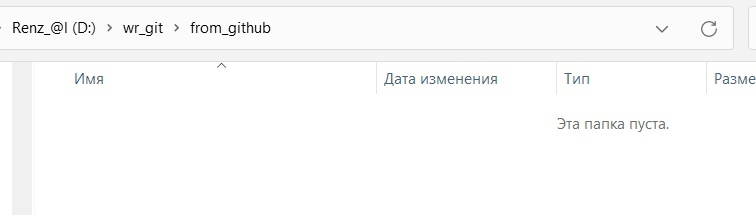


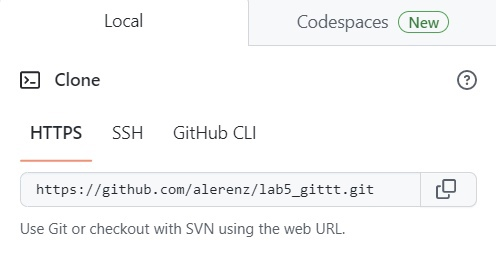
Добавим наши ветки, для этого нужно в них сначала перейти.



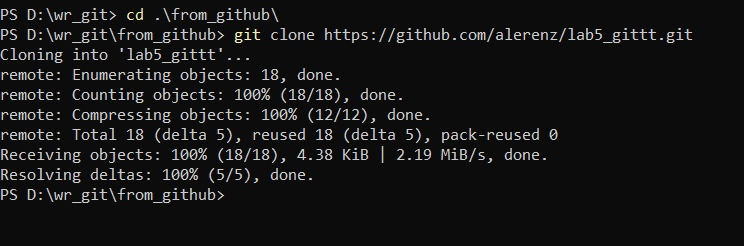
Если посмотреть, то все наши коммиты были показаны.

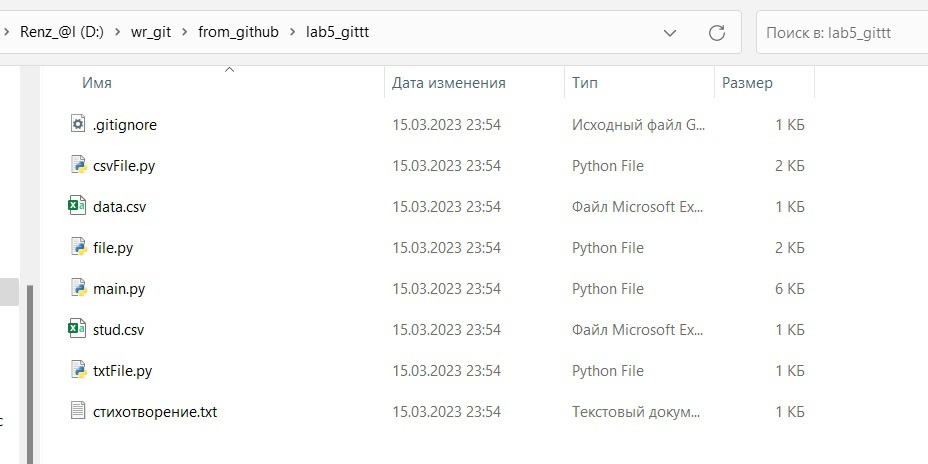
Далее работа с выгрузкой репозитория на локальный компьютер. Создаем пустую папку, отличную от оригинальной.



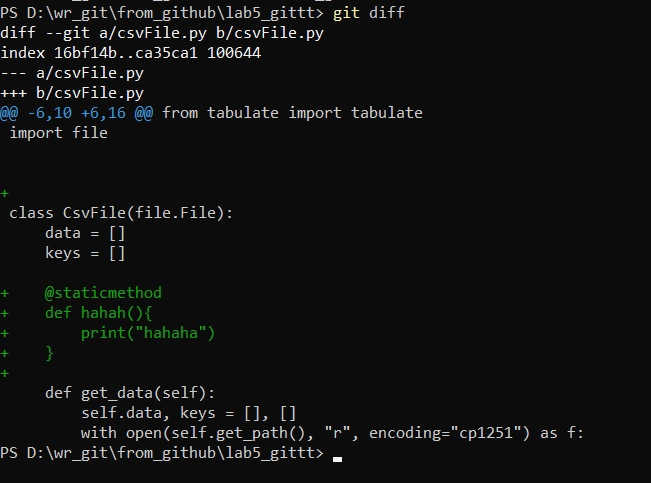
Копируем ссылку репозитория(при условии, что мы создатели репозитория или же мы туда приглашены!)

Переходим в созданную папку и с помощью команды **git clone <ссылка>** выгружаем репозиторий.

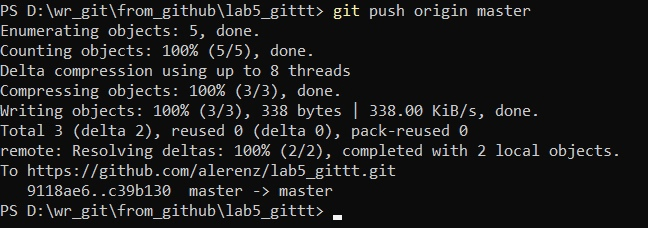


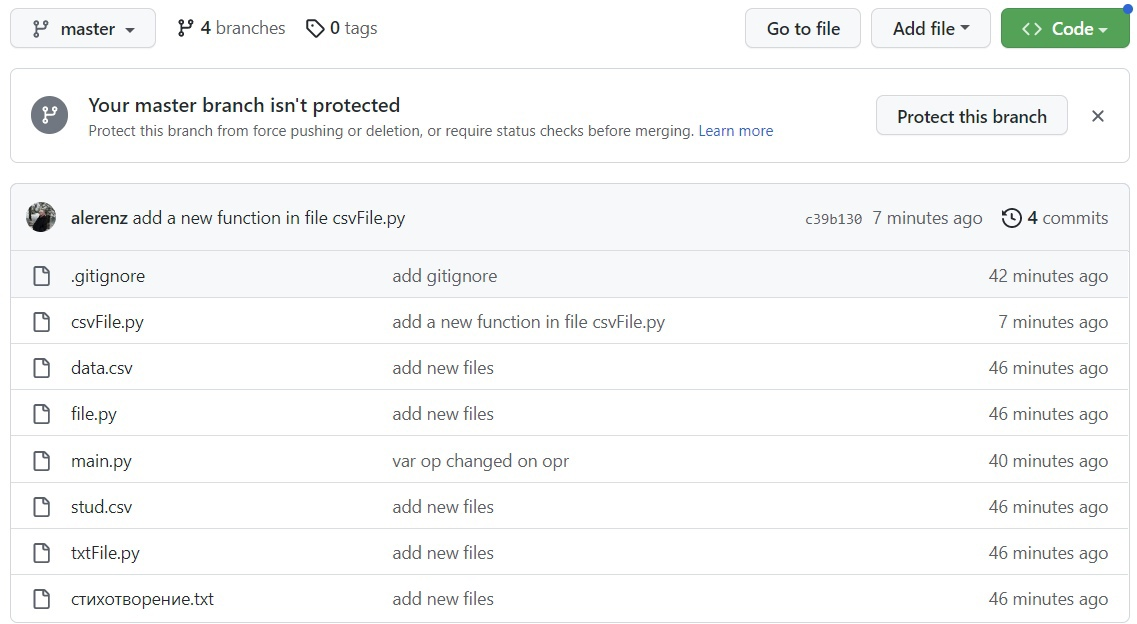


Далее работа с выгруженным репозиторием. Добавим в какой нибудь файл изменения( добавили новую функцию) и посмотрим различия с удаленным репозиторием - команда **git diff**

****

Команда показала в каком файле различия и что именно добавили. Перед тем как добавить измененный файл в репозиторий, добавим его на локальном компьютере(git add), добавим коммит и с помощью команды **git push origin master**(изменяли в главной ветке master) добавим в удаленный репозиторий.





**Описание возникших затруднений:**

При выполнении лабораторной работы, затруднений не возникло.

**Описание альтернативных способ решения:**

Альтернативных способ решения при выполнении лабораторной работы не возникло.

**Вывод о проделанной работе:**

Во время выполнения лабораторной работы, познакомились с git, как с ним работать, какие команды надо использовать, а также новые возможности github.