**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ**

**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

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

Кафедра «Математическая Кибернетика и Информационные технологии»

Лабораторная работа

Основы работы с системой контроля версий

Выполнил: Студент группы

БВТ2201

Овчинников Дмитрий

Москва

2023

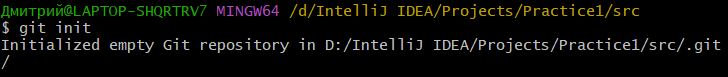
**Цель работы:** изучение базовых функций и подходов к командной разработке в системе контроля версий Git.

**Ход работы:**

1) Создание репозитория или клонирование существующего.

·git init – создание нового локального репозитория;

·git clone – клонирование репозитория.

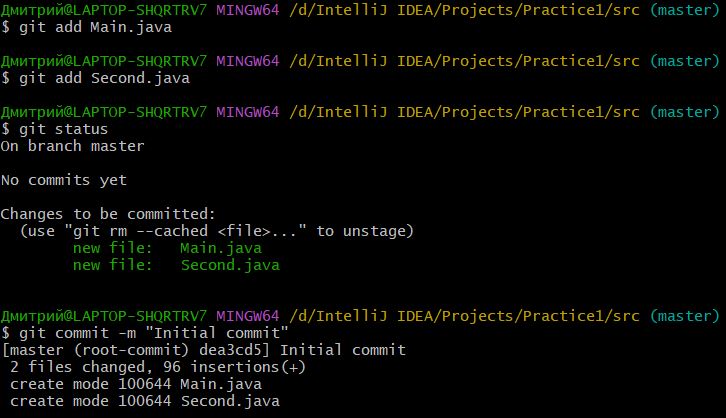


2) Внесение ряда изменений в файл и выполнение коммитов.

·git add – добавление файлов в индекс;

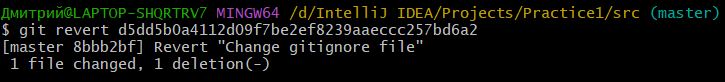
·git status – проверка списка файлов (в индексе или нет);

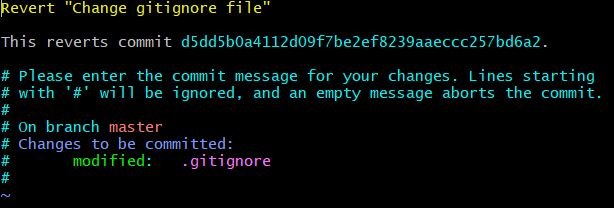
·git commit –m – коммит добавленных в индекс файлов с сообщением.



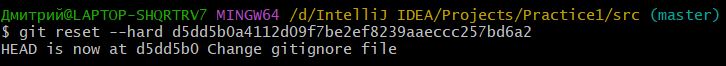
3) Откат изменений до одного из предыдущих коммитов двумя разными способами.

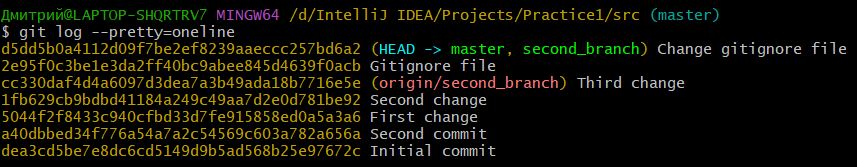
·git revert – создание нового коммита, который отменяет изменения, сделанные в указанном коммите;





·git reset --hard – удаление всех коммитов со всеми изменениями до указанного коммита.





4) Создание, переименование и удаление ветки.

·git branch – список всех локальных веток;

·git checkout –b – создание ветки и переключение на неё;

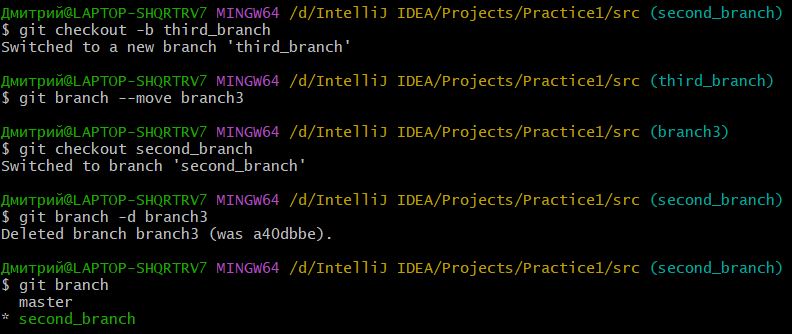
·git branch --move – переименование текущей ветки;

·git checkout – перемещение на другую ветку;

·git branch –d – удаление ветки.

C:\Users\Дмитрий\Desktop\Лаба 1\4 (1).JPG

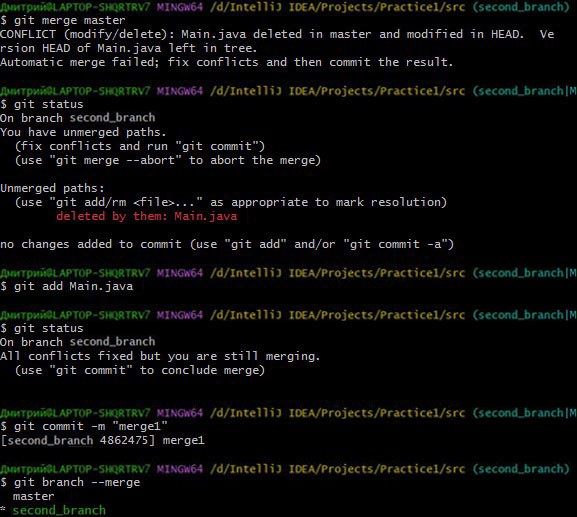
C:\Users\Дмитрий\Desktop\Лаба 1\4 (2).JPG



5) Слияние ветки с веткой мастер с решением конфликта содержания файлов.

·git merge – объединение веток;

·git branch --merge – вывод веток, которые сливались.



6) Настройка игнорирования git’ом файлов определенного расширения с использованием glob-шаблонов.

·touch – создание файла (чтобы игнорировать файлы с определённым расширением, надо добавить \* перед точкой и названием расширения в файле .gitignore);

·cat – показывает содержимое объекта репозитория.

C:\Users\Дмитрий\Desktop\Лаба 1\6 (1).JPG

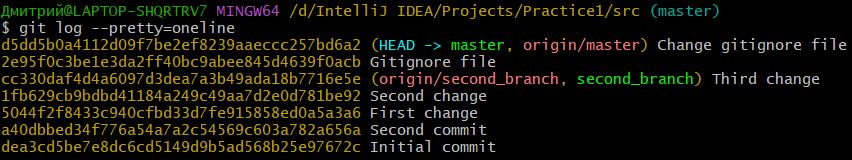
C:\Users\Дмитрий\Desktop\Лаба 1\6 (3).JPG

7) Вывод лога коммитов в измененном пользовательском формате.

·git log – список коммитов;

·git log --pretty – смена формата вывода списка коммитов;

·git log --pretty=oneline – вывод каждого коммита в одну строку.

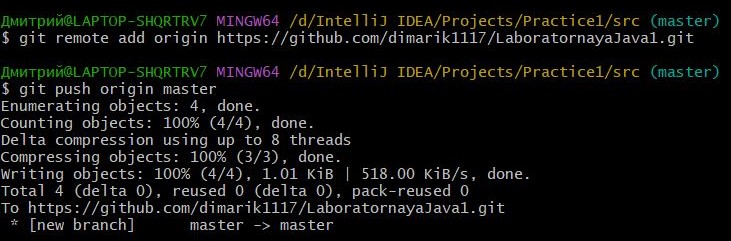


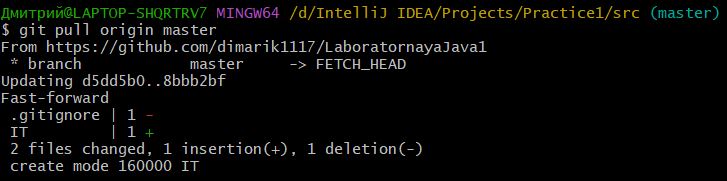
8) Настройка удалённого репозитория: получение и отправка изменений в репозиторий.

·git remote add origin – добавление адреса удалённого репозитория;

·git push origin – отправление локальных коммитов в удалённый репозиторий;

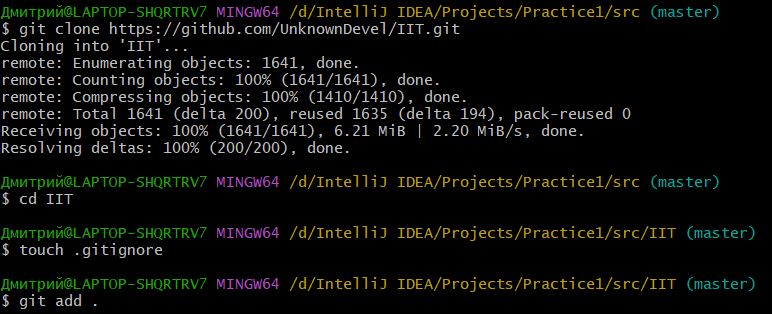
·git pull origin – скачивание изменений из удалённого репозитория.

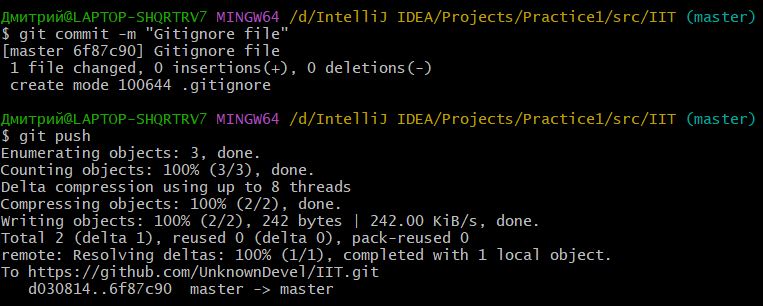


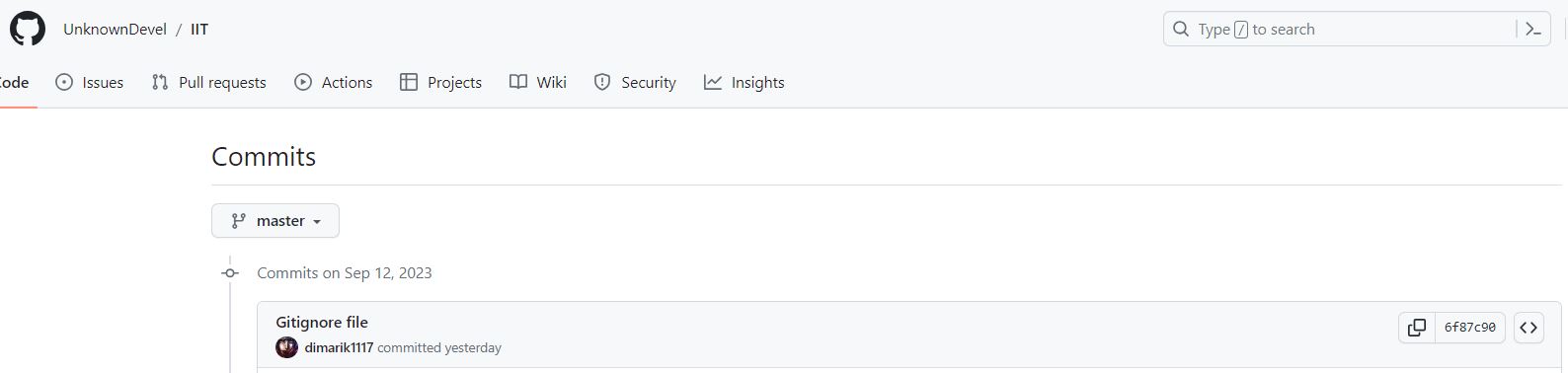


9) Организация совместной работы над проектом: клонирование чужого репозитория, выполнение коммитов, отправка изменений.

·git clone – клонирование чужого репозитория.







**Вывод:** проделав работу, я изучил базовые функции и подходы к командной разработке в системе контроля версий Git.

Репозиторий: <https://github.com/dimarik1117/LaboratornayaJava1>

Чужой репозиторий: <https://github.com/UnknownDevel/IIT>