**Федеральное агентство связи**

**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение**

**Высшего образования**

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

Кафедра Математической кибернетики и информационных технологий

**Отчет по лабораторной работе № 0.2**

По дисциплине «Объектно-Ориентированное-Программирование»

На тему:

«Методы ветвления в системе контроля версий»

Выполнил:

студент группы БВТ

Москва 2020

**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.

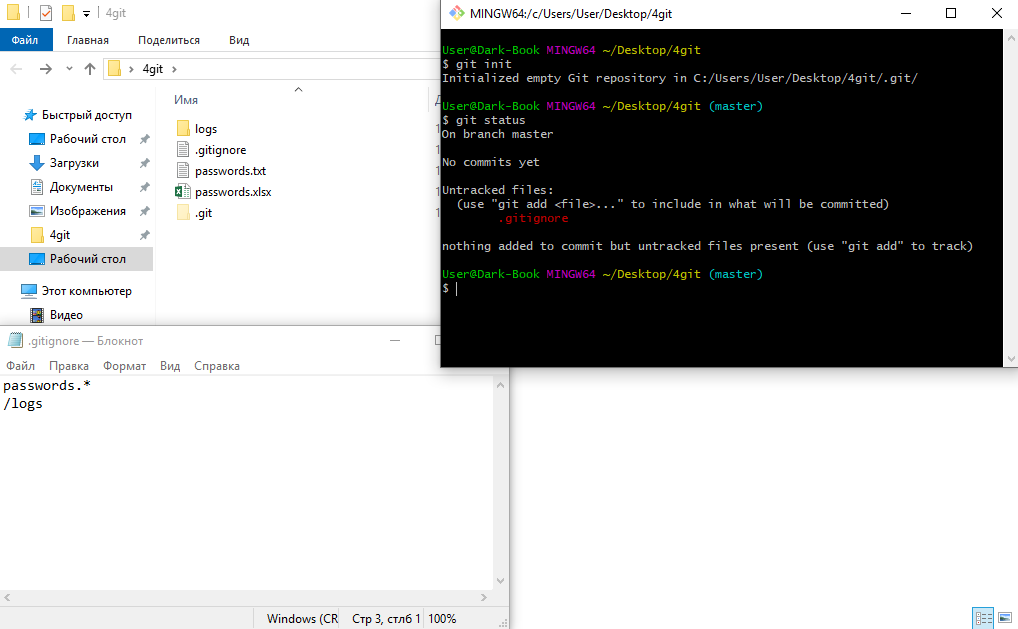
5. Настроить удалённый репозиторий: получить изменения из репозитория и отправить изменения.

**Ход работы:**

1. **Настроить игнорирование git’ом файлов определенного расширения с использованием glob-шаблонов.**

Для начала необходимо создать папку с инициализированным гитом.

Далее файл .gitignore без расширений и далее путем glob шаблонов вписывать те объекты что нам не требуется заливать на Гит, как например временные файлы .tmp и тому подобное.

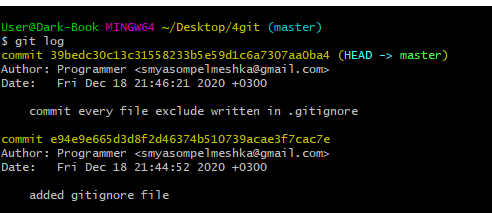


Изображение 1 – Демонстрация файла .gitignore

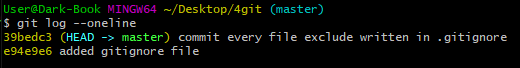
В данном примере мы закомментировали все файлы с названием, начинающимся на *password,* а также папку *logs.*

1. **Вывести лог коммитов в измененном пользовательском формате.**

Привычно можно вывести логи коммитов командой *git log*.

Изображение 2 – Вывод стандартных логов.

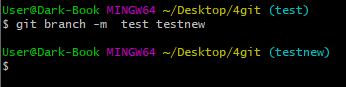
Но так же можно добавить параметр функции –*oneline* который выведет лишь 16ричный код коммита и сам комментарий.

Изображение 3 – Вывод лога с использованием параметра *–oneline*

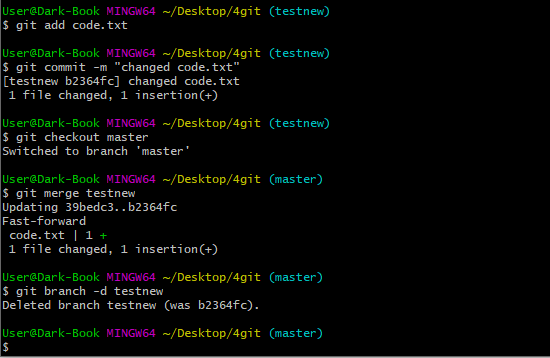
1. **Переименовать ветку.**

*В Git переименовать ветку можно одной простой командой*

*git branch -m old\_branch new\_branch*

**Изображение 4 – переименование ветки

1. **Провести в ветке работу и слить с веткой master, удалить слитую ветку.**

**Изображение 5 – Удаление ветки

В Git удаление ветки производится при помощи команды

*git branch -d имя\_ветки*

По заданию 4 необходимо провести работу в созданной ветке, слить её с веткой мастер и удалить, что продемонстрировано в Изображении 5.

1. **Настроить удалённый репозиторий: получить изменения из репозитория и отправить изменения.**

Посмотреть список доступных удаленных репозиториев можно при помощи команды *git remote.* Она выведет названия доступных удалённых репозиториев. Если вы клонировали репозиторий, то увидите как минимум origin — имя по умолчанию, которое Git даёт серверу, с которого производилось клонирование.

Изображение 6 – Список доступных репозиториев.

Для того чтобы добавить удаленный репозиторий существует команда

*Git remote add <shortname> <url>*

Чтобы получать изменения из удаленного репозитория необходимо изпользовать команду

*Git fetch [remote-name]*

Данная команда связывается с указанным удалённым проектом и забирает все те данные проекта, которых у вас ещё нет. После того как вы выполнили команду, у вас должны появиться ссылки на все ветки из этого удалённого проекта, которые вы можете просмотреть или слить в любой момент.

Для того чтобы наоборот отправить свои наработки на удаленный репозиторий необходимо использовать команду

*git push <remote-name> <branch-name>*

Но это работает с удаленными репозиториями на которых у вас есть права доступа.

**Выводы:**

Научились настраивать файл. gitignore для игнорирования файлов с определенным расширением при помощи glob-шаблонов.

Выводить логи коммитов в измененном пользовательском формате а так же переименовывать ветку.

После проведения тестовых работ на созданной ветке сливать и удалять созданную ветку.

Настраивать удаленный репозиторий а так же получать и отправлять наработки на удаленные репозитории при помощи команд pull И push.