**Лабораторная работа №1**

**Основы Git и Github**

**Цель работы:**

Выполнение лабораторной работы направлено на изучение:

1. 1. наиболее распространенных практик в области контроля версий

программного обеспечения, его использования в командной разработке ПО и

DevOps;

1. концепции Git, основанной на понятиях репозитория и ветвления

версий ПО;

1. порядка использования GitHub и его базовых операций.

**Порядок работы:**

Зарегистрировался в GitHub, как показано на рисунке 1.

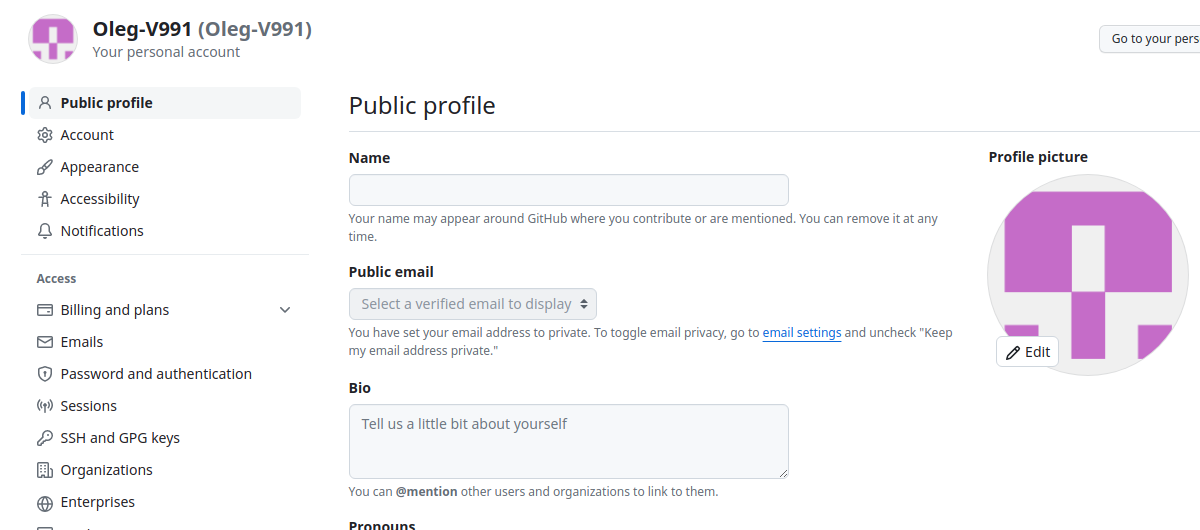


Рисунок 1 - Аккаунт GitHub

2. Создал новый репозиторий: задал имя репозитория, добавил описание, выбрал видимость репозитория «публичный», выбрал опцию «Initialize this repository with a README», затем отредактировал файл README, как показано на рисунке 2.

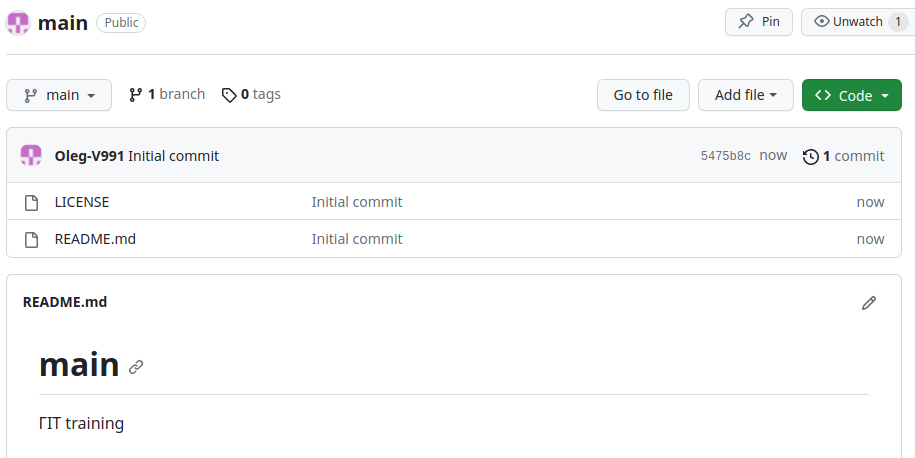
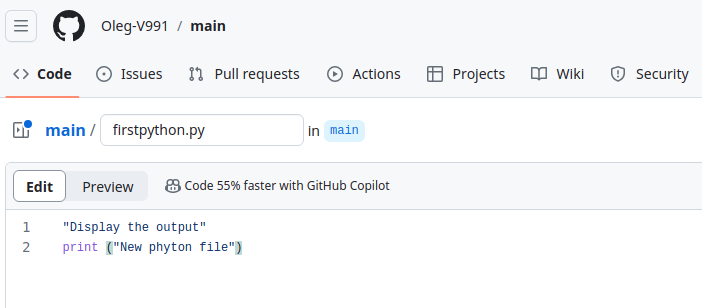


Рисунок 2 - Новый репозиторий

Создал файл firstpython.py с помощью встроенного веб-редактора GitHub. Зафиксировал изменения в репозитории, как показано на рисунке 3.



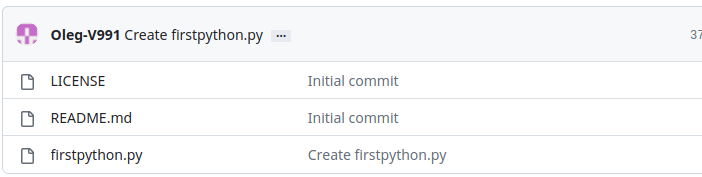


Рисунок 3 - Создание файла

3. Создал новую ветку «student», создал новый файл. Убедился, что файл, добавленный в дочернюю ветку, не добавляется автоматически в основную ветку, как показано на рисунке 4.

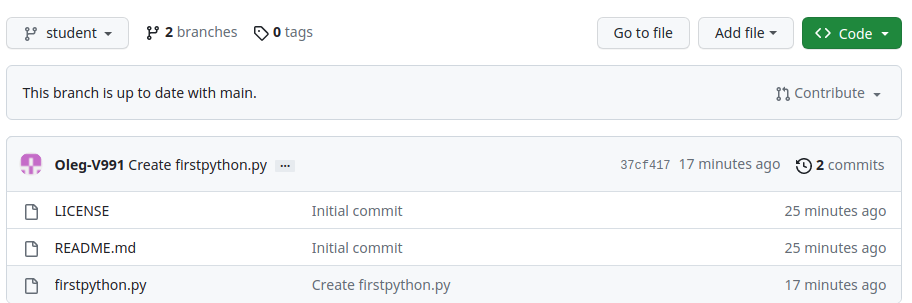


Рисунок 4 - Создание новой ветки

В Child\_Branch нажал кнопку «Compare & pull request», убедился, что в списке указан 1 testchild.py, как показано на рисунке 5.

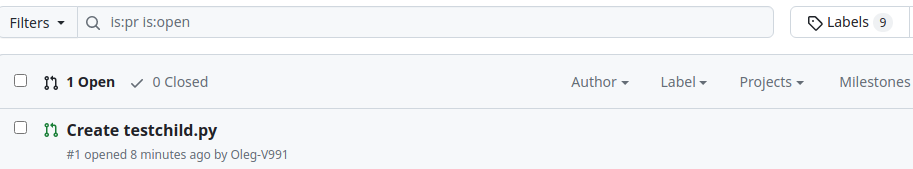


Рисунок 5 - Список измененных файлов

4. Чтобы объединить ветки по запросу pull request в проекте, открыл вкладку «Pull requests». Отображается список ожидающих запросов на включение. Перешел на нужный pull request и нажал «Merge pull request», чтобы принять запрос на включение и объединить обновления, как показано на рисунке 6.

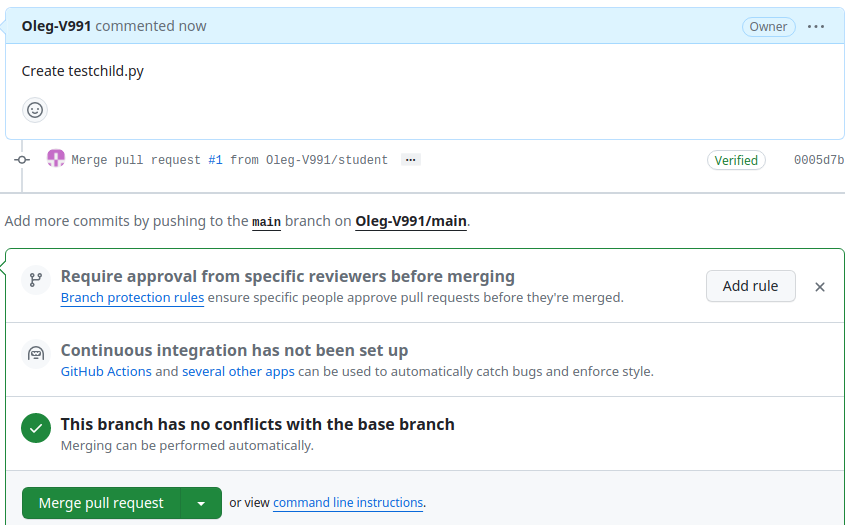
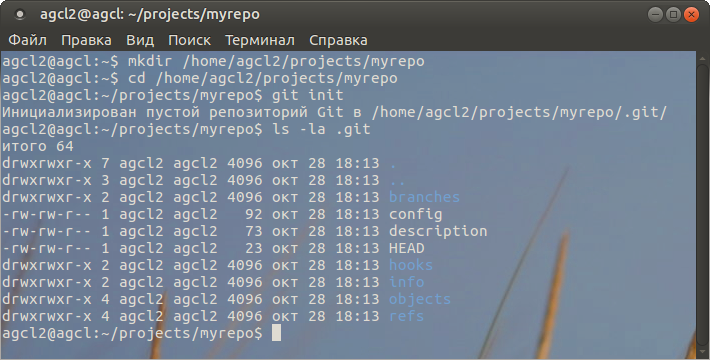


Рисунок 6 - Запрос на включение

5. Далле ввиду отсутствия ПК с windows и по скольку являюсь пользователем Linux

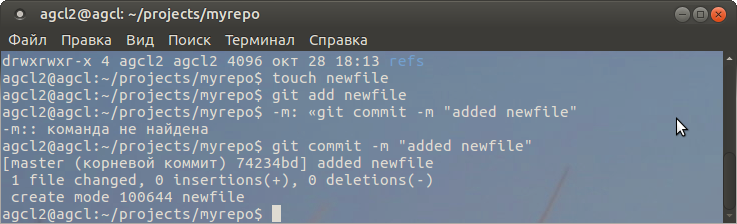
Произвел работу с локальным репозиторием посредством терминала Linux

Создал каталог myrepo и перешел в созданный каталог, затем создал локальный репозиторий, вывел на экран содержимое подкаталога .git, в котором находится локальный репозиторий, как показано на рисунке 7.

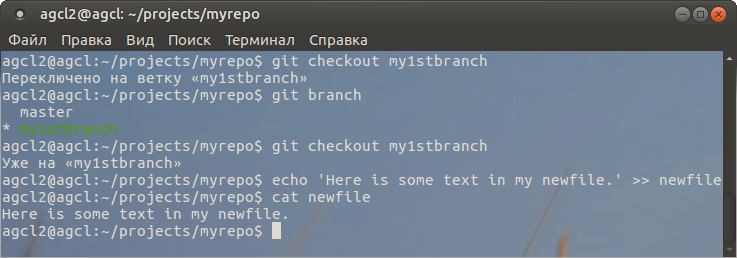
Рисунок 7 - Новый каталог

Создал пустой файл newfile, добавил его в репозиторий. Прежде чем зафиксировать изменения, сообщил git информацию пользователя.

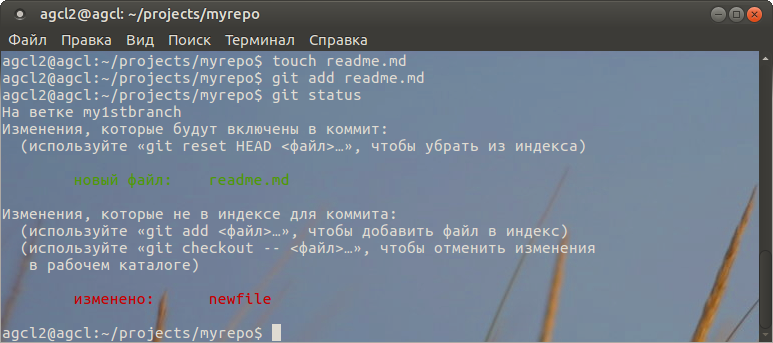
Появившийся в репозитории newfile зафиксировал с добавлением сообщения «added new file», как показано на рисунке 8.

Рисунок 8 - Новый файл

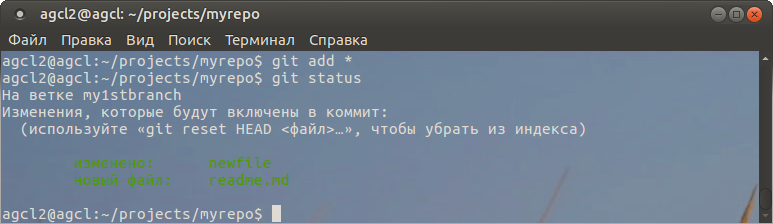
Создал новую ветку под названием my1stbranch, убедился в наличии двух веток в репозитории и переключился с ветки master на ветку my1stbranch, а также внес изменения в newfile путем добавления текста, убедился что текст добавлен, как показано на рисунке 9.

Рисунок 9 - Запись и чтение из файла

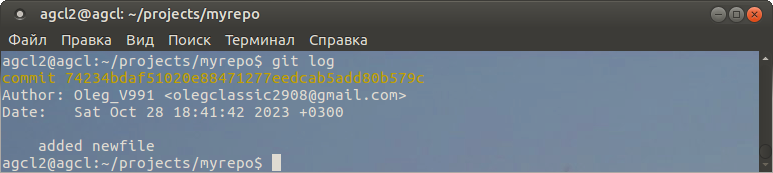
Создал новый файл readme.md и добавил в репозиторий, проверил изменения в текущей ветке my1stbranch, затем добавил newfile явно, как показано на рисунке 10.

Рисунок 1 - Добавление файла

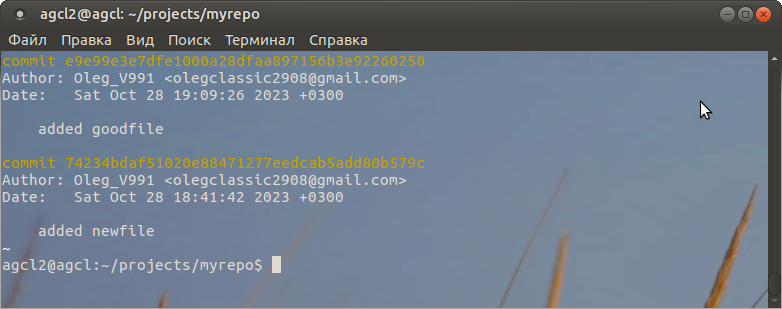
Сохранил изменения в ветку, прикрепив сообщение «added readme.md modified newfile», далее получил историю последних коммитов — последний коммит в my1stbranch, а также предыдущий коммит в master, как показано на рисунке 12.

Рисунок 12 - Сохранение изменений

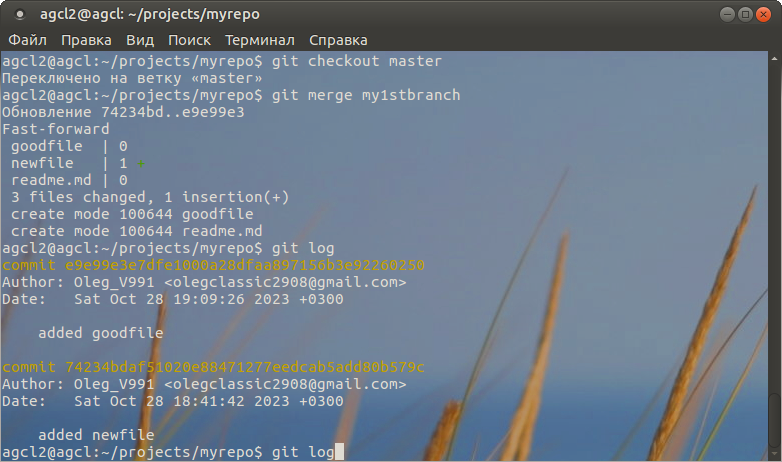
Вывод лога представлен на рисунке 13

Рисунок — 13

Вывод журнала показывает, что недавно добавленный goodfile был зафиксирован в my1stbranch показано на рисунке 14.

Рисунок 14 - Новый файл

Далее объединил содержимое my1stbranch с основной веткой, для этого сначала сделал ветку master активной. Произвел слияние веток и вывел на экран журнал. После того, как слияние успешно завершилось, удалил ветку my1stbranch, как показано на рисунке 15.

Рисунок 15 - Слияние веток и вывод в лог

**Задание №1:**

git branch newbranch // Создание новой ветки под названием newbranch.

git checkout newbranch // Создание новой ветки активной.

touch newbranchfile // Создание пустого файла с именем newbranchfile.

git merge newbranch // Добавлен созданный файл в ветку.

git add newbranch // Зафиксировал изменения в новой ветке.

git commit -m "added newbranch"

git revert HEAD —no-edit // Отменение последних зафиксированных изменений.

touch newgoodfile // Создался новый файл с именем newgoodfile.

git add newgoodfile // Добавилен последний файл в новую ветку.

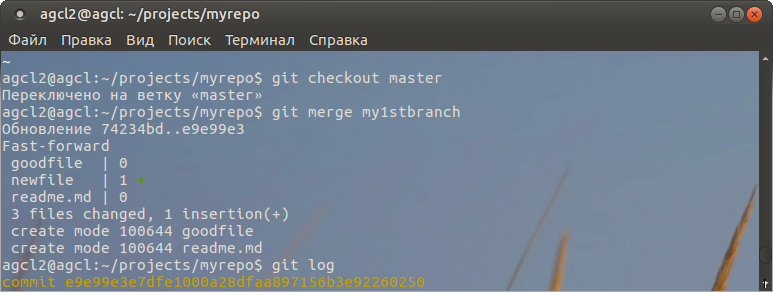
git commit -m "added newgoodfile" // Зафиксированы изменения.

git checkout master

git merge newgoodfile

- Объедините изменения в новой ветке с основной.

Результат выполнения представлен на рисунке 16.

Рисунок - 16