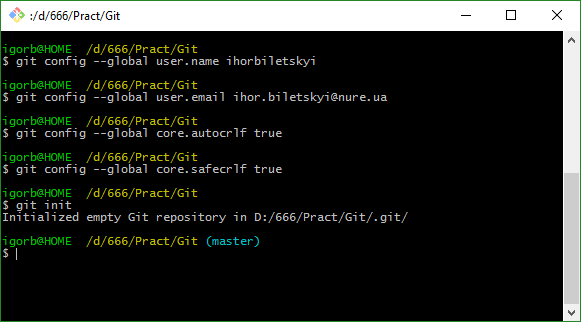
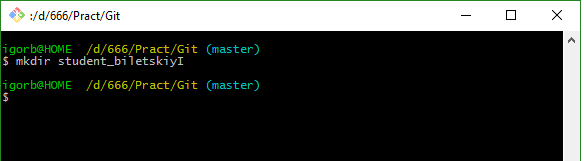
**Часть 1. Инициализация репозитория и первый коммит.**

1. Создать рабочий каталог.

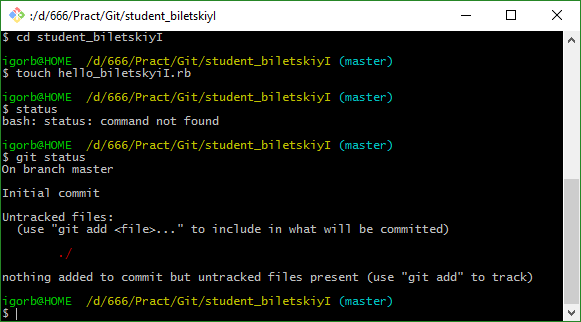
Инициализировать в нем репозиторий. Настроить файл config.



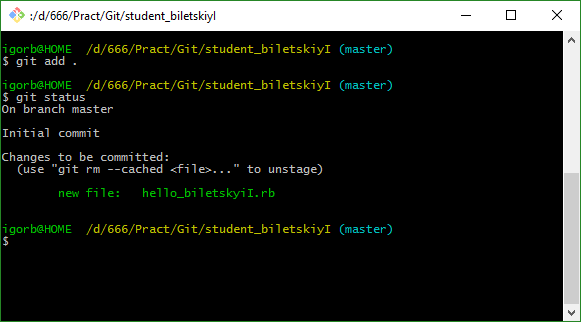
1. Создать в ней рабочий каталог c именем student\_name (где name – Ваша фамилия).



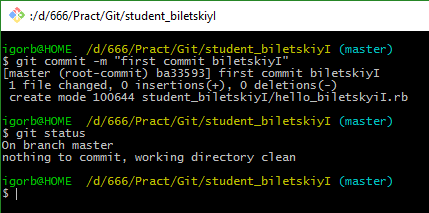
1. Создать в рабочем каталоге 3 файла проекта: (пример hello\_name.rb) (где name – Ваша фамилия). - проверить статус репозитория



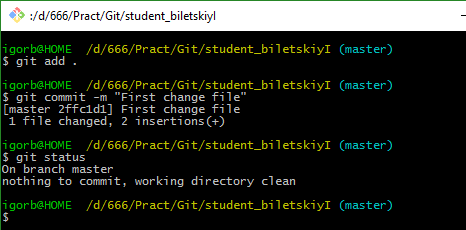
1. Проиндексировать файлы.



1. Сделать первый commit с комментарием ‘’fist commit name” (где name – Ваша фамилия)

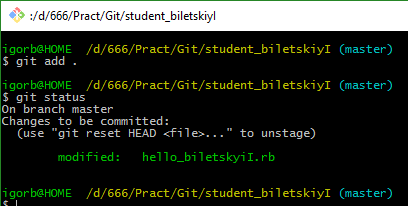


1. Сделать изменения в файлах. - cделать commit изменений.

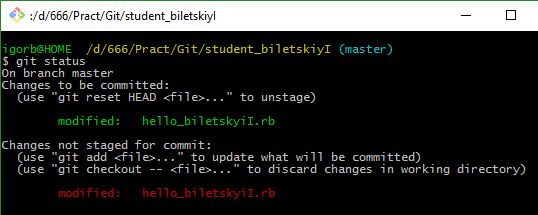


**Часть 2. Работа с изменениями**

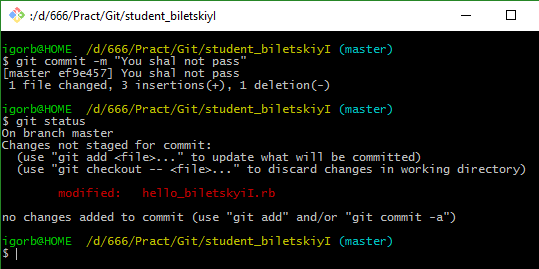
1. Сделать изменения в одном из файлов – файл 1. Проиндексировать файл



2) Сделать новые изменения в файле1. - проверить статус



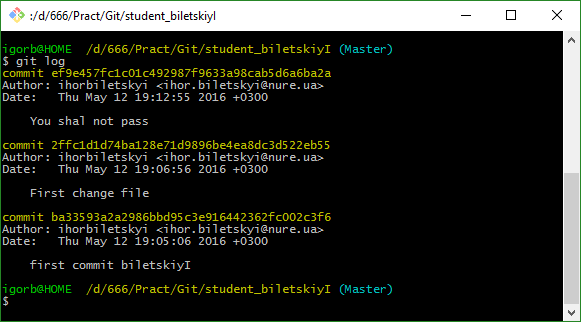
3) Сделать commit - проверить статус



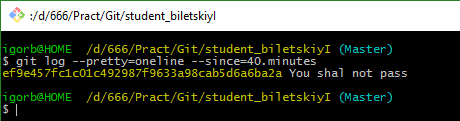
Объяснить результат:

Для добавления был сделан snapshot файла, но он не был добавлен в репозиторий, а был изменен. В результате, мы отправили старый snapshot в репозиторий, а текущее состояние файла считается modified.

4) Вывести историю проекта. $ git log



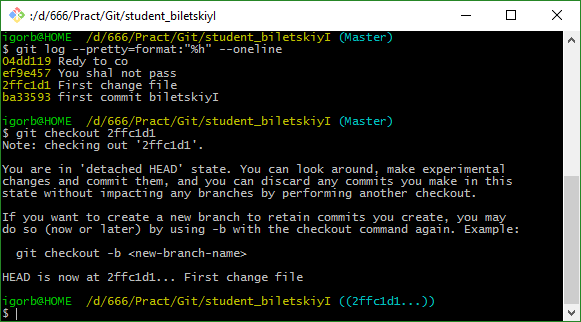
5) Используйте команду $ git log --pretty=oneline , настраивая соответствующие параметры выведите на экран последние commits за 5 минут.



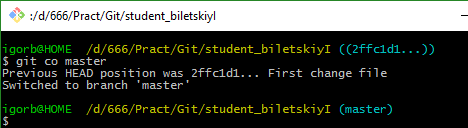
На стриншоте команда, то выводу комитов за последние 40 минут, а не 5, по тому что за последние 5 минут не было комитов и плохо отобразить правильность работы команды.

**Часть 3. Работа с предыдущими состояниями репозитория.**

1. Выполните возврат состояния файлов рабочего каталога на второй commit со времени начала лаб.работы ( ОБЯЗАТЕЛЬНО используйте команду git checkout <hash>)

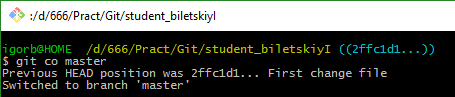


1. Произведите возврат состояния файлов к последнему commit .

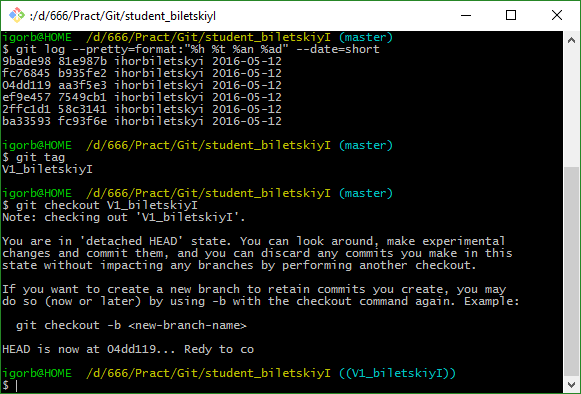


**Часть 4. Работа с тэгами.**

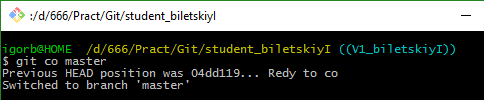
1. Создайте тэг в текущем snapshot репозитория. Название тэга - V1\_name (где name – Ваша фамилия)



1. Выполните изменения и как минимум два commits в текущем каталоге.
2. Продемонстрируйте возврат к состоянию репозитория помеченную тегом.

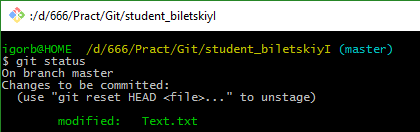


1. Выполните возврат к последнему состоянию master

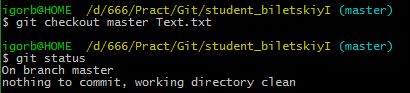


**Часть 5. Работа с функцией revert**

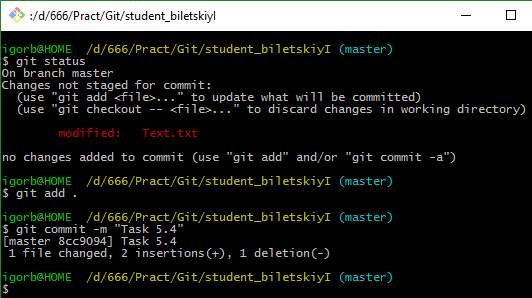
1. Выполните изменения текущего состояния рабочего каталога.
2. Проиндексируйте изменения



1. Командой revert верните исходное состояние файлов



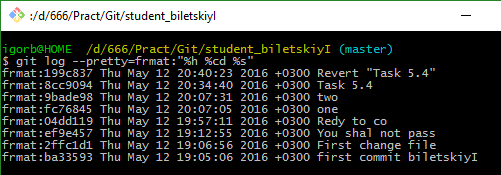
1. Сделайте изменения состояния каталога
2. Проиндексируйте и выполните commit.



1. Выполните revert последнего commit



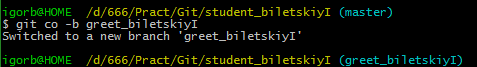
1. Продемонстрируйте log



**Лабораторная работа №3 часть 2**

**Часть 6. Создание ветвей и слияние**

1. Создайте новую ветвь greet\_name (где name – Ваша фамилия), используя команду git checkout -b <branchname>



1. Добавьте новый каталог (пример lib\_name) и файл greeter\_ name.rb c содержимым:

class Greeter

def initialize(who)

@who = who

end

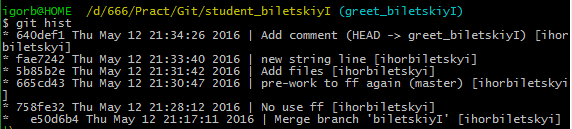
def greet

"Hello, #{@who}"

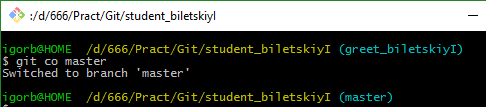
end

end

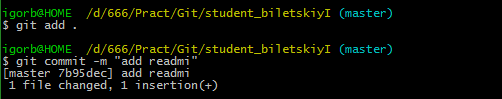
1. Сделайте 3 commits в ветку greet\_name



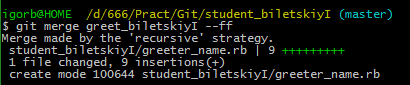
1. Продемонстрируйте переключение на ветку master



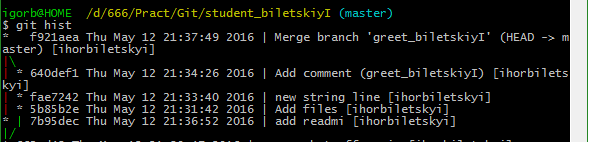
1. Создайте файл README и произведите commit в master.



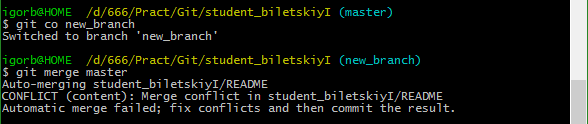
1. Произведите слияние ветвей greet и master – продемонстрируйте историю commits. (используйте опцию Fast forward)

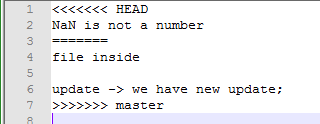


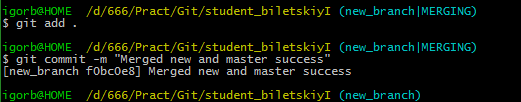
1. Изобразите в удобном для Вас графическом редакторе граф коммитов, где вершинами графа будут коды коммитов.



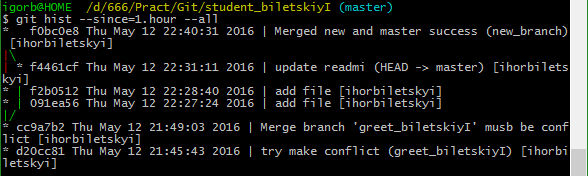
**Часть 7.  
Смоделируйте искусственно конфликт, при слиянии описанном в предыдущей части.**







Вручную внесли изменения в файл и завершили слияние коммитом.



**Часть 8.**

**Используя команду reset верните указатель на состояние ветвей до**

**их слияния.**



**Часть 9.**

**Проделайте шаги 3-5 в части 6. Выполните слияние изменений командой rebase. Изобразите в удобном для Вас графическом редакторе граф коммитов, где вершинами графа будут коды коммитов**

