Очёт по лабораторной работе

Лабораторная работа №2

Дикач Анна Олеговна

Содержание

# 1 Цель работы

Изучить идеологию и приминение средств контроля версий, а также освоить умения по работе с git.

# 2 Выполнение лабораторной работы

Так как аккаунт в Github был создан и настроен в прошлом семестре, приступаю к созданию репозитория курса на основе шаблона (рис. 1)

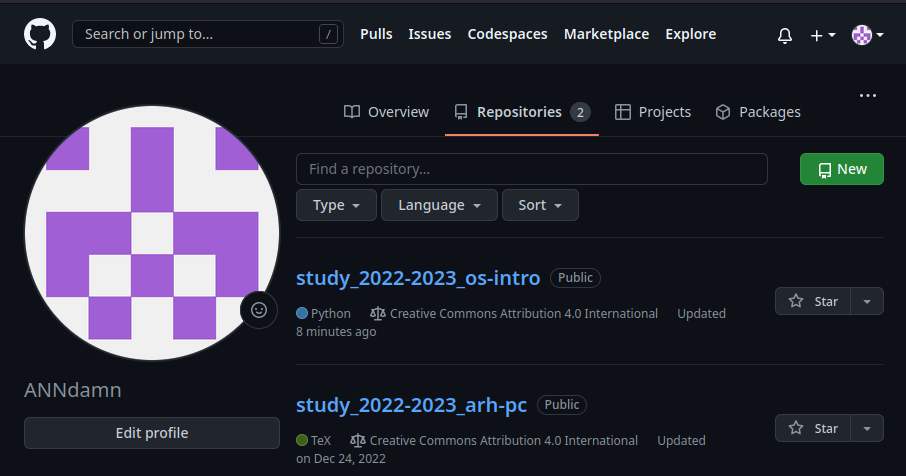


Рис. 1: созданный аккаунт в Github

## 2.1 Создание репозитория курса на основе шаблона

1. Создаю шаблон рабочего пространства (рис. 2) (рис. 3)

Рис. 2: создание репозитория курса

Рис. 2: создание репозитория курса



Рис. 3: создание репозитория курса

## 2.2 Настройка каталога курса

1. Перехожу в каталог курса (рис. 4)

Рис. 4: команда для перехода в каталог курса

Рис. 4: команда для перехода в каталог курса

1. Удаляю лишние файлы (рис. 5)

Рис. 5: команда для удаления лишних файлов

Рис. 5: команда для удаления лишних файлов

1. Создаю необходимые каталоги (рис. 6)

Рис. 6: команда для создания каталогов

Рис. 6: команда для создания каталогов

1. Отправляю файлы на сервер (рис. 7)

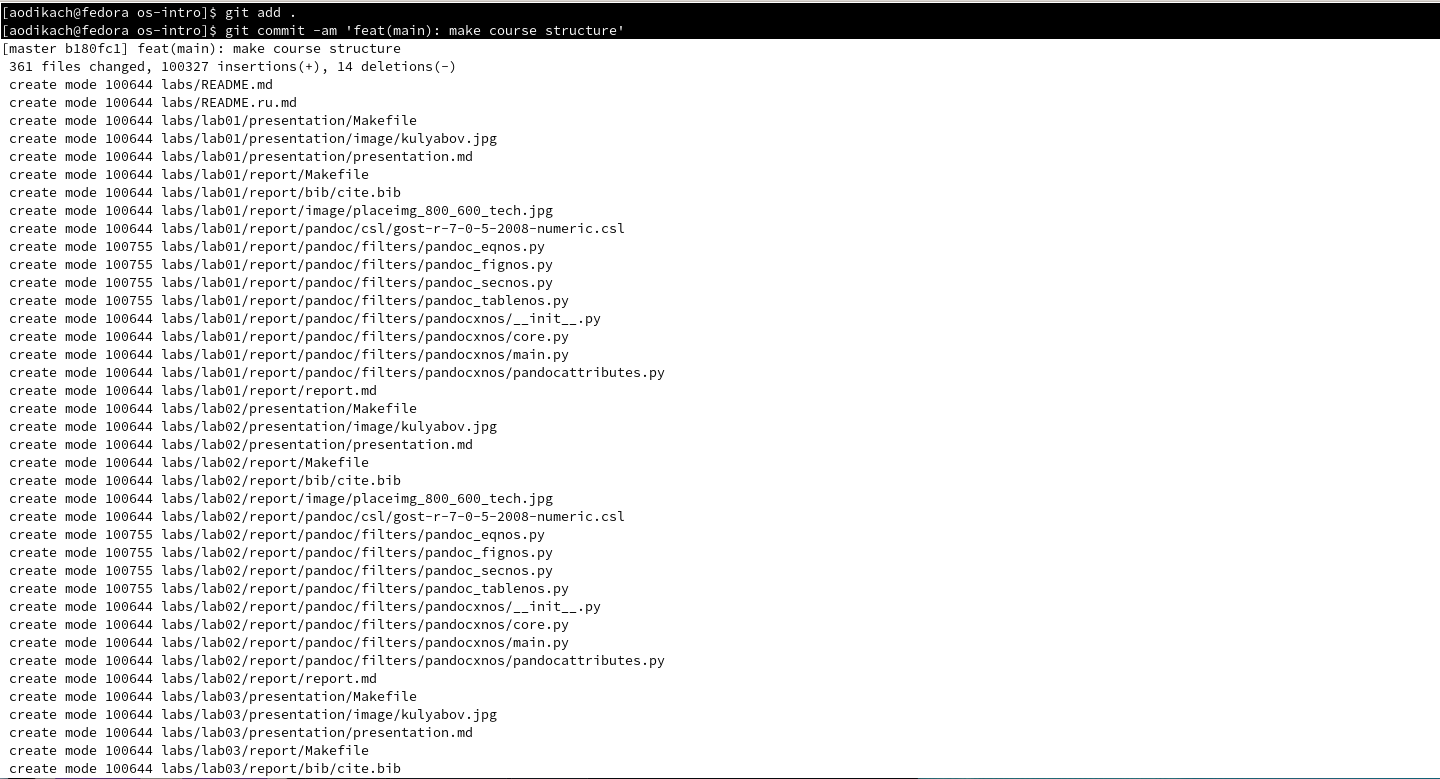


Рис. 7: команда для отправки файлов на сервер

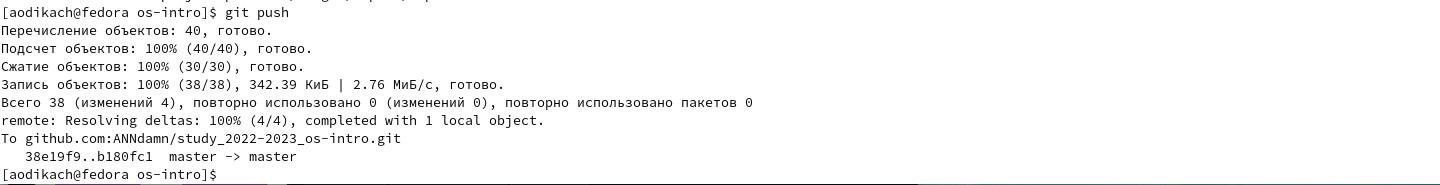


Рис. 8: команда для отправки файлов на сервер

1. Итог проведённой работы (рис. 9)

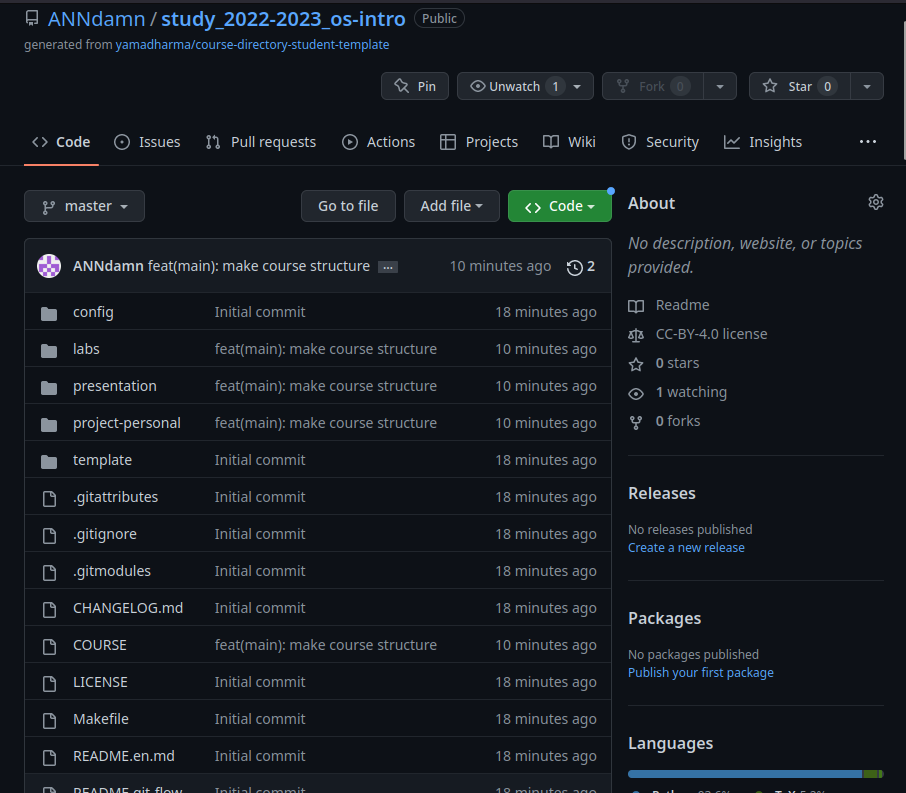


Рис. 9: скринкаст проведённой работы

## 2.3 Контрольные вопросы

1. Системы контроля версий (VCS) применяются при работе нескольких человек над одним проектом.  
   Система управления версиями позволяет хранить версии одного и того же документа, возвращаться к ним и следить за изменениями и фиксировать их.
2. хранилище - место хранения файлов с которым могут взаимодействовать все участники commit - делает для проекта снимок текущего состояния изменений, добавленных в раздел проиндексированных файлов. история - показывает операции, которые были произведены с объектом. имеет большое количество опций для поиска коммитов по разным критериям рабочая копия - является снимком одной версии проекта. файлы извлекаются из сжатой базы данных в каталоге Git и помещаются на диск для того чтобы их могли редактировать другие пользователи
3. Централизованные системы контроля версий представляют собой приложения типа клиент-сервер, когда репозиторий проекта существует в единственном экземпляре и хранится на сервере. Доступ к нему осуществлялся через специальное клиентское приложение. Примеры: CVS, Subversion Децентрализированные системы контроля версий (Distributed Version Control System, DVCS) позволяют хранить репозиторий (его копию) у каждого разработчика, работающего с данной системой. При этом можно выделить центральный репозиторий (условно), в который будут отправляться изменения из локальных и, с ним же эти локальные репозитории будут синхронизироваться. Примеры: Git и Mercurial
4. создания и работа с файлами, выгрузка их в хранилище. не требует создания рабочих копий. можно сделать проект конфедициальным, не доступным для других пользователей git.
5. создание репозитория, обновить проект, скопировать последнюю версию, внести изменения, разрешить конфликты, создать ветку.
6. основными задачами являются: проверка ветки, переключение на новую, написание кода, просмотр статуса, создание коммита, отправка
7. git add - добавляет содержимое рабочего каталога в индекс git status - показывает состояние файлов git diff - используется для вычисления разницы между git difftool - запускает внешнюю утилиту сравнения для показа различий двух деревьев git commit - берёт все данные, добавленные в индекс с помощью git add, и сохраняет их слепок во внутренней базе данных, а потом сдвигает указатель текущей ветки на этот слепок git reset - используется для отмены изменений git rm - используется в Git для удаления файлов из индекса и рабочей копии git mv - перемещает файлы, добавляет содержимое рабочего каталога в индекс нового файла и и удаляет старый git clean - удаляет мусор их рабочего каталога
8. с помощью git add пользователь изменяет файл, проверяет состояние других файлов с помощью git status, с помощью git difftool сверяет проделанную работу с работой своего партнёра, с помощью git rm удаляет рабочую копию своего партнёра.
9. ветви нужны для того чтобы программисты могли вести совместную работу и при этом не мешать друг другу своими действиями
10. некоторые файлы игнорируются для их исключения из истории git.

# 3 Выводы

Создала аккаунт в Github, изучила его идеологию и применение средств контроля версий.