**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

Отчет по лабораторной работе № 1

по дисциплине Java-программирование

Выполнил: студент группы БИБ1902 \_\_\_\_\_\_\_Арустамян А.Б.

Проверил: \_\_\_\_\_\_\_\_Херсонский А. В.

Москва, 2021

**Основы работы с системой контроля версий**

**Цель работы:** изучение базовых функций системы контроля версий на примере 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

2. Внесение ряда изменений в файл(ы) и выполнение коммитов (кол-во не меньше 2)

3. Создание веток и выполнение ряда изменений в проекте в созданных ветках

4. Слияние ветки с веткой мастер без конфликтов

5. Слияние ветки с веткой мастер с решением возникающих конфликтов

6. Откат изменений до одного из предыдущих коммитов

**Создание репозитория**

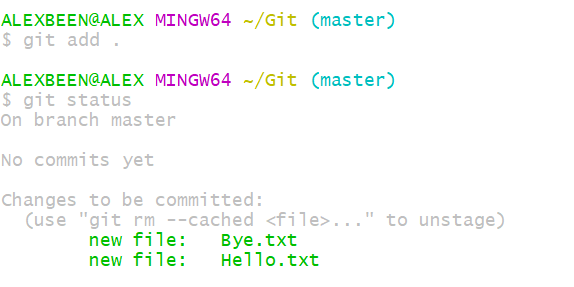
Для того чтобы создать репозиторию, необходимо открыть папку, в которой будет создаваться репозитория и выполнить следующую команду: $ git init



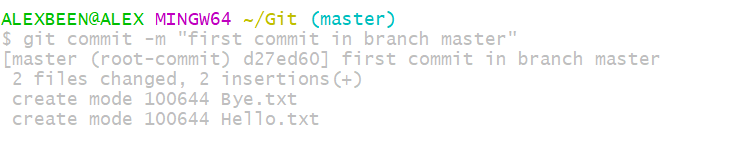
**Внесение ряда изменений в файл и выполнение коммитов**

Для того чтобы создать коммит нужно сначала проиндексировать файл с помощью: $ git add «Название файла»

С помощью: $ git status можно посмотреть в каком состоянии находиться файлы



После чего пишем: $ git commit -m «Осмысленный коментарий»



**Создание веток и выполнение ряда изменений в проекте в созданных ветках**

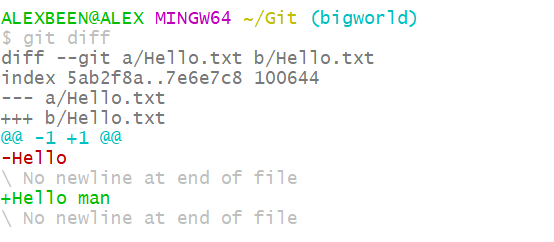
Для того чтобы создать ветку необходимо прописать: $ git checkout -b имя ветки

Для просмотра всех веток можно воспользоваться командой: $ git branch –a

Звездочка указывает на ту ветку в которой сейчас находимся



С помощью: $ git diff можно посмотреть какие изменения были внесены в файл

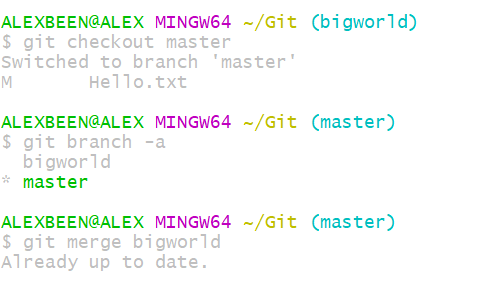


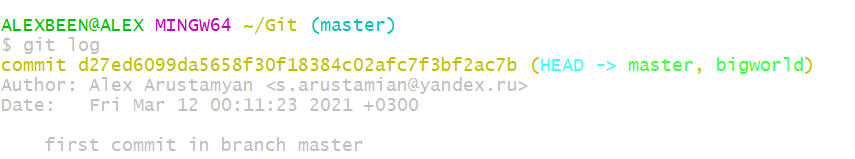
**Слияние ветки с веткой мастер без конфликтов**

Чтобы сделать слияние нужно перейти в ветку с помощью: $ git checkout master

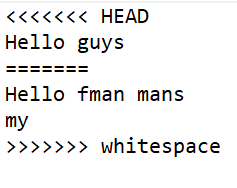
После чего сделать слияние с помощью команды: $ git merge “ `название ветки, которую хотим слить с текущей веткой` ”

С помощь: $ git log можно убедиться ,что ветки слились в одну





**Слияние ветки с веткой мастер с решением возникающих конфликтов**

Ошибки при слиянии могут возникнуть из-за того ,что были изменены одинаковые строчки в фале в разных ветках и после чего сделали слияние. Гит конфликт прописывает в файле после чего нужно вручную исправить этот конфликт. После редактирования файла осуществляем индексацию файла и выполнение коммита.gi

|  |  |
| --- | --- |
|  |  |

**Откат изменений до одного из предыдущих коммитов**

Для того чтобы откатить комиит

$ git reset --hard HEAD~(указать на сколько откатиться назад или hash нужно коммита)

Также можно использовать флаги --mixed и -- soft

|  |  |  |
| --- | --- | --- |
|  |  |  |



**Вывод**

С помощью Git-a можно откатить свой проект до более старой версии, сравнивать, анализировать или сливать свои изменения в репозиторий.

**Список использованных источников**

1 Камаев В.А., Костерин В.В. Технологии программирования. М.: Высшая школа, 2006

2 Javarush

Git:

1 Scott Chacon, Ben Straub «Pro Git»

2 https://javarush.ru/groups/posts/2683-nachalo-rabotih-s-git-podrobnihy-gayd-dlja-novichkov