**Белорусский государственный технологический университет**

**Факультет информационных технологий**

**Специальность программная инженерия**

Отчёт по лабораторной работе №9-11

По дисциплине «Основы программной инженерии»

На тему «Системы контроля версий. Основы работы в Git. Работа с GitHub»

Выполнил:

Студент 1 курса 10 группы

Мамонько Денис Александрович

Преподаватель: асс. Архипенко О.А.

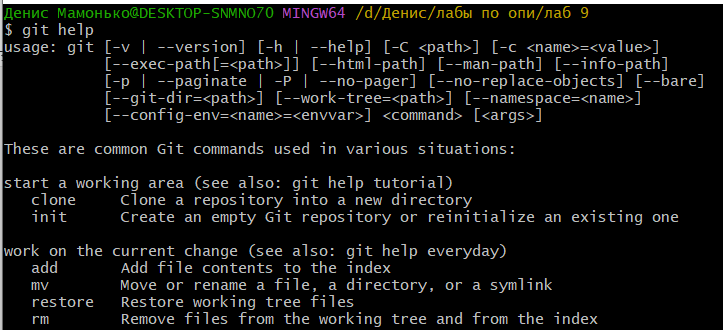
2023, Минск

**Лабораторная работа №9**

**Задание 4**

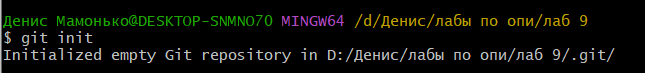
****

**Задание 5**

****

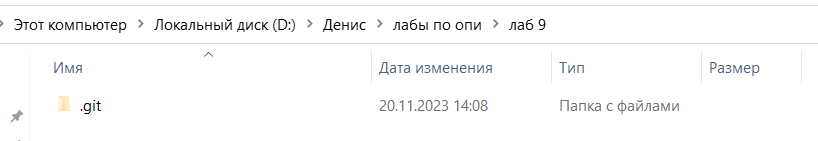
Все возможные команды можно получить с помощью команды $ git help.

**Задание 6**

****

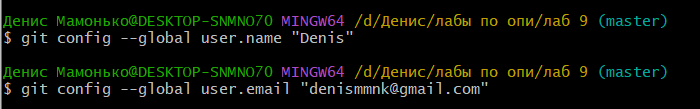
Перейти в проводнике в рабочую папку, где планируется создать репозиторий, и запустить Git Bash с помощью контекстного меню: «Git Bash Here».

Инициализация репозитория в выбранной папке выполняется командой $ git init.

****

В папке появилась новая скрытая папка: .git – локальный репозиторий.

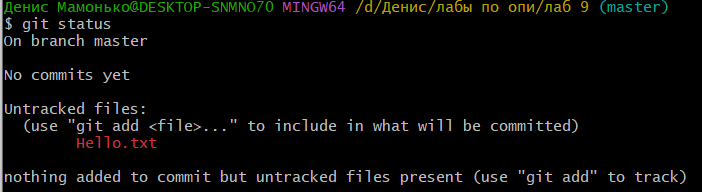
**Задание 7**

****

**Задание 8**

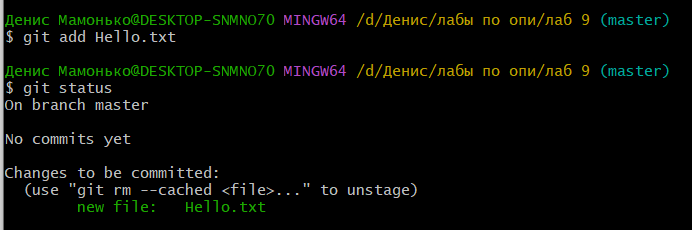
****

**Задание 11**

****

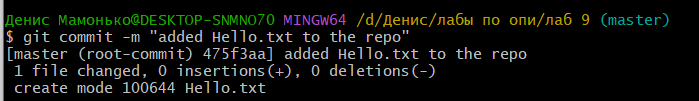
Текущее состояние (status) репозитория отображается командой: $ git status Кроме того, была создана ветка master. Красным цветом отмечаются новые и модифицированные файлы и папки.

**Задание 12**

****

Добавим файл Hello.txt в репозиторий индексированных файлов командой: $ git add Hello.txt. Текущее (обновленное) состояние репозитория отображается командой: $ git status. Теперь файл Hello.txt проиндексирован.

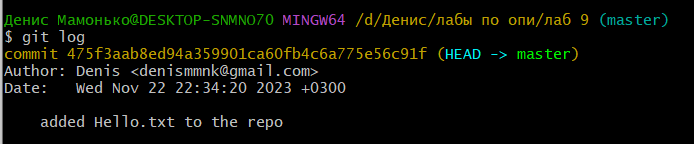
**Задание 13**

****

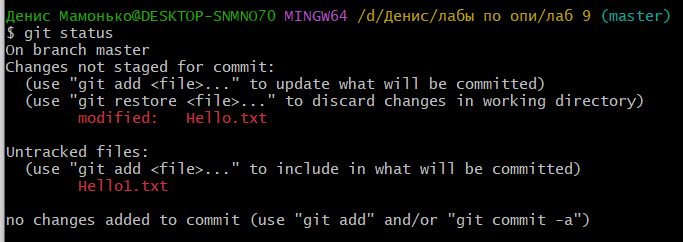
$ git commit -m "added Hello.txt to the repo". Ключ –m позволяет добавить комментарий, описывающий, что именно было изменено в коммите ("added Hello.txt to the repo"). Git информирует об успешном создании нового коммита (в ветку master добавлен 1 файл):

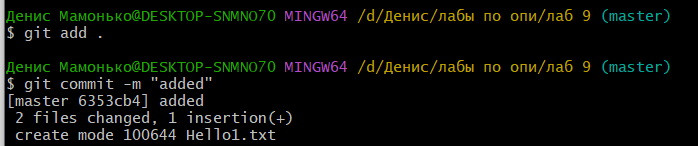
****

**Задание 14**

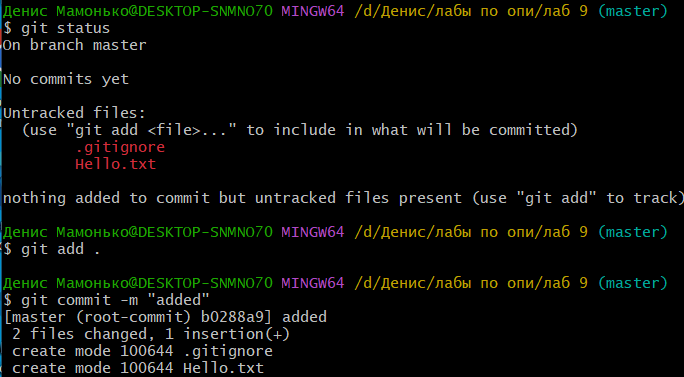
****

**Задание 15**

****

****

**Задание 17**



Мы создали внутри нашего локального репозитория текстовый документ. gitignore, внутри которого был помещен другой файл Hello1.txt, это делается для того, чтобы файл Hello1.txt не был загружен на сервер и его никто не видел кроме самого пользователя внутри своей локальной папки.

**Вопросы**

1. Система управления версиями (от англ. Version Control System, VCS или Revision Control System, RCS) – программное обеспечение для облегчения работы с изменяющейся информацией и разработки проекта совместно с коллегами.

2. Git использует принцип хранения снимков, а не изменений.

3. В Git все данные – это набор снимков состояния миниатюрной файловой системы, а в других системах хранимые данные – это набор файлов и изменений, которые вносились в эти файлы в течение их жизни.

4. Модифицированное, индексированное, зафиксированное.

5. Индексация файла в Git – это процесс добавления или обновления файла в вашем текущем рабочем каталоге в Git индексе. Команда: $git add.

6. Фиксация файла в Git – это процесс сохранения ваших изменений в Git репозитории. Команда: $git commit -m “”.

7. Нужно использовать $git config --list.

8. Используем команду $git init.

9. Используем команду $git add.

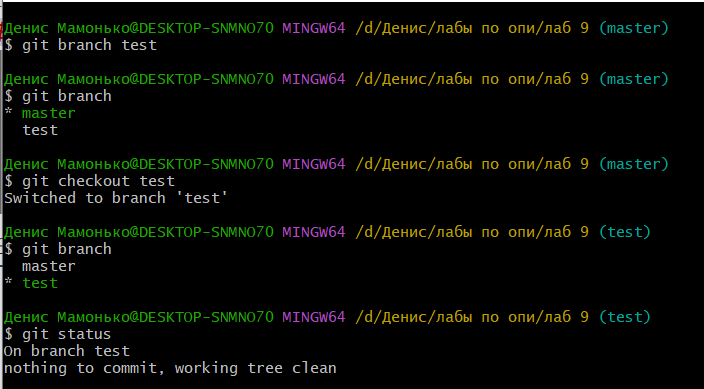
10. Используем команду $git commit -m””.

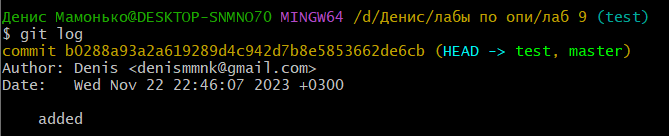
11. Используем команду $git status.

12. Файл. gitignore – это текстовый файл, который сообщает Git, какие файлы или папки следует игнорировать в проекте. Обычно этот файл размещается в корневом каталоге проекта. Также можно создать глобальный файл, и все записи в этом файле будут игнорироваться во всех репозиториях Git.

**Лабораторная работа №10**

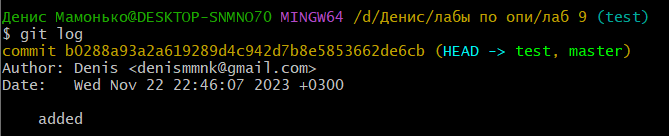
**Задание 3**



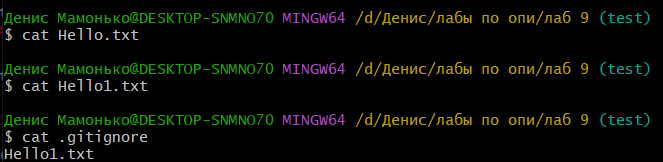


**Задание 4**

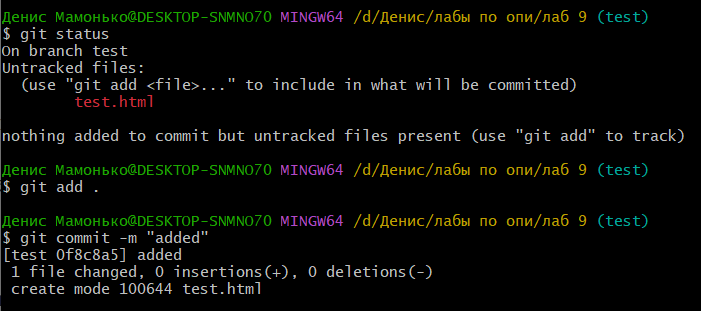
HEAD указывает на ветку test:



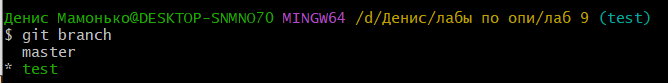
**Задание 5**

****

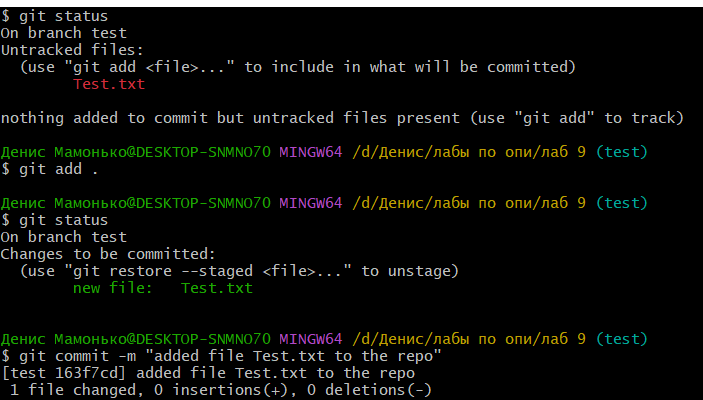
**Задание 6**

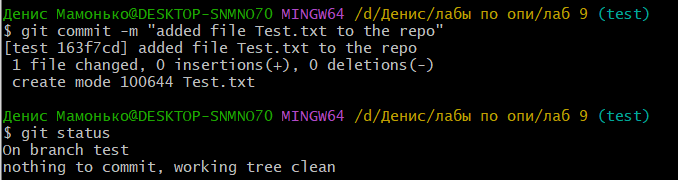
****

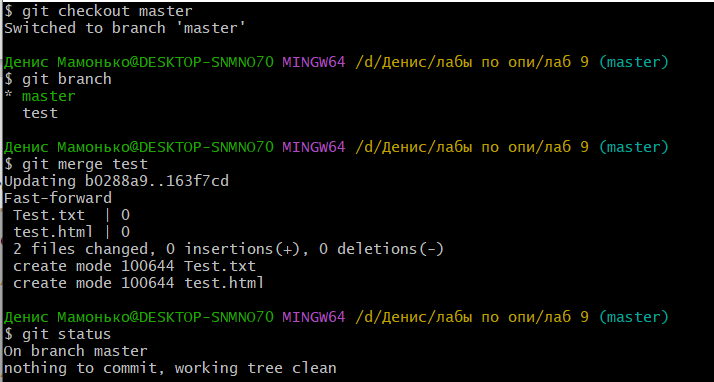
**Задание 7**

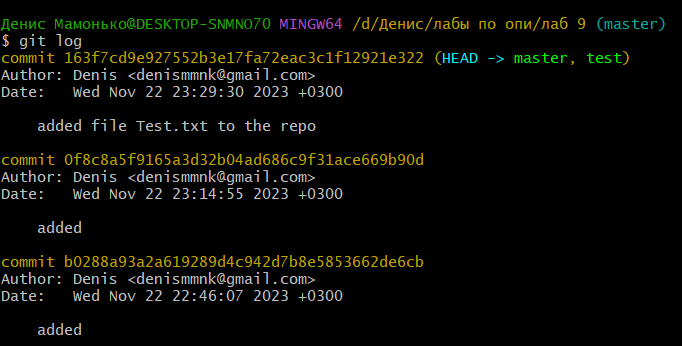
****

**Задание 8**

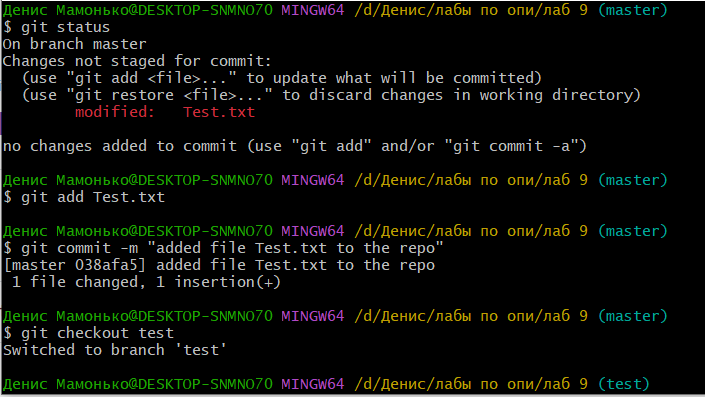
****

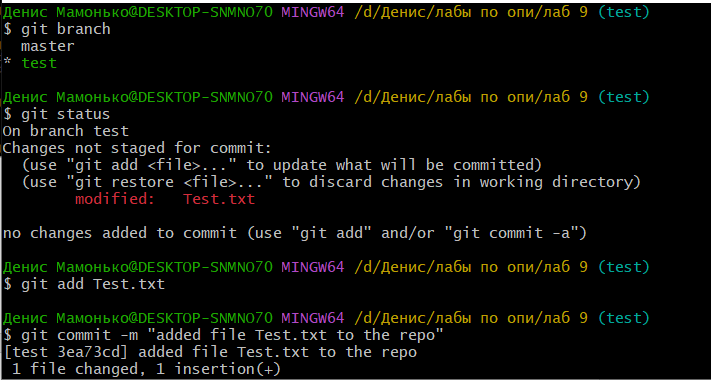
****

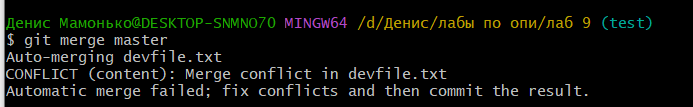
****

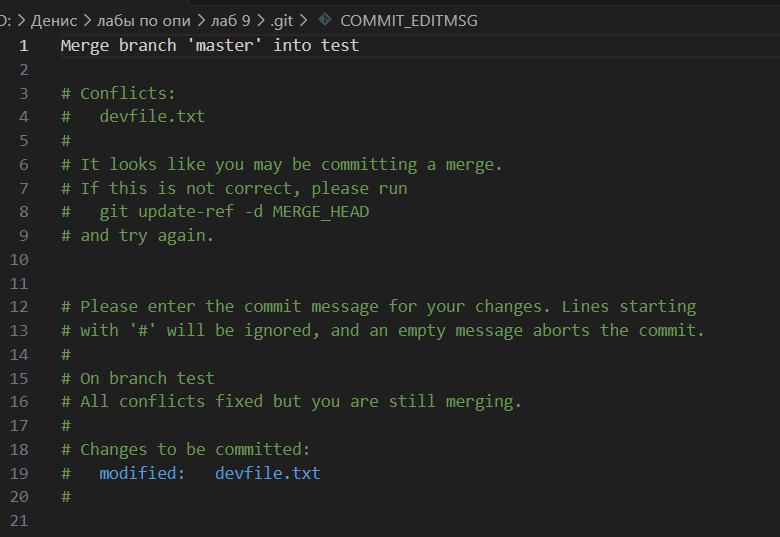
****

**Задание 9**

****

****

****

****

**Вопросы**

1. Все коммиты в Git логируются. Просмотреть журнал можно с помощью команды: $git log.

2. Ветвление (branching) означает отклонение от основной линии разработки, после которого работа перестает затрагивать основную линию и переходит в ветвь.

3. С помощью этой команды происходит переключение веток.

4. Указывает на смены веток, которые меняют файлы в рабочей папке.

5. Создание ветки (branch) выполняется следующей командой: $ git branch.

6. $git checkout.

7. Команда merge используется для слияния веток.

8. $git log.

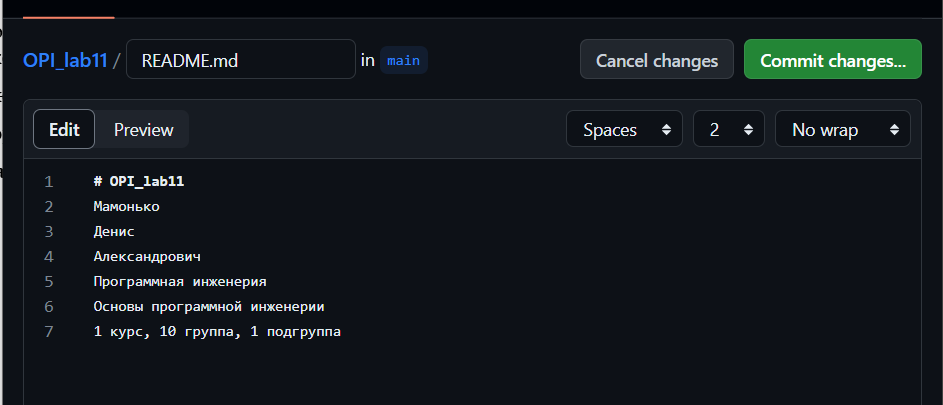
9. Нужно выбрать ветку, с изменениями которой мы хотим осуществить слияние, и выбрать ветку, слияние в которую будет осуществляться с помощью команды git merge.

10. Сбой в процессе слияния говорит о наличии конфликта между текущей локальной веткой и веткой, с которой выполняется слияние.

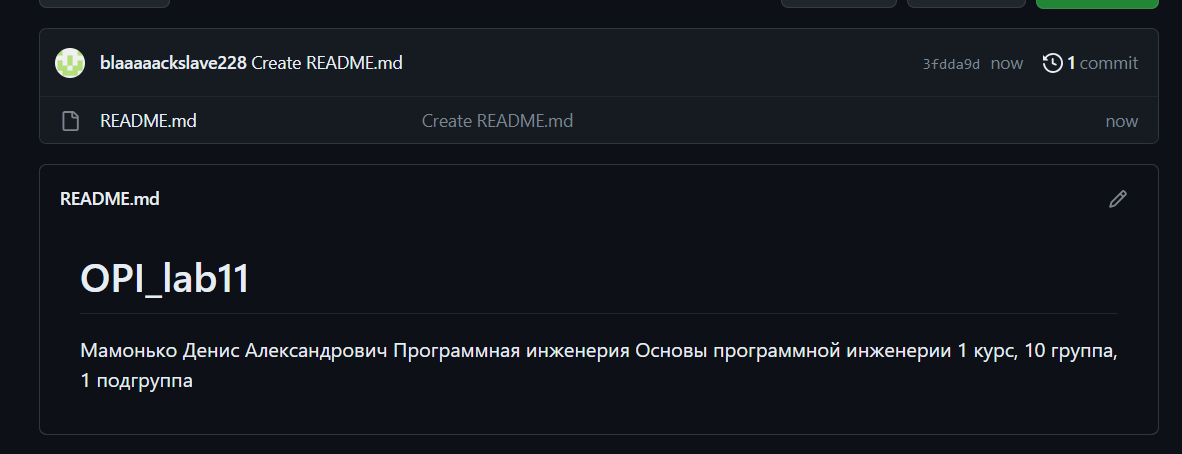
Конфликты разрешаются вручную.

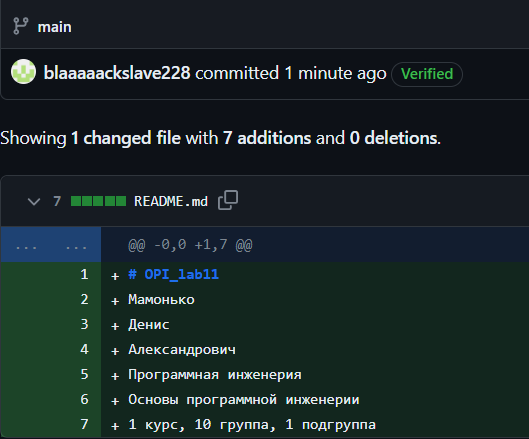
**Лабораторная работа №11**

**Задание 5-7**

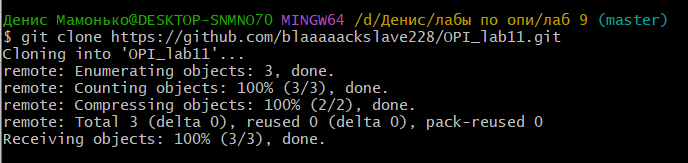
****

**Задание 8**

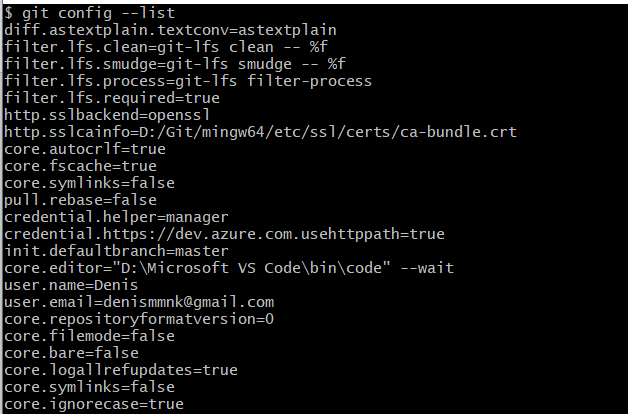
****

****

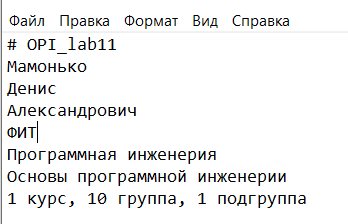
**Задание 10**

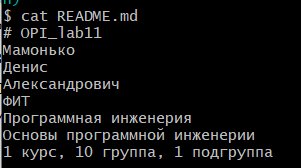
****

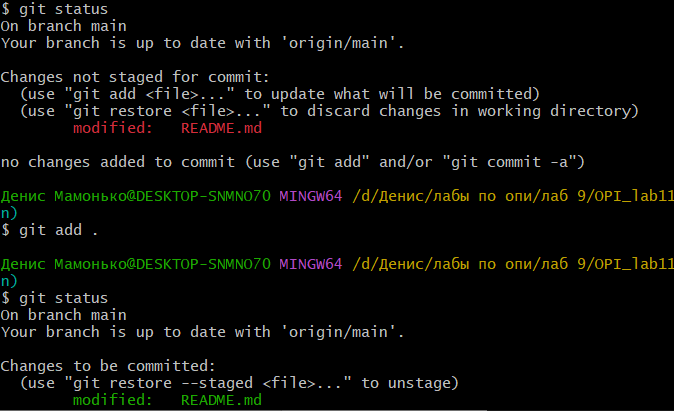
**Задание 11**

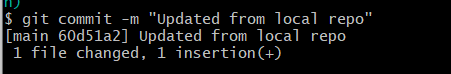
****

**Задание 12**

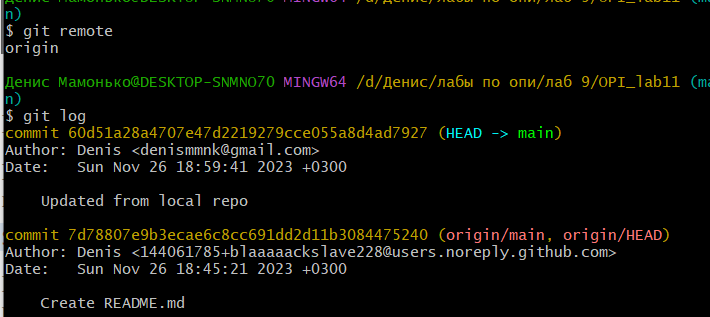
****

****

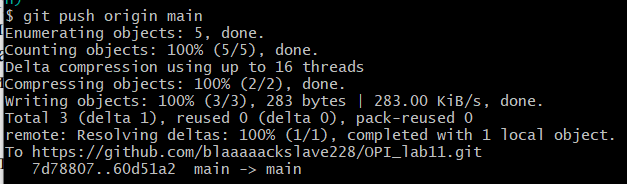
****



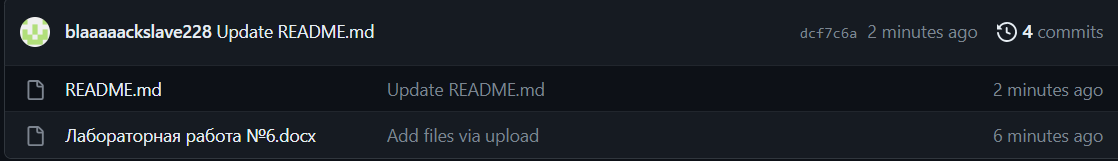
**Задание 13**

****

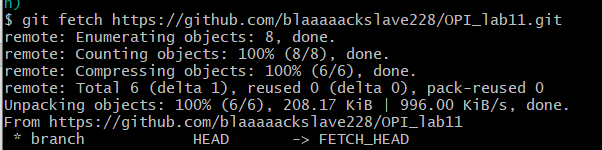
**Задание 14**

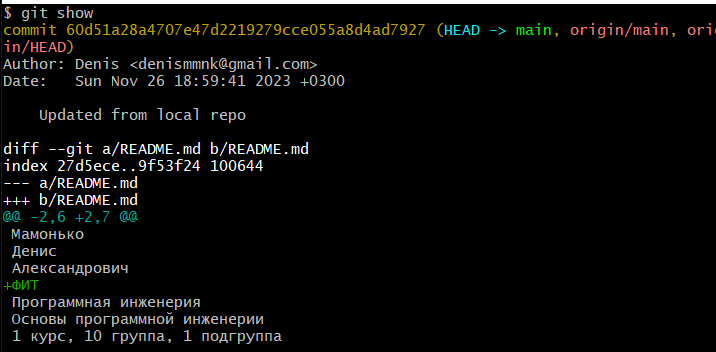
****

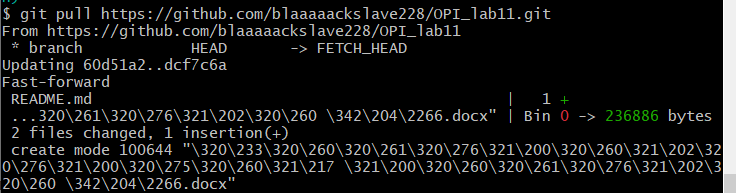
**Задание 16**

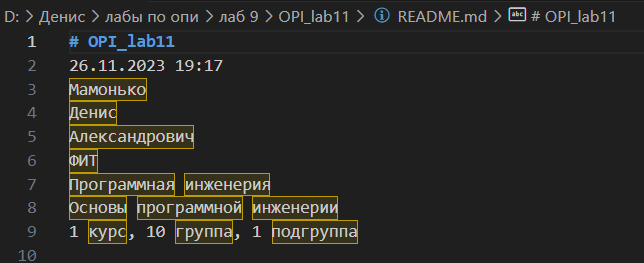
****

**Задание 18**

****

****

****

****

****

**Вопросы**

1.Github – крупнейший веб-сервис онлайн-хостинга репозиториев, используется для хостинга IT-проектов и их совместной разработки. Основан на системе контроля версий Git.

2. Зайти на GitHub → нажать на «+» и выбрать «New Repository» → введите имя вашего репозитория → выберите, будет ли ваш репозиторий публичным (открытым для всех) или приватным (только для вас и людей, которых вы пригласите) → если хотите, можете добавить описание репозитория → если нужно, выберите, чтобы добавить файл README, .gitignore или лицензию → нажмите на кнопку «**Create repository».**

**3.** Клонирование в Git - это процесс копирования удаленного репозитория на ваш локальный компьютер. Это позволяет вам работать с файлами репозитория локально, делать изменения и затем отправлять эти изменения обратно в удаленный репозиторий.

4. **git log:** Эта команда показывает историю коммитов для текущей ветки. Вы можете увидеть, когда были сделаны изменения, кто их сделал и описание этих изменений.

**git fetch** и **git pull:** Эти команды используются для получения последних изменений из удаленного репозитория. git fetch получает последние изменения, но не объединяет их с вашей текущей веткой. git pull делает то же самое, что и git fetch, но также объединяет изменения с вашей текущей веткой.

5. Git init и git clone.

6. Git remote.

7. С помощью команды git push.

8. По команде git fetch выполняется обращение к указанному удаленному репозиторию и забираются все те данные проекта, которых у вас ещё нет. При этом их слияния с вашими наработками не происходит и то, над чем вы работаете в данный момент, не модифицируется. Можно использовать команду git pull чтобы автоматически получить изменения из удалённой ветки и слить их со своей текущей.

9. Совместная работа с репозиторием требуется, когда необходимо учитывать текущие задачи, выполнять требования к ним и исправлять баги.

10. Откройте репозиторий, который вы хотите скопировать → В правом верхнем углу страницы нажмите на кнопку **«Fork»** → GitHub создаст копию репозитория и добавит ее в ваш аккаунт.

11. Команда-запрос, используя которую можно подать запрос коллеге, предложив свои изменения в репозитории.