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"Системное программирование"

Лабораторная работа № 0

Основы работы с системой контроля версий по дисциплине

“Информационные технологии и программирование”

Выполнил: студент гр. БВТ2201

Шамсутдинов Р.Ф.

Москва, 2023 г

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

Задание: определить проект (состоящий из нескольких файлов), который будет добавлен под версионный контроль, и произвести ряд операций в СКВ, включающих: 1) Создание репозитория или клонирование существующего

1. Внесение ряда изменений в файл(ы) и выполнение коммитов (кол-во не меньше 2)
2. Откат изменений до одного из предыдущих коммитов двумя разными способами
3. Создание, переименование и удаление ветки
4. Слияние ветки с веткой мастер с решением конфликта содержания файлов6) Настройка игнорирования git’ом файлов определенного расширения с использованием globшаблонов

7) Вывод лога коммитов в измененном пользовательском формате 8) Настройка удалённого репозитория: получение и отправка изменений в репозиторий

9) Организация совместной работы над проектом: клонирование чужого репозитория, выполнение коммитов, отправка изменений Перечень основных используемых команд ($ git …): init

add commit

status log diff

branch checkout

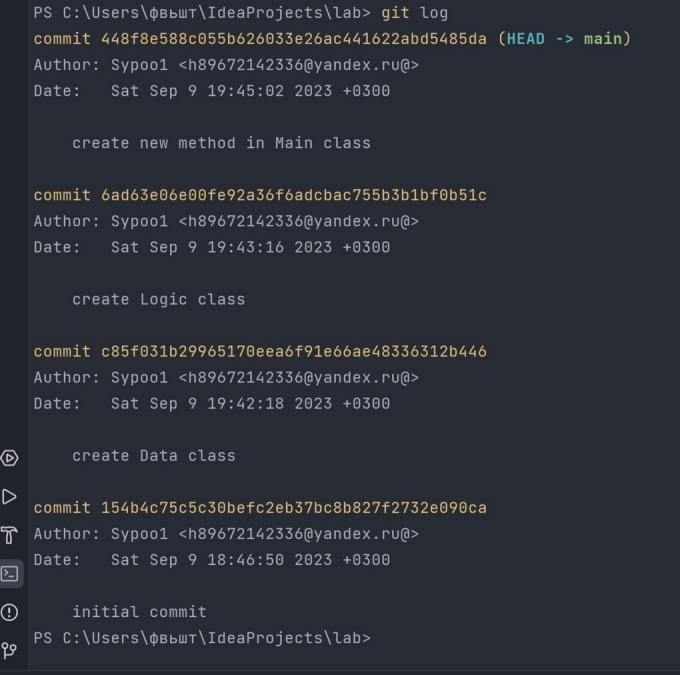
merge reset clone fetch pull push

Ход работы:

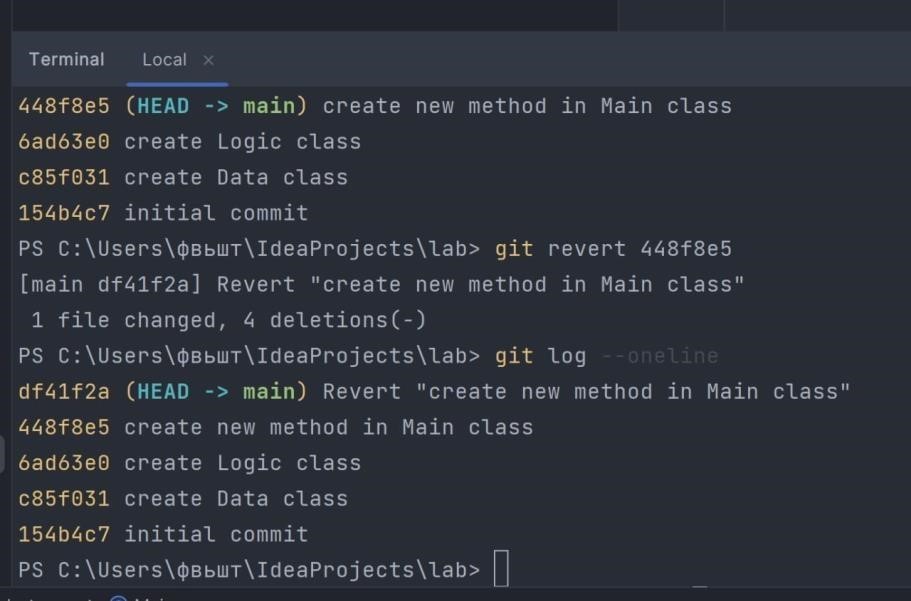
1. Создание репозитория или клонирование существующего git init (инициализируем гит репозиторий)
2. Внесение ряда изменений в файл(ы) и выполнение коммитов

git status (смотрим статус у отслеживаемых файлов) git add . (добавляем все отслеживаемые файлы в индекс)

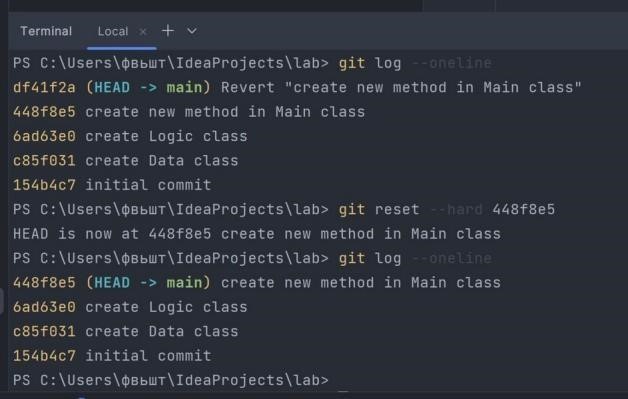
git commit -m “описание коммита” (создаем коммит из всех файлов, содержащихся в индексе) git log (смотрим историю коммитов)



1. Откат изменений до одного из предыдущих коммитов двумя разными способами
2. git revert 448f8e5 (создает новый коммит, который отменяет изменения, сделанные в указанном коммите)

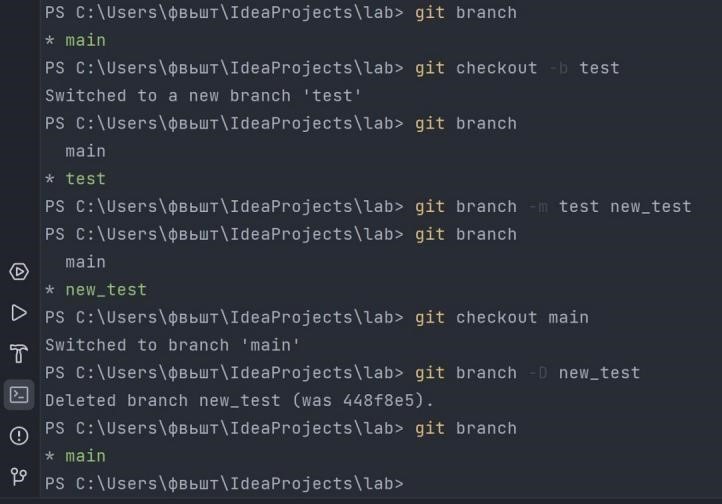


1. git reset --hard 448f8e5 (удаляем все коммиты до указанного, а также все изменения с помощью флага --hard)



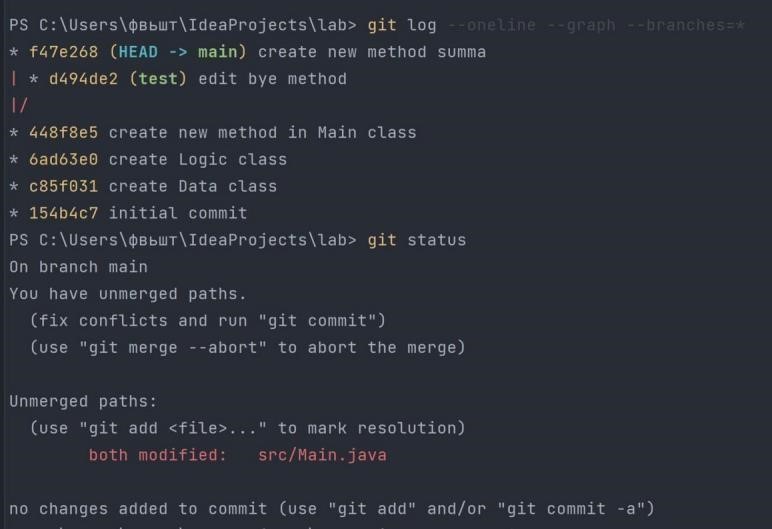
4)Создание, переименование и удаление ветки

git branch (смотрим какие есть ветки) git checkout -b test (создаем ветку test и переходим на неё) git branch -m test new\_test (меняем имя ветки test на new\_test) git checkout main (переходим на ветку main) git branch -D new\_test (удаляем ветку new\_test)

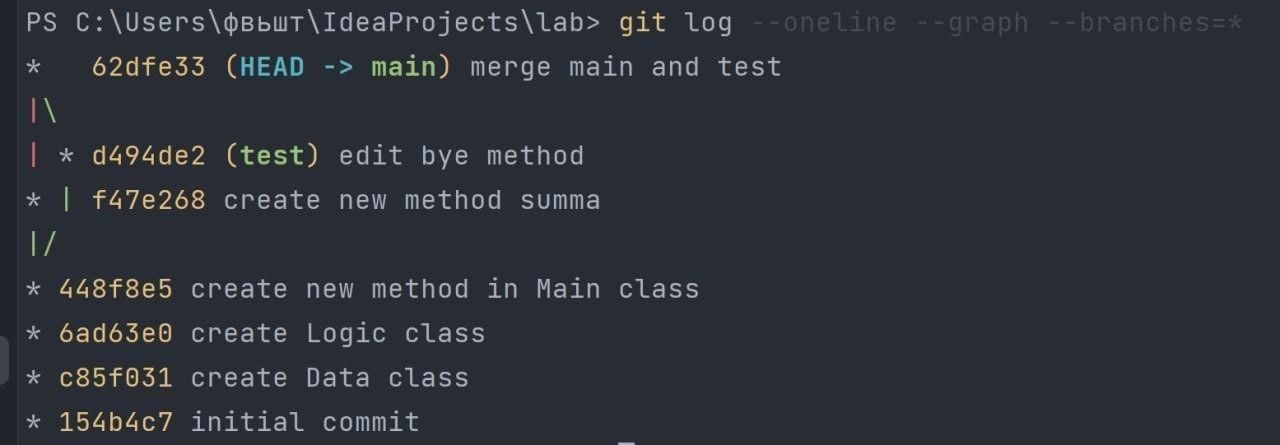


1. Слияние ветки с веткой мастер с решением конфликта содержания файлов git merge test

(слияние ветки test с ветко main) git status (видим что есть конфлик в файле src/Main.java)

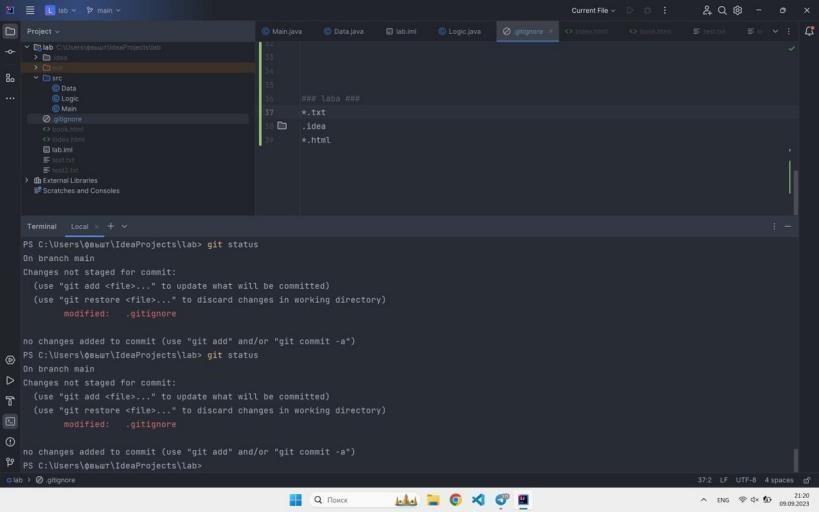


решаем конфлик путем редактирования нужного файла, делаем коммит



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

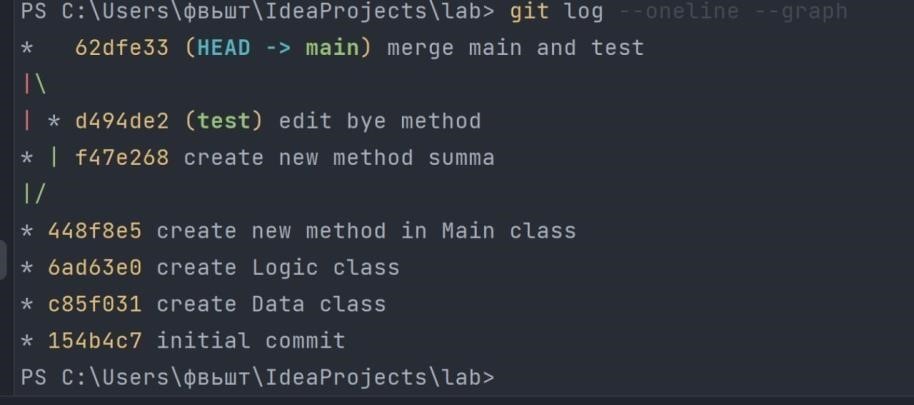
В файл .gitignore добавляем имена файлов, которые не хотим отслеживать. Для игнорирования файлов с определенным расширением, добавляем \* перед .названием расширения



1. Вывод лога коммитов в измененном пользовательском формате git log --oneline --graph

(выводим все коммиты, флаг --oneline выводит информацию о каждом коммите в одну строку а также сокращает хеш коммита до 7 символов, флаг --graph

отображает граф с ветвлениями и слияниями)



1. Настройка удалённого репозитория: получение и отправка изменений в репозиторийgit remote add origin [https://github.com/Sypoo1/mtuci-java-labs.git (](https://github.com/Sypoo1/mtuci-java-labs.git)добавляем удаленный репозиторий с сокращенным именем origin)

git remote -v (выводим доступные удаленные репозитории) git push origin

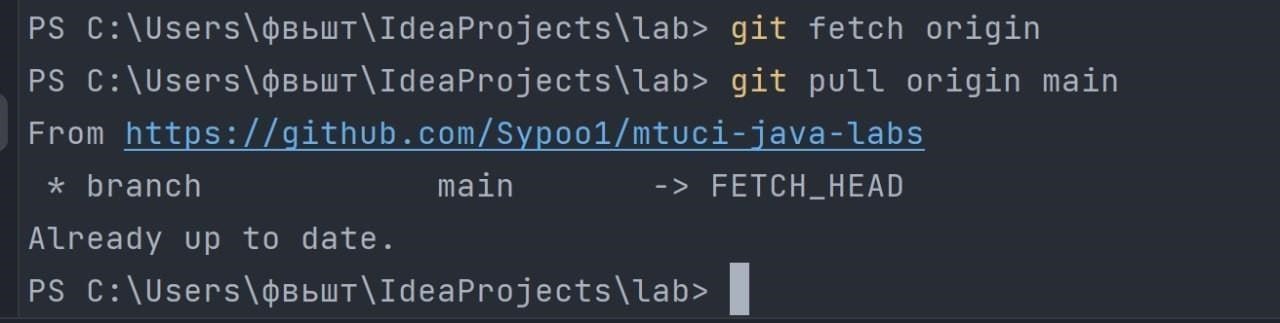
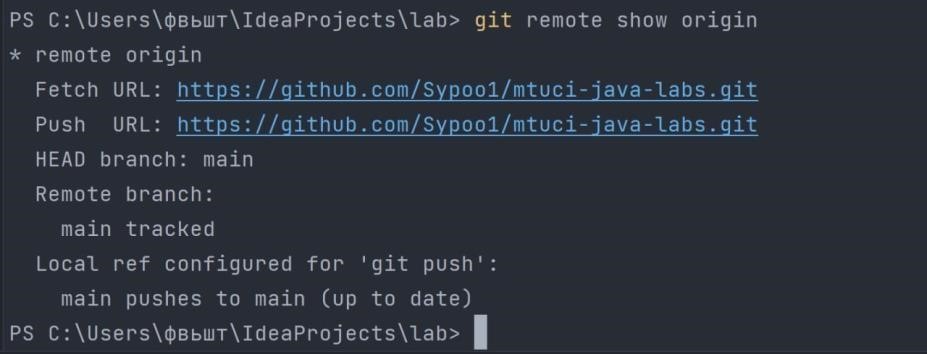
main (отправляем последний коммит ветки main на удаленный

репозиторий) git remote show origin (выводим информацию об удаленном репозитории)

git fetch origin (забирает последнюю версию с удаленного репозитозрия)

git pull origin main (делает то же самое, что и git fetch origin, но также бытается слить изменения в ветку main)



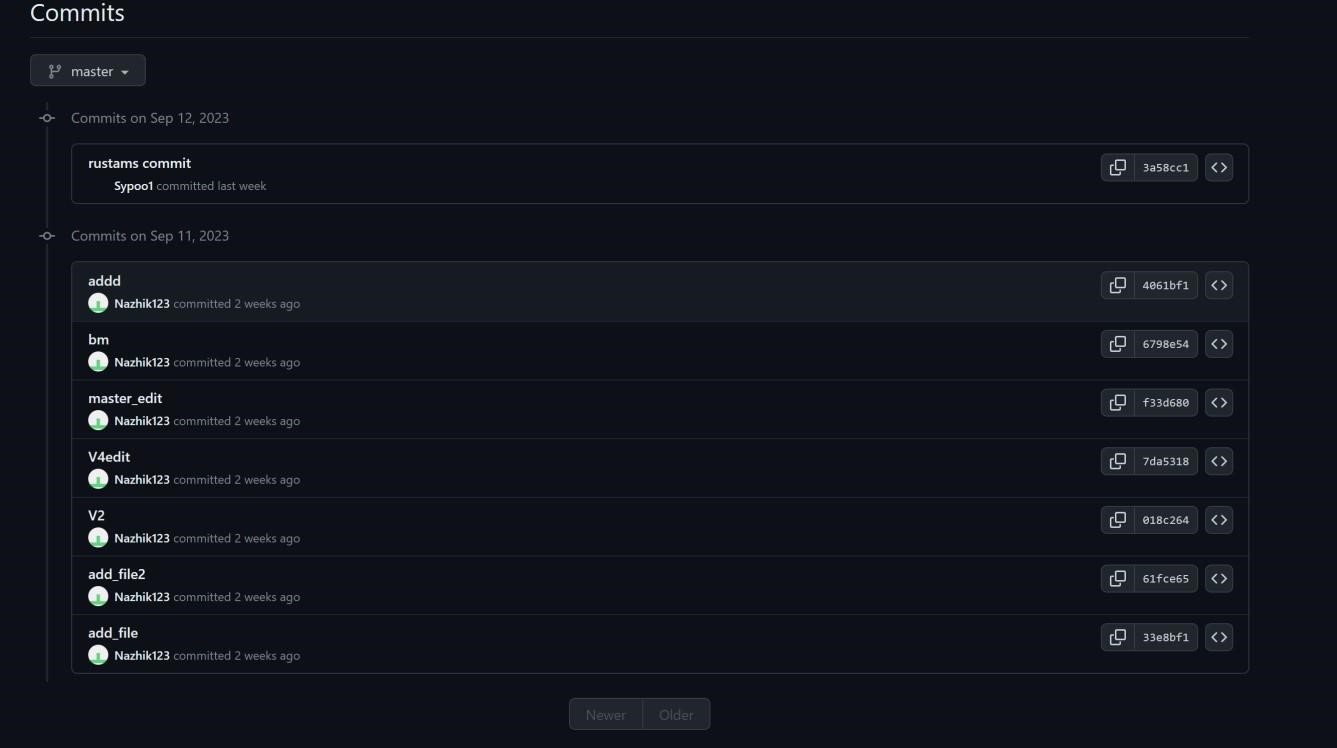


1. Организация совместной работы над проектом: клонирование чужого репозитория, выполнение коммитов, отправка изменений

git clone https://github.com/Nazhik123/inf\_laba.git (клонируем проект из удаленного

репозитория) вносим изменения, создаем коммит, отправляем изменения на

удаленный репозиторий



Вывод: проделав данную работу, я познакомился с базовыми командами контроля версий Git, научился отслеживать версии проекта, а также взаимодействовать с удаленными репозиториями на GitHub.

Ссылка на репозиторий: https://github.com/Sypoo1/mtuci-java-labs/tree/main/nullLab