**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский технический университет связи и информатики»**

Кафедра «Математическая кибернетика и информационные технологии»

**Отчет по лабораторной работе №0.2 «Методы ветвления в системе контроля версий»**

# по дисциплине «Информационные технологии и программирование»

Выполнил: студент группы

БВТ2105

Канев Демид Александрович

Руководитель:

Мосева Марина Сергеевна

Москва 2022**Цель работы**

изучение подходов к командной разработке в системе контроля версий git

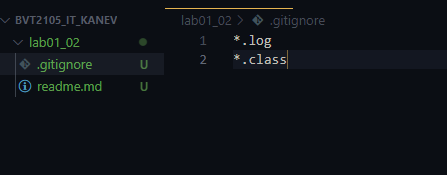
**Задание**

В созданном в предыдущей работе либо вновь созданном репозитории:

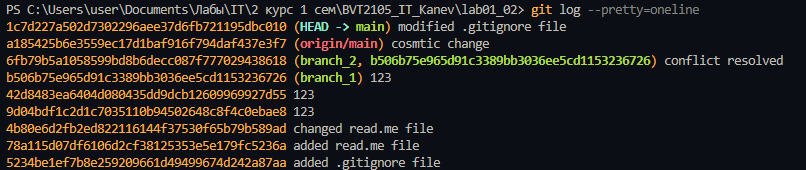
1. Настроить игнорирование git’ом файлов определенного расширения с использованием glob-шаблонов
2. Вывести лог коммитов в измененном пользовательском формате
3. Переименовать ветку
4. Провести в ветке работу и слить с веткой master, удалить слитую ветку
5. Настроить удалённый репозиторий: получить изменения из репозитория и отправить изменения.
6. Организовать совместную работу над проектом в соответствии с выбранной идеологией

**Ход работы**

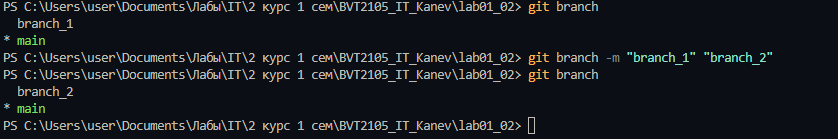
1. Настроим игнорированием файлов с помощью файла .gitignore, с расширением .class и .log, с помощью glob-шаблона \* мы сможем игнорировать все файлы, которые имеют такие определенные расширения.

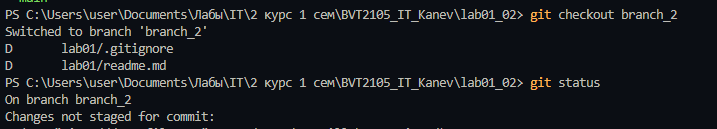


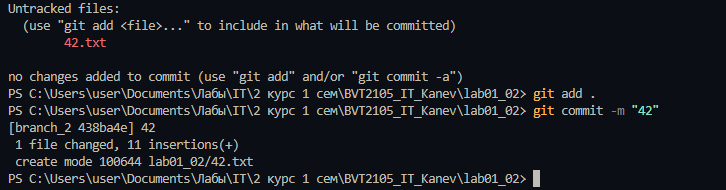
2. Выведем лог коммитов в измененном пользовательском формате. Для этого, например, добавим к команде git log параметр –pretty, который и позволяет сделать нам пользовательский формат, для этого возьмем имеющийся шаблон oneline, который выведет все коммиты в одну строку



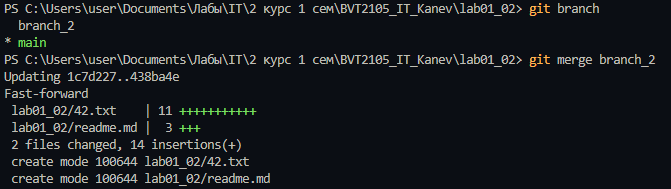
3. Находясь в главной ветке, переименуем созданную в первой работе ветку с помощью команды git branch и параметра –m, как первый аргумент передадим предыдущее название, а следующим – новое



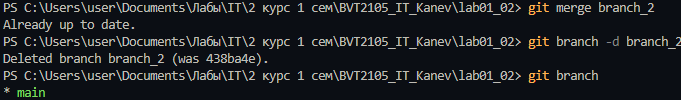
4. Переключимся на новую ветку, создадим там файл



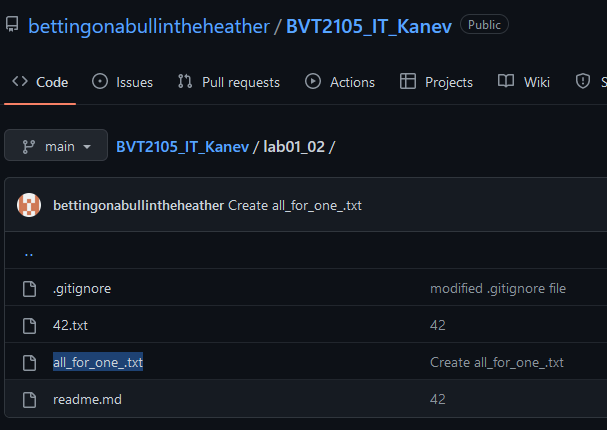
Вернемся на основную ветку и попробуем произвести слияние с помощью команды git merge



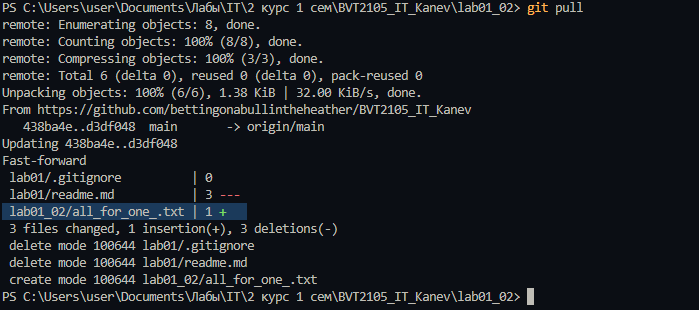
Удалим слившуюся ветку с помощью команды git branch и параметра –d



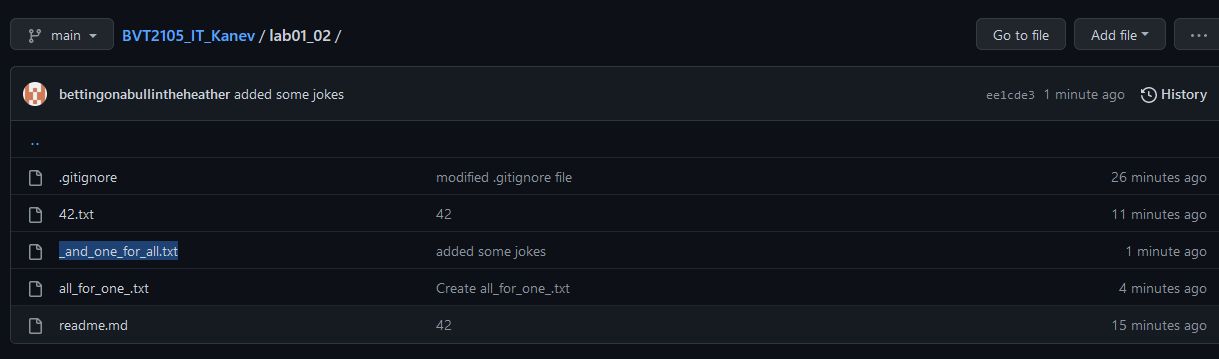
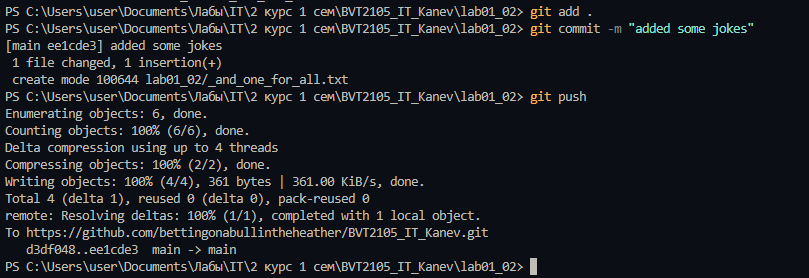
5. В удаленном репозитории, из которого осуществлялось клонирование, был удаленно создан новый файл all\_for\_one\_.txt

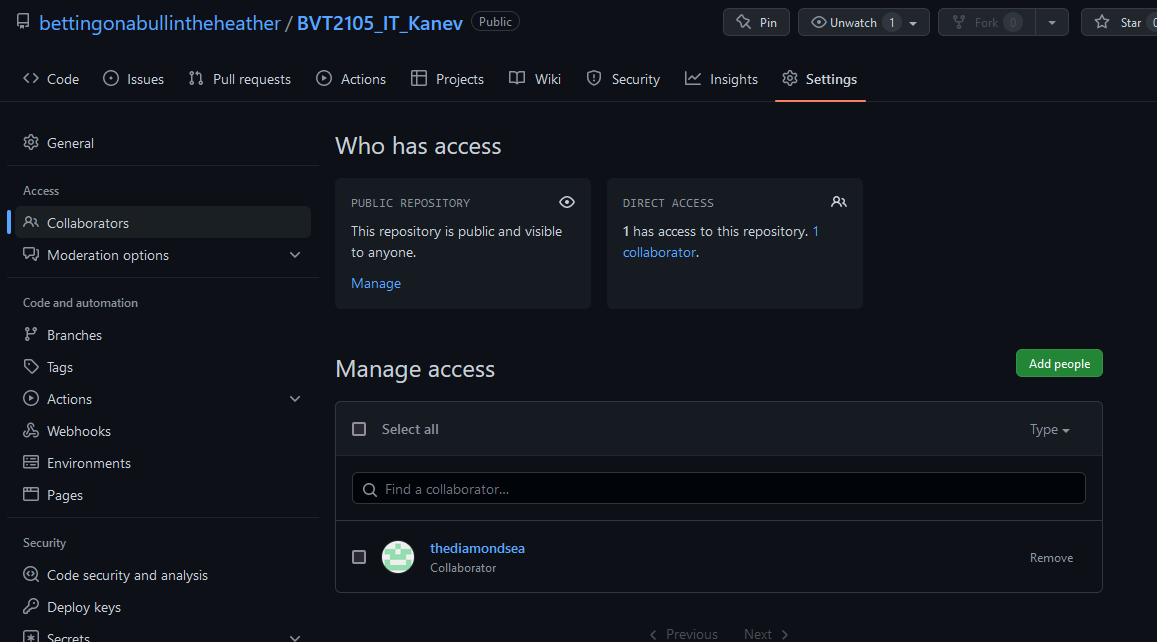


Получим из него изменения в наш локальный репозиторий с помощью команды git pull



Для отправки изменений создадим новый файл \_and\_one\_for\_all.txt и отправим изменения в удаленный репозиторий с помощью команды git push



6. Для совместной работы, к репозиторию на сайте github был добавлен новый участник thediamondsea

С этого момента второй участник может создать новую ветку, в которой будет проводить свои изменения проекта. На сайте гитхаб, можно будет отследить изменения проекта участниками

**Вывод**

В результате проделанной работы, мы изучили подходы к командной разработке в системе контроля версий git. Для чего использовали методы ветвления, настроили игнорирование, научились выводить логи коммитов и работать с ветками, после чего объединились с другим участником и произвели совместную работу.

**Литература**

1. Камаев В.А., Костерин В.В. Технологии программирования. М.: Высшая школа,

2006.

2. Жоголев Е.А.Технология программирования. – М.: Научный мир, 2004.

**Ссылка на репозиторий: https://github.com/SanctumInvictum/Java-repository**