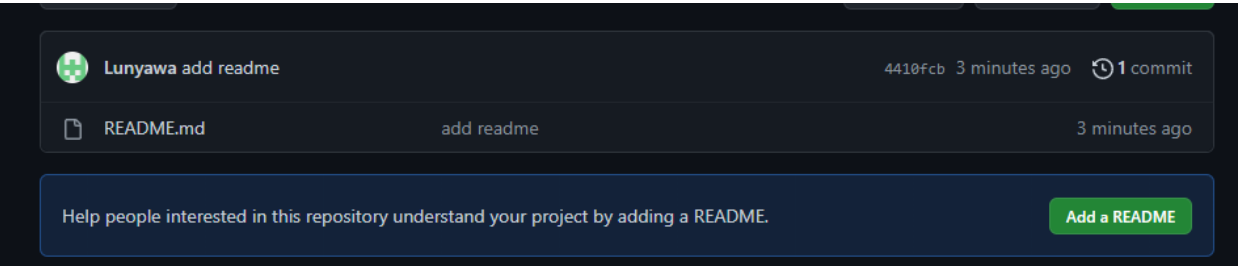


# Основы программной инженерии

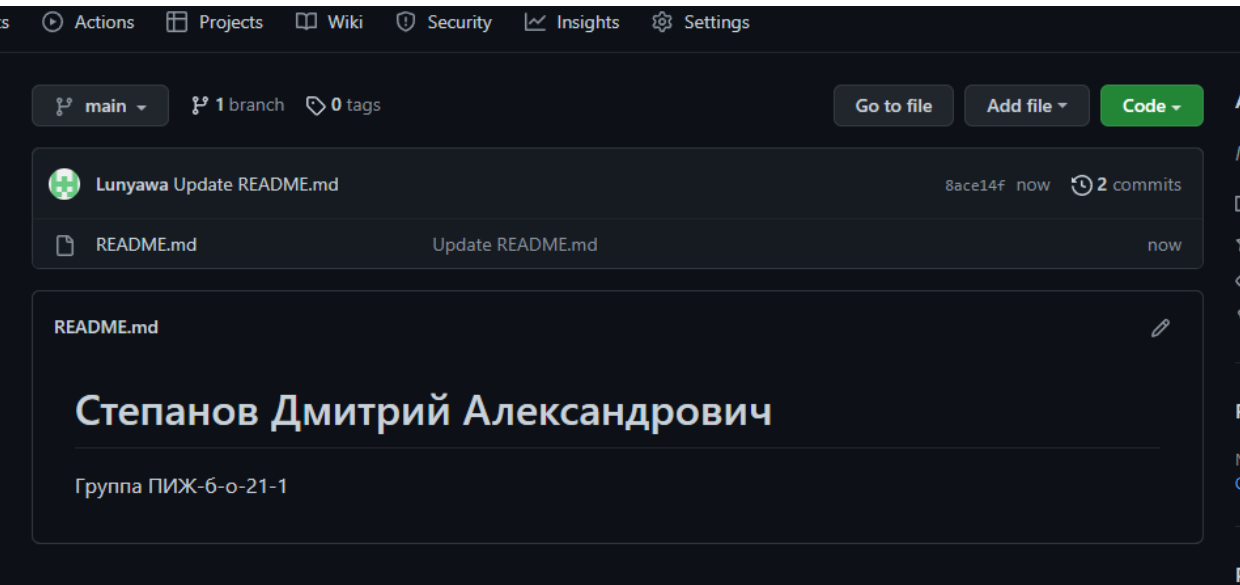
## Отчёт по лабораторной работе №1

№1

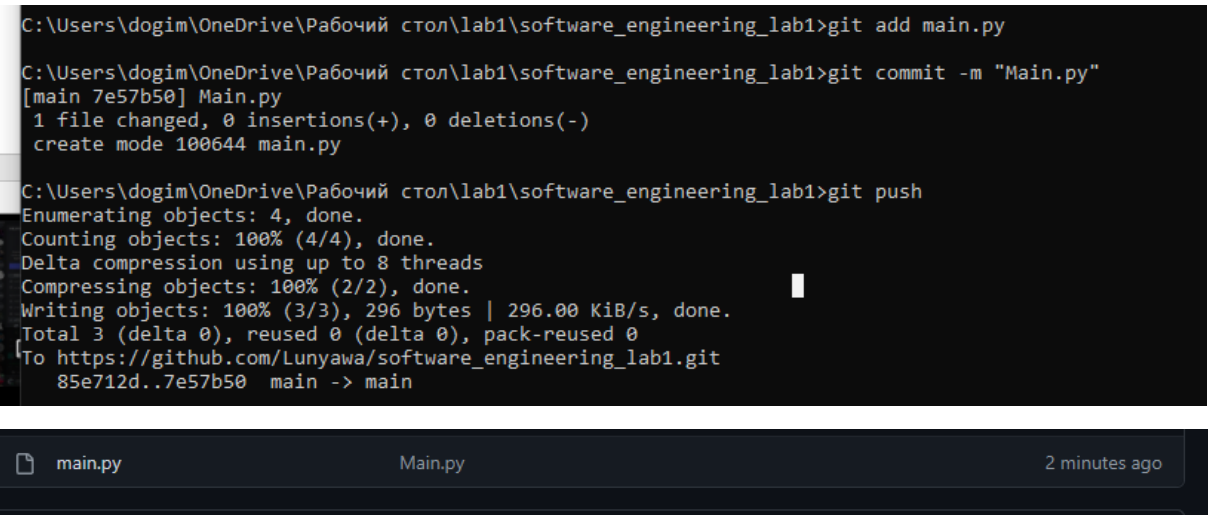
```
C:\Users\dogim\OneDrive\Рабочий стол\lab1\software_engineering_lab1>git add .  
C:\Users\dogim\OneDrive\Рабочий стол\lab1\software_engineering_lab1>git commit -m "add readme"  
[root commit] 4410fcb1 add readme
```



№2



№3



## №4

```
C:\Users\dogim\OneDrive\Рабочий стол\lab1\software_engineering_lab1>git add .

C:\Users\dogim\OneDrive\Рабочий стол\lab1\software_engineering_lab1>git commit -m "hello world"
[main 453ccb1] hello world
1 file changed, 1 insertion(+)

C:\Users\dogim\OneDrive\Рабочий стол\lab1\software_engineering_lab1>git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 265 bytes | 265.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Lunyawa/software_engineering_lab1.git
b0984f7..453ccb1 main -> main
```

## №5

```
C:\Users\dogim\OneDrive\Рабочий стол\lab1\software_engineering_lab1>git add .

C:\Users\dogim\OneDrive\Рабочий стол\lab1\software_engineering_lab1>git commit -m "print sum = 10 + 5"
[main eded6eb] print sum = 10 + 5
1 file changed, 2 insertions(+), 1 deletion(-)

C:\Users\dogim\OneDrive\Рабочий стол\lab1\software_engineering_lab1>git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 272 bytes | 272.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Lunyawa/software_engineering_lab1.git
453ccb1..eded6eb main -> main
```

## №6

```
C:\Users\dogim\OneDrive\Рабочий стол\lab1\software_engineering_lab1>git add .

C:\Users\dogim\OneDrive\Рабочий стол\lab1\software_engineering_lab1>git commit -m "print sum = a + b"
[main 0dd164a] print sum = a + b
1 file changed, 2 insertions(+), 1 deletion(-)

C:\Users\dogim\OneDrive\Рабочий стол\lab1\software_engineering_lab1>git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 303 bytes | 303.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Lunyawa/software_engineering_lab1.git
eded6eb..0dd164a main -> main
```

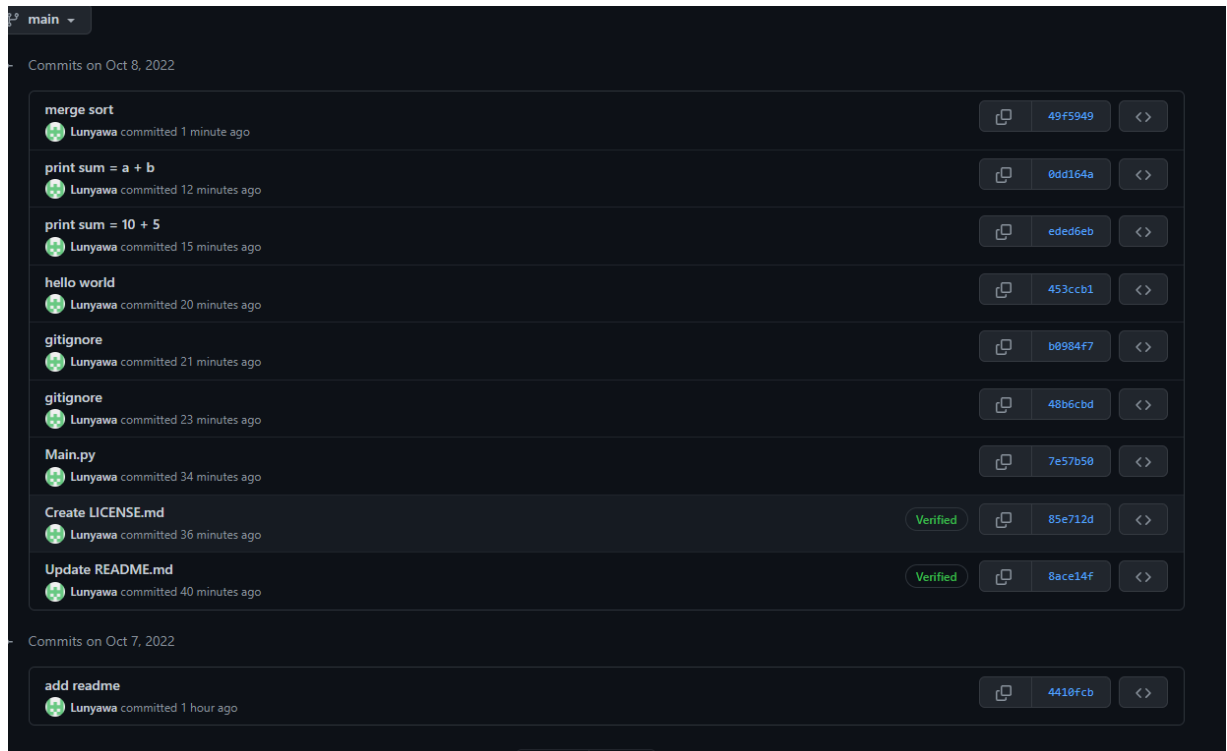
## №7

```
C:\Users\dogim\OneDrive\Рабочий стол\lab1\software_engineering_lab1>git add .

C:\Users\dogim\OneDrive\Рабочий стол\lab1\software_engineering_lab1>git commit -m "merge sort"
[main 49f5949] merge sort
1 file changed, 30 insertions(+), 3 deletions(-)

C:\Users\dogim\OneDrive\Рабочий стол\lab1\software_engineering_lab1>git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 525 bytes | 525.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Lunyawa/software_engineering_lab1.git
0dd164a..49f5949 main -> main
```

## История всех коммитов



### Ответы на вопросы

#### 1. Что такое СКВ и каково ее назначение?

- Система контроля версий (СКВ) — это система, изменения в одном или нескольких файлах с тем, чтобы в дальнейшем была вернуться к определённым старым версиям этих файлов. Программисты обычно помещают в систему контроля версий исходные коды программ, но на самом деле под версионный контроль можно поместить файлы практически любого типа.

#### 2. В чем недостатки локальных и централизованных СКВ?

- Локальные СКВ: невозможность работать с другими разработчиками.  
Центральные СКВ: единая точка отказа, представленная централизованным сервером.

#### 3. К какой СКВ относится Git?

- Распределённые СКВ.

#### 4. В чем концептуальное отличие Git от других СКВ?

- Git не хранит и не обрабатывает данные в виде набора файлов и изменений. Вместо этого, подход Git к хранению данных больше похож на набор снимков миниатюрной файловой системы. Каждый раз, когда вы делаете коммит, то есть сохраняете состояние своего проекта в Git, система запоминает, как

выглядит каждый файл в этот момент, и сохраняет ссылку на этот снимок.

5. Как обеспечивается целостность хранимых данных в Git?

- В Git для всего вычисляется хеш-сумма, и только потом происходит сохранение. В дальнейшем обращение к сохранённым объектам происходит по этой хеш-сумме. Это значит, что невозможно изменить содержимое файла или директории так, чтобы Git не узнал об этом.

6. В каких состояниях могут находиться файлы в Git? Как связаны эти состояния?

7. Что такое профиль пользователя в GitHub?

8. Какие бывают репозитории в GitHub?

9. Укажите основные этапы модели работы с GitHub.

10. Как осуществляется первоначальная настройка Git после установки?

11. Опишите этапы создания репозитория в GitHub.

12. Какие типы лицензий поддерживаются GitHub при создании репозитория?

13. Как осуществляется клонирование репозитория GitHub? Зачем нужно клонировать репозиторий?

14. Как проверить состояние локального репозитория Git?

15. Как изменяется состояние локального репозитория Git после выполнения следующих

операций: добавления/изменения файла в локальный репозиторий Git;

добавления нового/

измененного файла под версионный контроль с помощью команды `git add` ; фиксации

(коммита) изменений с помощью команды `git commit` и отправки изменений на сервер с

помощью команды `git push` ?

16. У Вас имеется репозиторий на GitHub и два рабочих компьютера, с помощью которых Вы

можете осуществлять работу над некоторым проектом с использованием этого репозитория.

Опишите последовательность команд, с помощью которых оба локальных репозитория,

связанных с репозиторием GitHub будут находиться в синхронизированном состоянии.

Примечание: описание необходимо начать с команды `git clone` .

17. GitHub является не единственным сервисом, работающим с Git. Какие сервисы еще Вам

известны? Приведите сравнительный анализ одного из таких сервисов с GitHub.

18. Интерфейс командной строки является не единственным и далеко не самым удобным способом работы с Git. Какие Вам известны программные средства с графическим интерфейсом пользователя для работы с Git? Приведите как реализуются описанные в лабораторной работе операции Git с помощью одного из таких программных средств.